University of Padova

Department of Information Engineering

Master Thesis in Automation Engineering

Anomaly detection for entertainment industry automatic machines

Supervisor

Professor Gian Antonio Susto

University of Padova

Co-supervisor

Doctor Chiara Masiero

Statwolf Data Science

Master Candidate

Marco Peroni

Academic Year

2021-2022

Sommario

L'industria 4.0 è un processo che sta portando alla produzione industriale del tutto automatizzata e interconnessa. Una direttrice di questo fenomeno è costituita dagli analytics: una volta raccolti i dati, bisogna ricavarne valore. Anche in ambito industriale le aziende stanno cercando di valorizzare la grande mole di dati che ne consegue. In questo contesto, i temi dell'apprendimento automatico forniscono uno strumento di indiscutibile rilevanza in svariate applicazioni. Questa tesi tratta l'utilizzo di un metodo di riconoscimento delle anomalie relativo ad un caso studio di un' azienda del settore dell'intrattenimento. Nello specifico si tratta di un contesto non supervisionato in quanto l'etichettatura dei dati non è disponibile a priori. In sinergia a ciò viene adoperato uno strumento che fornisca un'interpretabilità ai risultati ottenuti in modo da fornire un aiuto nell'analisi delle cause principali.

Abstract

Industry 4.0 is a process that is leading to fully automated and interconnected industrial production. A guideline of this phenomenon is constituted by analytics: once the data has been collected, it is necessary to derive value from it. Even in the industrial field, companies are trying to exploit the large amount of data that follows. In this context, machine learning topics provide a tool of indisputable relevance in a variety of applications. This thesis deals with the development of a feature-based anomaly detection method related to a case study of a company in the entertainment field. Specifically, motivated by the lack of a reliable labeled set, the approach takes shape in an unsupervised scenario. In synergy with this, a tool is adopted that provides the interpretability of the results. Understanding why a point is labeled anomalous is becoming increasingly important, especially by virtue of root cause analysis.

Contents

Listing of figures

Listing of tables

Listing of acronyms

AcME ..... Accelerated Model-agnostic Explanations

AD ..... Anomaly Detection

AI ..... Artificial Intelligence

API ..... Application Programming Interface

AS ..... Anomaly Score

DAD ..... Deep Anomaly Detection

DSS ..... Decision Support System

IF ..... Isolation Forest

ML ..... Machine Learning

PCA ..... Principal Component Analysis

PCC ..... Pearson Correlation Coefficient

PLC ..... Programmable Logic Controller

RMS ..... Root Mean Square

SD ..... Standard Deviation

SHAP ..... SHapley Additive exPlanations

XAI ..... eXplainable Artificial Intelligence

1 Introduction

1.1 Machine Learning and Industry 4.0

Industry 4.0 has been at the center of economic transformation in Italy and in the world for some years. During the *COVID-19* pandemic, Industry 4.0 and related technologies proved to be fundamental in countering the crisis. But what exactly is Industry 4.0?

Up to now, reference has been made to three major industrial revolutions in the Western world, each of which has led to a gradual improvement in working methods and has allowed the involvement of various production sectors thanks to the affirmation of new technologies.

- The first industrial revolution was the one that in the second half of the 18th century made it possible to mechanize production in the textile and metallurgical sector thanks to use of the steam engine.

- The second industrial revolution was instead conventionally started in 1870 with the introduction of electricity, chemicals, with the advent of the internal combustion engine and the consequent increase in the use of oil as a new energy source. It has favored the emergence of new communication and transport systems, mass production and the assembly chain with consequent increases in production capacities.

- The third industrial revolution finally took hold in 1970 with the birth of information technology. Hence the beginning of the digital era which was then destined to increase the levels of industrial automation using electronic systems and IT (Information Tech-

nology). During this period there was a strong push for technological innovation closely linked to the birth of computers, robots, the first spacecraft and satellites.

The change we are about to witness have an important role such as to have earned the significant title of the Fourth Industrial Revolution which will see the birth of new models, strategies and paradigms: the so-called Industry 4.0. While there is no commonly accepted definition, Industry 4.0 is generally seen as a process that will culminate in a new conception of the industry, from the development of new products and services, to research and innovation, to validation and production, with the least common denominator consisting of a high degree of automation and interconnection.

What are the main aspects of this phenomenon? The first relates to the management and storage of large amounts of data available on the network (big data) and acquired by objects with the ability to interact with each other thanks to a network, the so-called internet of things: remote controls, appliances, cars. These objects, suitably equipped with sensors, will be able to be interconnected to a network as today we are used to doing with smartphones or computers.

What to do with this huge amount of data? Here comes the second aspect consisting of the so-called analytics, that is the set of techniques and algorithms necessary to extract useful information from the data and, ultimately, derive a value from it. In this regard, the development of artificial intelligence techniques can play a fundamental role: machine learning, that is the automatic learning of machines, currently very little widespread on an industrial level, should undergo a real explosion in the coming months and years. The development of smart factories represents an incredible opportunity to enter the fourth industrial revolution for the manufacturing sector. Analyzing the large amounts of data collected by production department sensors provides real-time visibility into production assets and can provide tools for performing predictive maintenance to minimize equipment downtime. The use of highly technological IoT devices in smart factories leads to greater productivity and an improvement in quality. Replacing manual inspection business models with AI-based visual insights reduces production errors and saves time and money. With minimal investment, QA personnel can set up a cloud-connected smartphone to monitor production processes from virtually any location. By applying machine learning algorithms, manufacturers can detect errors immediately rather than later, when repair work is more expensive. The concepts and technologies of Industry 4.0 can be applied to all types of industrial companies, including discrete and process manufacturing companies, as well as to the oil, mining and other industrial segments.

We talked about machine learning before, but what exactly is machine learning? It is a data analysis method that automates the construction of analytical models. Basically it is based on

the idea that systems (machines) can learn from available data, so as to identify models independently and make decisions with minimal human intervention. Since machine learning software needs large amounts of data to "learn", it is a technology that fits perfectly with the theme of Industry 4.0 which. Indeed the latter, as we have seen, leads to the creation of enormous volumes of data. In this sense, a machine learning solution - for example - is able to use the data flows available to teach its algorithm what to expect from the production machines it is monitoring. In this way the software can make the most appropriate decisions at a given time and under certain circumstances, without the need for a new physical rewriting of code by a "human" programmer. Leveraging machine learning models in many cases can avoid physical modeling of a system which is usually a very onerous operation because it requires considerable knowledge and experience of the domain. Since we are talking about machine learning, that is, learning, it is good to clarify that algorithms must be appropriately "trained" before operating in a real industrial context, thanks to the use of specific training data.

1.2 University-Industry Collaboration (UIC)

The thesis work is in collaboration with the company *Antonio Zamperla S.p.A* which embraces this digital change and wants to implement intelligence in their machines through the use of machine learning. The company in question has been involved in the development and marketing of attractions for amusement parks since 1966. The topic concerns the improvement of the diagnostic and monitoring techniques of the equipment in order to detect possible anomalies. In this scenario it must be pointed out that safety is one of the main cornerstones of the *Zamperla* company. For this reason the machine is equipped with sophisticated systems that guarantee the safety of passengers. The term anomaly is therefore linked to something that deviates from the usual and optimal behavior of the machine thus detection of anomalies is reduced to detecting sub-optimal working conditions. By monitoring the various data acquired by the machine, we would like to detect anomalous behaviors linked to them and also to justify the model's predictions in terms of factors causing the anomaly. In general terms, the costs arising directly and indirectly from support or assistance activities (as well as from any replacement or repair of equipment) necessary to keep customer satisfaction high, they heavily affect the company's profit. Without the use of tools for analysis and prediction, it is not infrequently necessary to send specialized technical personnel to the customers with relative travel, board and lodging costs. Being able to detect and establish the cause of the anomaly from the first moments in which it occurs becomes essential to provide the right service to customer and to

evaluate the type of assistance. In this thesis we also employ the services provided by *Statwolf*, a company whose goal is to make the benefits of complex data science accessible to a wide range of industries that embrace the current digital revolution. With advanced knowledge of data science, and state-of-the-art machine learning techniques, *Statwolf* helps companies navigate and interpret highly complex projects. In the context of this thesis, data are integrated through *Statwolf*'s Platform. The premise of data integration is to make data more freely available and easier to consume. The platform allows the use of some tools for automatic monitoring of the data quality. Moreover, filtering, slicing, aggregating and visualizing the data interactively is very simple since no code is required and operations are efficient. Once the required datasets have been configured within the platform, a direct access via *Statwolf* API allows users to download data in different ways.

In the following, the thesis is structured as follows. Chapter 2 will present the case study, namely the attraction realized by *Zamperla S.p.A* on which intelligence has been implemented with a view to detecting anomalies. A description of the operation of the attraction will be introduced in the latter chapter. Chapter 3 introduces the main signals acquired by the machine which will be subsequently cataloged in terms of their function. Since the data was acquired mainly during the testing phase of the machine, it is necessary to clean up the data to remove unreliable data. Chapter 3 also covers the data cleaning phase. In Chapter 4 the raw data in form of time series are transformed into features that can be used in machine learning algorithms. The features will be analyzed to understand the distribution of their values. To continue with Chapter 5, which will cover some basic theoretical background of anomaly detection in machine learning and will introduce the anomaly detection method adopted in the case study. Chapter 6 covers some of the concepts related to the *interpretability* of a model and introduces the related methods adopted. The motivation that led to the use of interpretability is that detecting an anomaly is becoming no longer enough and providing a reason why a cycle has been labeled as anomalous is getting more and more importance. In Chapter 7 some results obtained by applying the proposed approach to the data acquired by the machine are illustrated. The thesis work is concluded with Chapter 8 which will discuss the results with some considerations.

2
Problem Overview

This chapter is divided into two main sections: Section 2.1, Section 2.2.
Section 2.1 aims to present the case study in question, namely the *NebulaZ* which is an attraction of the entertainment industry *Antonio Zamperla S.p.A.* The same section describes how the attraction works in order to have a more intuitive and complete overview of the role of its components. To conclude, in Section 2.2 we want to exhibit the approach used to detect anomalies in the operation of the machine and why the role of interpretability is crucial.

2.1 Case Study: NebulaZ Machine

2.1.1 General Operation

In this work *NebulaZ* is the vehicle created by *Zamperla* which is used as a case study to identify sub-optimal working conditions. *NebulaZ* is a family ride in which four arms rotate in fast intermeshing orbits. Eight gondolas at the end of the arms allow seating for four passengers each. The gondolas always remain upright so that in this way riders are never upside down. The central tower rotates around itself while the arms swing about horizontal axes in a circular motion to let riders catch air as they fly over the top of the ride. The arms are synchronized by a redundant central gearing system guaranteeing their intertwining. Figure 2.1 provides an illustration of the machine under consideration.

The activity of the attraction begins with the passenger loading phase. During this phase, the gondolas that are already at the ground level of the ride are occupied and then a partial rotation of the arms is carried out to allow the occupation of the gondolas that were previously at the top. As soon as the passenger loading procedure is completed, the central tower rises in altitude by means of a system with an hydraulic pump to allow the complete rotation of the four arms and the consequent play experience. At the end, as soon as the rotation of the four arms is stopped, the central tower is brought back to the home position and the loading procedure is repeated in this case to let the passengers get off. Later we will use the terms central tower or machine center without distinction, and we denote as "home position" the position in which the machine center is on the ground while with "top position" the position in which the machine center is at its maximum height. Moreover, from here on we will refer to the main engine to indicate the motor dedicated to the rotation of the four arms. The entire operation of the attraction and cooperation between all its devices are managed by one or more PLCs (Programmable Logic Controller), capable of issuing the necessary commands in a synchronous manner so that each device works in a cohesive manner.

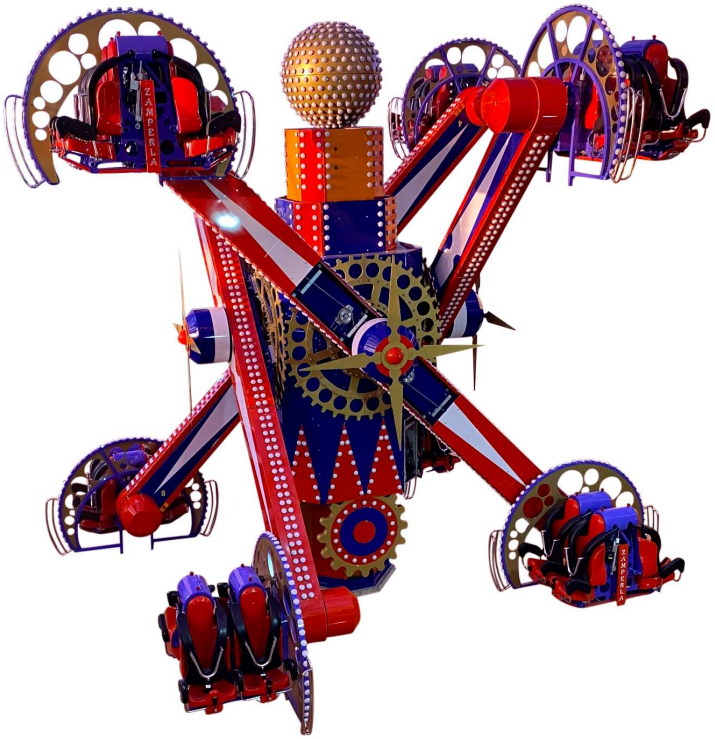


Figure 2.1: Case study: *NebulaZ* ride.

2.1.2 Operativity Conditions

According to a previous case study [1] on a different attraction of *Zamperla* , the same terms that will be used in the following are introduced here:

- Cycle: it is the unit of reference for data acquisition and it refers to the activity of the machine for the time of use of a single user.
- Session/Acquisition: means the collection of data including a set of successive cycles, usually captured during the same day.
- Signal: one of the time series acquired during a cycle which represents one of the measurable quantities acquired by the PLC.

In the following we identify a cycle by means of the session date and the corresponding number assigned to it by the machine acquisition system. Usually, but not necessarily, a progressive number is assigned to successive cycles. For the *NebulaZ* attraction, four main types of operativity conditions during a cycle have been implemented by *Zamperla* group. These configurations differ from each other according to the direction of rotation of the main engine and consequently of the arms with attached gondolas. Clockwise rotation is defined as the direction of rotation such that the sensor used to measure the rotational speed of the main engine acquires a positive quantity. Similarly, the anticlockwise direction of rotation is defined as the direction of rotation such that the sensor used to measure the rotational speed of the main engine acquires a negative quantity. The four types of operation in a cycle can therefore be represented as follows:

- Cycle type 2A: characterized by maintaining the same direction of rotation for the entire duration of the cycle. The gondolas run clockwise.
- Cycle type 2B: characterized by maintaining the same direction of rotation for the entire duration of the cycle. The gondolas run anticlockwise.
- Cycle type 1A: it is characterized by a change of rotation of the main motor in the middle of the cycle. The gondolas initially turn clockwise and then counterclockwise. This type of cycle can be understood as a kind of subsequent composition of types 2A and 2B.
- Cycle type 1B: it is similar to the type 1A cycle, but it can be considered its dual case as the rotation occurs first in an anticlockwise direction and then in a clockwise direction. This type of cycle can be understood as a kind of subsequent composition of types 2B and 2A.

For future reference, it is worth mentioning that the tag that identifies the type of cycle is not yet available in the acquisitions made during this work. For this reason, a basic logic has been implemented to tag cycle types. Hence the fact that the name assigned to cycle types was assigned unilaterally. The choice of the type of cycle to be adopted is up to the customer who installs the attraction. Besides the types of cycles mentioned above, the available data also contain cycles that correspond to the so-called passenger loading and unloading phase. We refer to them as type 3 cycles. With a view to detecting anomalies for the *NebulaZ* ride, it was decided to exclude type 3 cycles from the analysis since they are very short in time and can be considered as a marginal part that does not contribute to the play experience. Furthermore, in any case, given the scarcity of data available regarding type 3 cycles, the aforementioned case could be incorrectly modeled.

In addition to the type of cycle, another parameter that can be tuned by the customer is the duration of the cycle itself, i.e., the duration of the playful experience. In this context, it is necessary to mention that there may be other environmental/operational factors concerning the state of the machine that can influence the overall duration of the cycle.

2.2 Proposed Approach

After a brief description of the machine under study, for a full understanding of the following chapters, it is necessary to summarize the approach used to recognize anomalies in the operation of the machine. In an industrial scenario like the one addressed in this work, supervised settings are rarely available. The unsupervised scenario is the most common in real world applications.

In the *NebulaZ* case study, the data are indeed obtained by observing and collecting, and the labelling procedures are time consuming and typically require domain experts to be involved. Hence, we deal with unlabeled data. The approach used for the detection of anomalies is of a multivariate type. This case take into account that an anomaly can occur as combinations of variables that if considered individually do not denote anomalous behavior. In the unsupervised scenario the learning algorithm lacks a ground truth of what is anomalous and what is not. Given that, the goal of the algorithm is to detect the most abnormal data, assigning to each one an Anomaly Score (AS). Under the assumption that outliers are few, different and easier to be separated from the rest of the data, the proposed method for detecting anomalies is Isolation Fores (IF) [2]. The method can be grouped among Ensemble-based methods and its goal is to quickly model the anomalies by isolating them, rather than spending resources on the modeling of the normal distribution. In the proposed multivariate approach, from the

acquired signals, a finite number of distinctive attributes are extracted. These scalar quantities, called features, are important in order to characterize the behavior of the signals. The features extracted from the signals related to each single cycle are concatenated into a vector which is used as the actual input of the Anomaly Detection (AD) model. Due to the unsupervised settings, the domain knowledge is especially essential in this phase because it is able to validate the choice of features. Although AD algorithms have proved to be useful and effective, their widespread adoption is far from being a reality. This is mainly due to the lack of confidence from the users in AD algorithm outcomes and not immediate association between AD algorithm outcomes and root causes. The concept of interpretability of the model naturally follows from the scenario in which we find ourselves. Adopting the principle of eXplainable Artificial Intelligence (XAI) [3] we would like to determine why a point has been labelled as anomalous to enable root cause analysis. Regarding this, in this work both a well-known state-of-the-art method, namely, SHapley Additive exPlanations(SHAP)[4] and a more recent one, namely, Accelerated Model-agnostic Explanations (AcME)[5] will be adopted.

3

Data Description and Cleaning

Section 3.1 describes the nature of the data acquired by the machine. The acquired data will be grouped here according to their role. The focus will be on the most relevant sensors. When it is informative, the signals will be represented with respect to each of the cycle types introduced in the previous chapter. In relation to this, Section 3.2 reports some qualitative considerations regarding the variability of the data. Section 3.3 covers the cleaning phase. After integrating the data, it is indeed crucial to execute a preliminary processing to remove unreliable data which could adversely affect the AD analysis.

3.1 Data Description

The process of acquiring the raw data by means of the on-board PLC led to the acquisition of about $N_s = 67$ signals. Each of these signals has the form of a univariate* time series and represents a sequence of temporally ordered values assumed by an equipment sensor at certain sampling instants. The signals available for the analysis were acquired in the months of October, November and December 2021. The data comes from 3 different *NebulaZ* attractions. For when it will be appropriate to discriminate between the different rides, reference is made to the following models:

*refer to a single observation over a time period.

- *NebulaZ*_C21111
- *NebulaZ*_C20174
- *NebulaZ*_C21148

As it is possible to see from Figure 3.1, the acquisitions of *NebulaZ*_C21111 are only those related to October, while those of *NebulaZ*_C20174 are related to November. Finally, during December data were acquired from *NebulaZ*_C21148. The tests took place inside the factory hall for the *NebulaZ*_C21111 and *NebulaZ*_C21148, while outside for the *NebulaZ*_C20174.

Figure 3.1: Cycle count over time, (—) *NebulaZ*_C21111, (—) *NebulaZ*_C20174, (—) *NebulaZ*_C21148.

The acquired signals can be analyzed to monitor the behavior of the attraction and can also be grouped in terms of the function for which they are intended. Starting from the domain knowledge of *Zamperla S.p.A.* it is possible to roughly identify the following types of signals:

- Signals relating to physical quantities of the main engine.
- Signals necessary to supervise the correct operation of the machine and the status of the acquisition device.
- Signals which detect the status or transit of some of the machine components. For instance, there are sensors that can detect the transit of the arms.
- Signals that represent commands to perform a specific task. For instance, the command to operate the safety mechanism that locks the machine center in the top position.
- Meteorological data

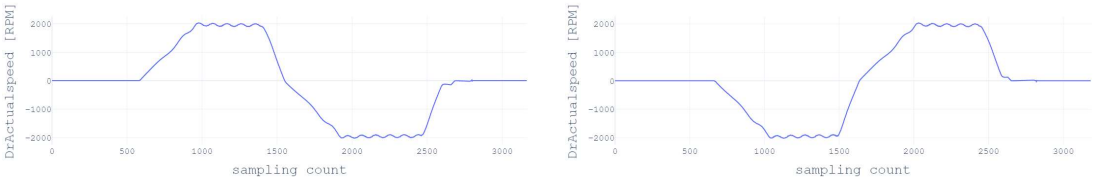
The following types of signals will now be treated individually and the most significant signals for each of them will be listed and displayed (if necessary with respect to each one of the cycle types).

3.1.1 Driver's Signals

Driver's signals mainly concern the signals coming from the equipment sensors relating to the main engine, which is the one dedicated to the rotation of the arms. Among these can be found signals representing consumption in terms of electrical voltage and current and other quantities such as the number of rotations per minute. As mentioned in Subsection 2.1.2, given the differences in operating conditions available to the machine, the signals will therefore be illustrated according to each one of the cycle types. The signals belonging to this category are:

- *DrActualspeedRpm*

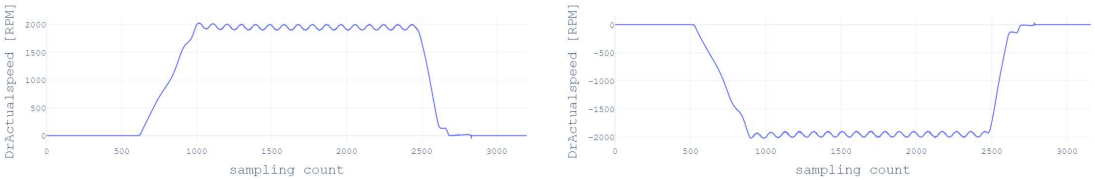
  This signal represents the rotational speed in $[RPM]$ of the main engine. A graphical representation of the latter (for the acquisition time related to a cycle) is available in Figures 3.2, 3.3. Regardless of the type of cycle, it can be noted that this signal is null in the initial portion. The same is valid for all the signals concerning the main engine that will be subsequently treated. The reason for this can be guessed by thinking about the operation of the machine in general terms according to what is reported in Subsection 2.1.1. In the initial fraction, the main motor is not powered as it is necessary to wait for the central tower to rise in altitude. Only when the central tower reaches the top position the four arms are free to rotate without hitting the ground. Consequently, the main engine is then powered.



(a) Cycle 1452 (——) type 1A, October 15th.

(b) Cycle 1250 (——) type 1B, October 18th.

Figure 3.2: *DrActualspeedRpm* [*RPM*], cycle type 1.



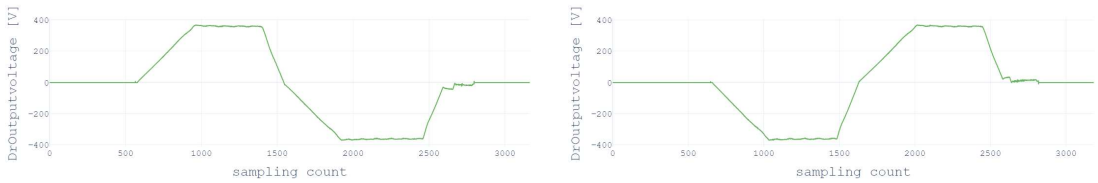
(a) Cycle 1168 (——) type 2A, October 13th.

(b) Cycle 1167 (——) type 2B, October 13th.

Figure 3.3: *DrActualspeedRpm* [*RPM*], cycle type 2.

- *DrOutputvoltageV*

  This signal represents the power supply voltage in $[V]$ of the main engine. A graphical representation of the latter is available in Figures 3.4, 3.5. It is worth noting that this signal is very similar in terms of waveform to the velocity signal shown above.



(a) Cycle 1452 (——) type 1A, October 15th. **(b)** Cycle 1250 (——) type 1B, October 18th.

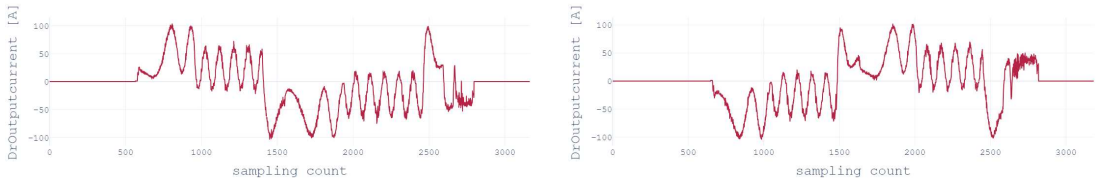
Figure 3.4: *DrOutputvoltageV* [V], cycle type 1.

(a) Cycle 1168 (——) type 2A, October 13th. **(b)** Cycle 1167 (——) type 2B, October 13th.

Figure 3.5: *DrOutputvoltageV* [V], cycle type 2.

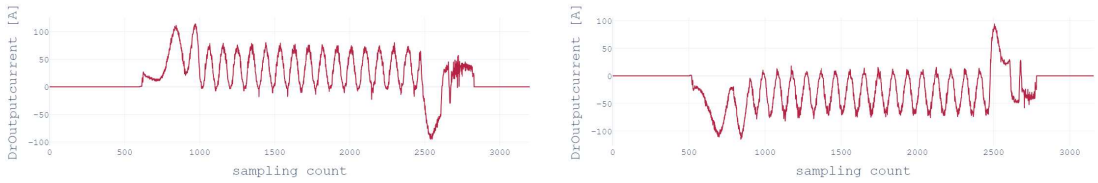
- *DrOutputcurrentA*

  This signal represents the supply current in $[A]$ of the main engine. A graphical representation of the latter during a cycle is available in Figures 3.6, 3.7.



(a) Cycle 1452 (——) type 1A, October 15th. **(b)** Cycle 1250 (——) type 1B, October 18th.

Figure 3.6: *DrOutputcurrentA* [A], cycle type 1.



(a) Cycle 1168 (——) type 2A, October 13th.

(b) Cycle 1167 (——) type 2B, October 13th.

Figure 3.7: *DrOutputcurrentA* [*A*], cycle type 2.

3.1.2 Control and Supervision Signals

These signals are designed to supervise the correct functioning of the devices of the machine. They are of binary type and most of them constantly maintain a logic value, high or low, corresponding to the situation of correct operation of the machine's device under control. A change in value in some of these signals would lead to the occurrence of an unexpected event concerning the monitored section of the system. Due to their little variability during a cycle, only the description in terms of the function for which they are intended is given. Some of these signals are:

- *ManualEstop*

  It is the signal indicating the status of the emergency button. There pressure of the latter would bring the machine into an emergency state with consequent interruption of the cycle.

- *StartTrigger*

  It is the signal that identifies the beginning of an acquisition cycle, thus establishing the start of the signal recording.

- *StopTrigger*

  It is the signal that identifies the end of an acquisition cycle, thus establishing the end of the signal recording.

- *PlcSafetyLocked*

  It is the signal that indicates the activation or not of the safety function of the PLC.

- *PlcSafetySignaturePresent*

  It is the signal indicating the presence of the safety signature generated by the PLC. It is used to track any alterations made to the program.

- *PlcOnline*

  It is a signal consisting of a square wave and allows to check the operating status of the PLC.

- *RideRunning*, *PlcRemRemrunmodeRunning*, *PlcRunmodeRunning*

  These signals indicate, respectively, the running status of the machine, the status of the PLC in remote operating mode and the status of the PLC in active operating mode. The distinction between these last two is represented by access to the editing function, which is possible only in remote operation mode.

- *TotalDispatch*

  It is the signal indicating the total number of cycles performed by the machine from the moment it is installed.

- *ResettableDispatch*

  It is the signal indicating the number of cycles performed by the machine from the moment of the start of an acquisition, therefore this value is reset at the beginning of each session.

3.1.3 Transit and Status Signals

Transit and Status signals are also binary signals and they switch their logic values according to the event they are monitoring and for which they have been used. It is worth illustrating these signals together with the speed signal to have an idea of the phase the machine is in.
Some relevant signals that can be used to detect the transit of the machine's arms are:

- *HomesensCarA*, *HomesensCarB*

  Once the passengers are seated, the machine arms are in a position such that four gondolas are at ground level while the remaining four of the corresponding arms are in the top position. Suppose we denote as type A the four gondolas on the ground while as type B the four gondolas in top position. The former signal assumes a high logic value every time that type A gondolas pass through their starting position, i.e., they make a complete rotation. Similarly, *HomesensCarB* assumes a high logical value every time that type B gondolas pass through the starting point of type A gondolas. In light of their operation, to avoid redundancy, the signals in question are shown with respect to the cycle types 2B and 1B only, respectively, in Figure 3.8 and Figure 3.9.

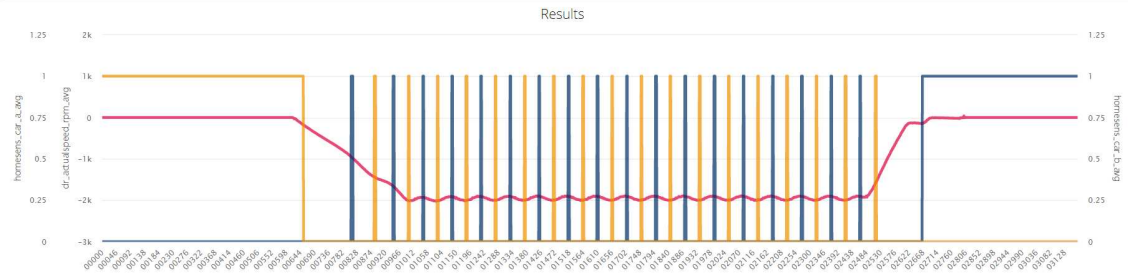


Figure 3.8: *HomesensCarA* (——), *HomesensCarB* (——), *DrActualspeedRpm* (——). Cycle 1171 type 2B, October 13th.

As can be seen, for type 2 cycles the gondolas of the same type that start from the ground end in the top position and vice versa. On the other hand, since the cycles of type 1 can be considered as a kind of composition of two cycles of type 2, it follows that the gondolas of the same type end up in the position from which they started.

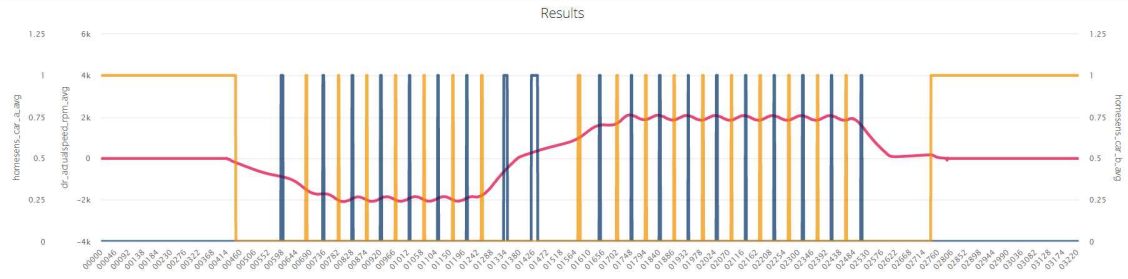


Figure 3.9: *HomesensCarA* (——), *HomesensCarB* (——), *DrActualspeedRpm* (——). Cycle 0006 type 1B, November 19th.

Some of the signals that can be used to retrieve information on the position of the machine center are:

- *HomesensClm*, *TopsensClm*

  The former assumes a high logic value when when the machine center is in the home position, while *TopsensClm* is the signal that assumes a high logic value when the machine center is in the top position. These signals refer to the movement of the machine center, so they do not depend on the type of cycle, i.e., on the direction of rotation of the arms. A graphical representation is given in Figure 3.10.

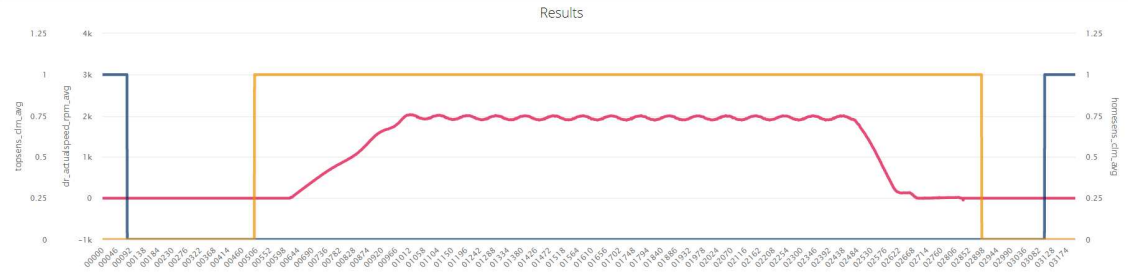


Figure 3.10: *HomesensClm* (——), *TopsensClm* (——), *DrActualspeedRpm* (——). Cycle 1170 type 2A, October 13th.

In the machine, two safety grippers are adopted to ensure that the machine center maintains the top position when appropriate. The status of the safety grippers can be monitored by the following:

- *Actuator1Locked*, *Actuator2Locked*

  These signals assume a high logical value when the corresponding safety gripper is fully inserted.

- *Actuator1Unlocked*, *Actuator2Unlocked*

  These signals assume a high logical value when the corresponding safety gripper is fully extracted. The case where both locked and unlocked signals are at a low logic level can happen. It is enough to consider the situation in which the gripper is not completely inserted or extracted. Figure 3.11 illustrates both "*Locked*" and "*Unlocked*" status signals. The two safety grippers are inserted/extracted almost at the same instant, therefore it is not possible to appreciate differences between the signals of both grippers. For this reason, only the signals related to the first gripper are displayed. Moreover, the signals do not depend on the type of cycle.

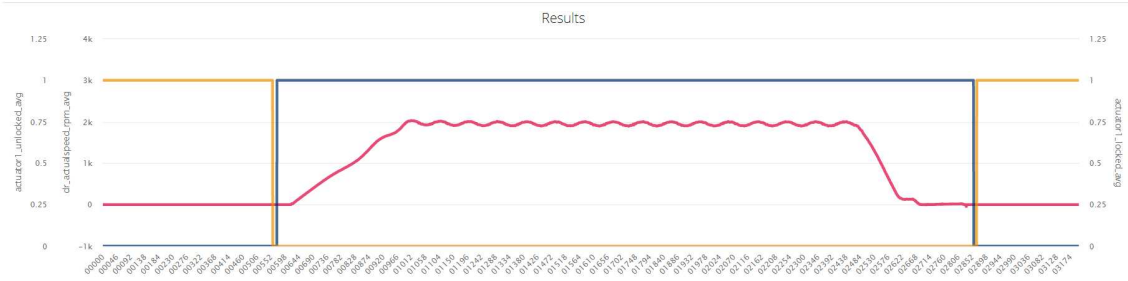


Figure 3.11: *Actuator1Locked* (——), *Actuator1Unlocked* (——), *DrActualspeedRpm* (——). Cycle 1170 type 2A, October 13th.

3.1.4 Commands

In addition to the previous signals, some binary signals that have been acquired have the role of representing the command to perform a certain action. These types of signals can be identified as the cause of an event, while the corresponding status signals can be considered as confirmation that the required action has been taken. Some examples of these so-called commands are:

- *CmdCntQ15901F, cmdCntQ15904F*

  These signals represent the command for inserting the two corresponding safety grippers. As soon as the command is at a high logic level, the procedure for inserting the corresponding safety gripper begins. These commands do not depend on the type of cycle. Again in this case, given the simultaneous action of the two commands, reference is made to the command related to a single gripper. It is also worth illustrating the corresponding status signal which identifies when the gripper is fully inserted. To appreciate the differences between the command and the relative status signal, it is necessary to depict the signals in a restriction, as shown in Figure 3.12. The insertion time of the gripper is the time that elapses between between the corresponding rising edges of status signal and command.

Figure 3.12: *CmdCntQ15901F* (——), *Actuator1Locked* (——), *DrActualspeedRpm* (——). Cycle 1170 type 2A, October 13th.

- *cmdCntQ15902F, cmdCntQ15905F*

  These signals represent the command for extracting the two corresponding safety grippers. Similarly to the previous case, a graphical representation is provided in Figure 3.13.

Figure 3.13: *cmdCntQ15905F* (——), *Actuator2Unlocked* (——), *DrActualspeedRpm* (——). Cycle 1170 type 2A, October 13th.

3.1.5 Meterological Signals

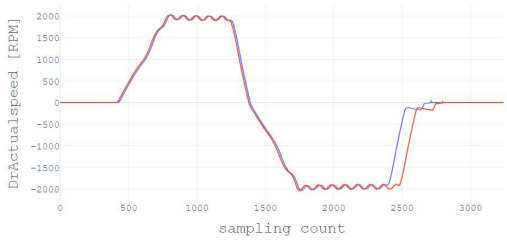
Meteorological signals are signals indicating the meteorological conditions of the environment that have been integrated by means of an online API service [6] by inserting the position[†] of the company. This service gives us information about the meteorological conditions in *Zamperla*'s company location, so the conditions may slightly differ from the one where the machine is placed to carry the tests. Moreover, the service offers meteorological data with hourly sampling. The signals retrieved from the service are the following ones: the mean, maximum and minimum of the temperature, "feels like" temperature, pressure, humidity, wind velocity, wind direction.

3.2 Considerations on the variability of Signals

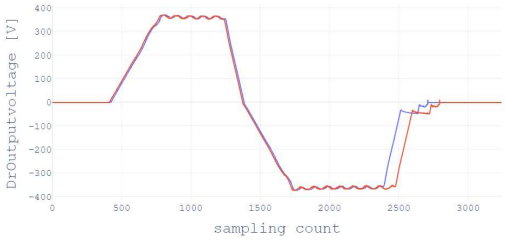
After the description of the available signals, it is interesting to evaluate the influence of some of the environmental/operational factors. In particular, it is significant to analyze, in a qualitative manner, the effects of the gondolas load variation. In this regard, the signals that will now be considered concern the driver's signals as the effect of the load is evident in the waveforms of the latter. In order to do so, we have tried to consider cycles of the same ride in which the load has varied but which are part of the same acquisition. This can avoid weighing the contribution of other environmental/operational factors and therefore allows to discriminate more accurately the effect of the load variation alone. For instance, considering two different acquisition cycles, there could be other variables involved such as the fact that one cycle was performed after lubrication while the other was not. From here on we refer to Arm$i$A, Arm$i$B as a way to denote the load value of the corresponding gondolas for $i = 1 \dots 4$. For instance,

[†]Altavilla Vicentina (VI), Veneto, Italy

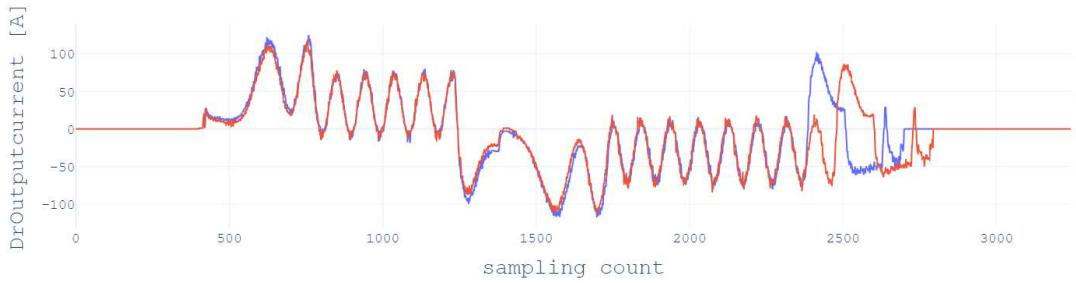
Arm2A denotes the value of the load in the gondolas A of arm 2. Up to now, we have talked about load variation, but this should not be understood as a variation of the overall load value. To visually appreciate, in a qualitative fashion, the variability in all of the drivers' signals, it is indeed more significant to intend the load variation as a variation of the load such that an imbalance is created in the structure. If this is not the case, consider, for example, the cycles 0011 and 0037 depicted in Figure 3.14. These cycles have been acquired on November 23rd and they were performed with an overall load of 900 [*Kg*] and 2100 [*Kg*], respectively. Even though in cycle 0037 the current signal could reveal a higher current draw, it is still not possible to visually appreciate any variability in the speed and voltage signals despite such a difference in the overall load value.



(a) *DrActualspeedRpm* [*RPM*]



(b) *DrOutputvoltageV* [*V*]



(c) *DrOutputcurrentA* [*A*]

Figure 3.14: Driver's signals comparison, Cycle 0011 (——) overall load = 900. Arm1A = 0, Arm1B = 0, Arm2A = 300, Arm2B = 0, Arm3A = 300, Arm3B = 300, Arm4A = 0, Arm4B = 0. Cycle 0037 (——) overall load = 2100. Arm1A = 300, Arm1B = 300, Arm2A = 300, Arm2B = 300, Arm3A = 300, Arm3B = 300, Arm4A = 300, Arm4B = 0, November 23rd.

With this in mind, i.e., not to associate the load variation as the mere variation of its overall value, the driver's signals will be shown with respect to different load configurations in order to report their variability. The load configurations subject to comparison are listed in Table 3.1. The purpose of these considerations is that being able to determine which parts of the signals vary the most with load is very informative regarding extracting features.

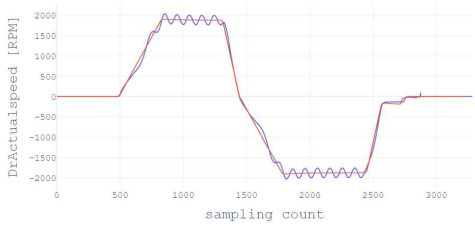
Session · Cycle	Type	Arm1A	Arm1B	Arm2A	Arm2B	Arm3A	Arm3B	Arm4A	Arm4B
2021-12-30 · 0111	1A	0	75	0	0	0	75	0	0
2021-12-30 · 0112	1A	0	0	0	0	0	0	0	0
2021-11-19 · 0008	1B	0	300	0	0	0	300	0	0
2021-11-19 · 0036	1B	300	300	0	0	300	300	0	0
2021-10-25 · 1718	2A	125	125	300	0	300	300	125	125
2021-10-25 · 1770	2A	0	0	0	0	0	0	0	0
2021-11-19 · 0009	2B	0	300	0	0	0	300	0	0
2021-11-19 · 0043	2B	300	300	0	0	300	300	0	0

Table 3.1: Load configurations subject of comparison.

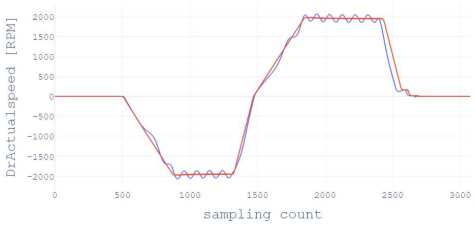
The behaviour of the speed signal *DrActualspeedRpm* is illustrated in Figure 3.15. As it is possible to notice, it seems that the load variation mainly affects the part of the signal where the speed is at its maximum value. In particular, oscillations are formed. In general, it can be intuitively deduced that the amplitude of the oscillations is influenced by the imbalance to which the machine is subjected. It follows that, beyond the cases examined here, cycles which correspond to slightly lower or higher levels of oscillations can be found. The voltage signal *DrOutputvoltageV* is represented in Figure 3.16. The latter seems to be affected by the variability of the load in the same way as the speed signal. The scenario seems different with regards to the current signal *DrOutputcurrentA*. From Figure 3.17 it can be seen that the load variation seems to lead to differences in the signal as a whole.

There are certainly other environmental/operational factors that can affect the driver's signal. For instance, the effect of lubrication in the machine could play a role as after lubrication the influence of friction could be less. For now there are no acquired data on this from the *NebulaZ* machine, nevertheless, in principle, it is still necessary to be aware of the fact that in a real context like this there are many other variables that can influence the signals of interest.

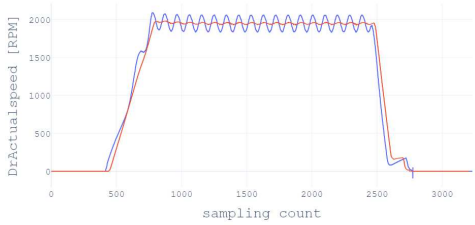
.



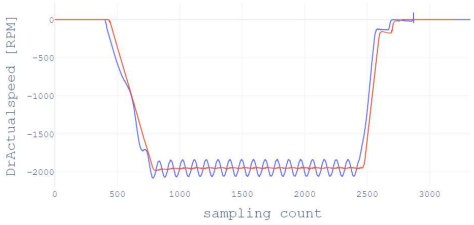
(a) Load distribution: Uniform (——) cycle 0112, Not Uniform (——) cycle 0111.



(b) Load distribution: Uniform (——) cycle 0036, Not Uniform (——) cycle 0008.

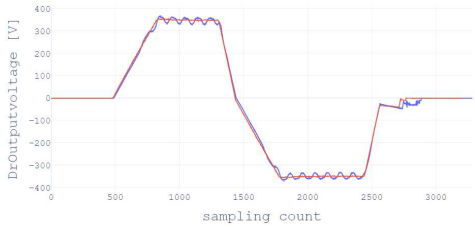


(c) Load distribution: Uniform (——) cycle 1770, Not Uniform (——) cycle 1718.

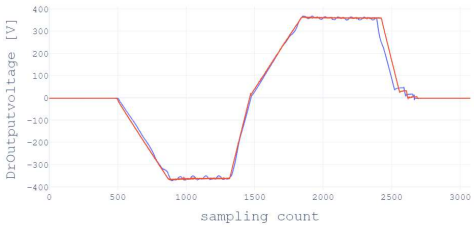


(d) Load distribution: Uniform (——) cycle 0043, Not Uniform (——) cycle 0009.

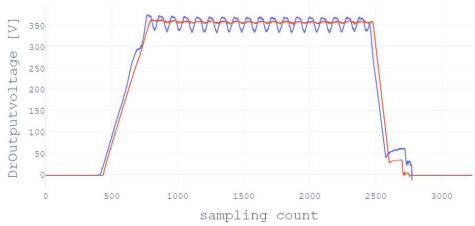
Figure 3.15: Effect of load variation on *DrActualspeedRpm* [*RPM*]



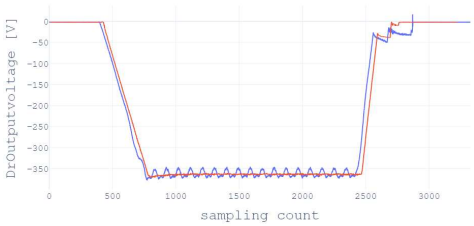
(a) Load distribution: Uniform (——) cycle 0112, Not Uniform (——) cycle 0111.



(b) Load distribution: Uniform (——) cycle 0036, Not Uniform (——) cycle 0008.

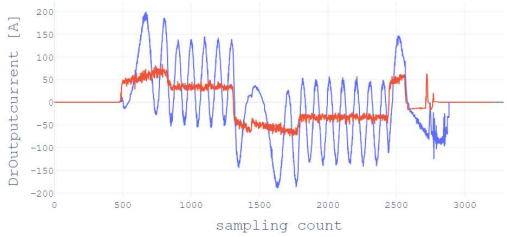


(c) Load distribution: Uniform (——) cycle 1770, Not Uniform (——) cycle 1718.

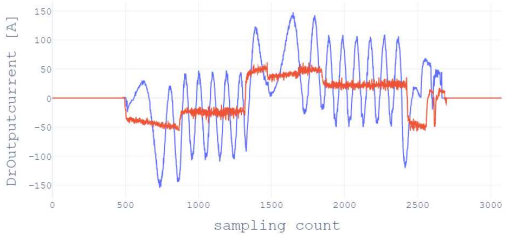


(d) Load distribution: Uniform (——) cycle 0043, Not Uniform (——) cycle 0009.

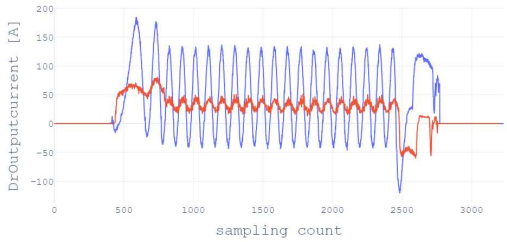
Figure 3.16: Effect of load variation on *DrOutputvoltageV* [*V*]



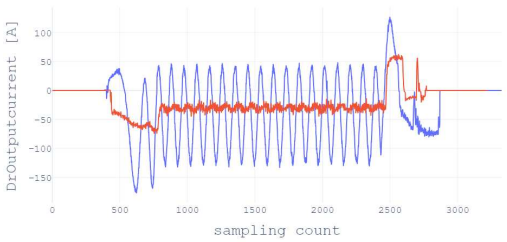
(a) Load distribution: Uniform (——) cycle 0112, Not Uniform (——) cycle 0111.



(b) Load distribution: Uniform (——) cycle 0036, Not Uniform (——) cycle 0008.



(c) Load distribution: Uniform (——) cycle 1770, Not Uniform (——) cycle 1718.



(d) Load distribution: Uniform (——) cycle 0043, Not Uniform (——) cycle 0009.

Figure 3.17: Effect of load variation on *DrOutputcurrentA* [*A*]

3.3 Data Cleaning

Since the data was acquired during the testing phase of the machine, it is necessary to remove corrupted cycles as they could negatively influence the analysis in detecting real anomalies. This preliminary step consists of removing the data that have the following problems:

- Type 3 cycle

  As anticipated in Subsection 2.1.2, this type of cycle can be considered as a marginal part of the activity of the machine, i.e. the passenger loading/unloading phase, and therefore it has not been considered with a view to detecting anomalies.

- Partial Information

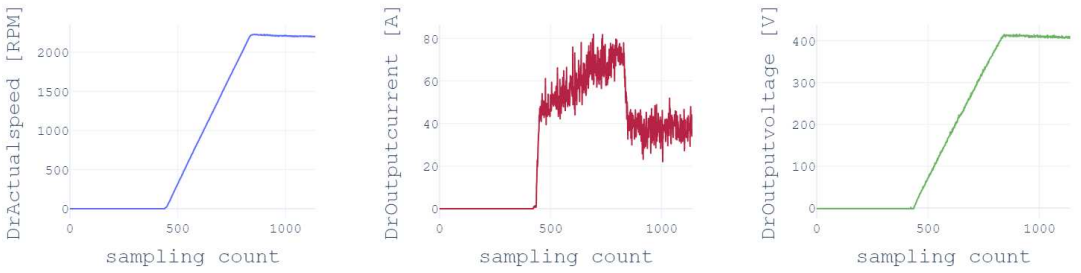
  This include:

  – Missing information on drivers' signals

    Some acquired cycles have been used to verify the partial operation of the machine. For instance, it was checked whether the central tower moved from the home position to the top position, but the rotation of the arms was not performed.

– Aborted cycles

This refers to the interruption of the acquisition of a cycle, probably due to the pressing of the emergency button by an operator or a problem with the acquisition system. Figure 3.18 shows an example of an aborted cycle in terms of the drivers' signals.

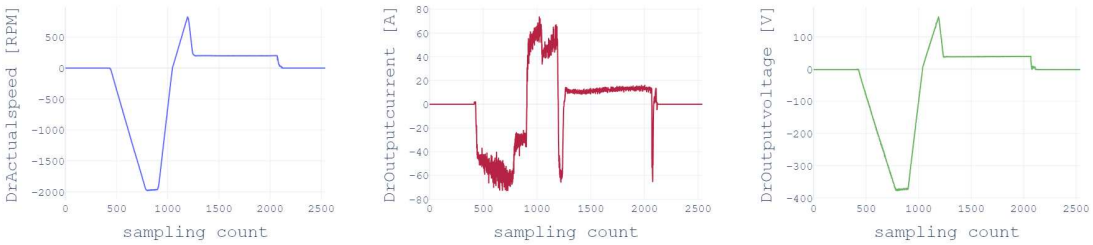


(a) (b) (c)

Figure 3.18: Aborted cycle 0003, December 27th.
(a) *DrActualspeedRpm* (——), **(b)** *DrOutputcurrentA* (——), **(c)** *DrOutputvoltageV* (——).

– Atypical cycle type

These cycles do not reflect the typical operation of the machine in terms of the types of cycles introduced in Subsection 2.1.2. They are the result of testing and therefore have been excluded. An example of atypical cycle type from the point of view of driver's signals is shown in Figure 3.19



(a) (b) (c)

Figure 3.19: Atypical cycle 0022, December 27th.
(a) *DrActualspeedRpm* (——), **(b)** *DrOutputcurrentA* (——), **(c)** *DrOutputvoltageV* (——).

– Missing information on binary signals

In some cycles, Transit and Status signals have switching instants that are not correctly acquired. This lack of information is a problem in the sense that it does not allow us to calculate certain interesting time intervals. For example, in Figure 3.20,

it is not possible to calculate the time elapsed by the machine center in transitioning from the home position to the top position. Indeed, there is no information on when the machine center leaves the home position based on transit/status signals.

Figure 3.20: *Actuator1Locked* (——), *Actuator1Unlocked* (——), *TopsensClm* (——), *HomesensClm* (——), *DrActualspeedRpm* (——). Cycle 0026 type 1A, November 19th.

4

Feature Engineering

The feature engineering pipeline is the preprocessing step that transform raw data in form of time series into features that can be used in machine learning algorithms. Basically, all classic machine learning algorithms use some input data to create outputs. This input data comprise features, which are usually in the form of structured columns. Algorithms require features with some specific characteristic to work properly, indeed, regardless of the data or architecture, the quality of features in the dataset bears a strong influence on the quality of the output derived from machine learning algorithms. Here, the need for feature engineering arises.
Summarizing, the intention of feature engineering is to achieve two primary goals:

- Preparing an input dataset that is compatible with and best fits the ML algorithm;
- Improving the performance of ML model.

Among the various processes involved in feature engineering, Section 4.1 will focus on feature extraction, whereas Section 4.2 will cover feature selection.
In addition Section 4.3 will show graphs relating to the features so as to intuitively understand their behavior. This includes information on the distribution of the features in order to provide an immediate visual representation of the values assumed by each of them.

4.1 Feature Extraction

Feature extraction refers to the process of transforming raw data, which are the time series that have been analyzed in Section 3.1, into numerical features that can be processed while trying to preserve the information of the original data set. The purpose of this step is to automatically reduce the volume of data into a more manageable set for modeling, namely, reducing the dimensionality of the dataset. The aim is indeed to represent the information content of every acquired cycle of the *NebulaZ* in terms of a vector of features. It is worth mentioning that automated feature engineering has been available in some machine learning software for a couple of years now. Automated feature engineering extracts useful and meaningful features using a framework that can be applied to any problem. This in theory will increase the efficiency of data scientists by helping them spend more time on other elements of machine learning but to the detriment of full understanding the nature of features. Furthermore, it is also true that feature extraction is a subjective process that requires human intervention and creativity, and the latter approach would neglect advices based on domain knowledge of *Zamperla Group*. In this work the features will be chosen without the use of auxiliary tools to have full control over which aspects to consider more important or not.

4.1.1 Driver's signals related features

The signals selected for the extraction of the majority of the features are driver's signals (Subsection 3.1.1). The reason is that from the domain knowledge they emerged as the most significant, moreover, from the graphical inspection in chapter 3, they have indeed shown a significantly variable waveform over the course of various cycles. Their variability can provide information that could be used to discriminate between cycles. This cannot be said for other signals which, given their binary nature, remain virtually unchanged in terms of waveforms. Apart from this, it would not be significant to insert binary signals to detect sub-optimal conditions as their anomalous behavior, i.e., an abnormal change of logic value, could be detected regardless of multivariate analysis. Some binary signals are still useful for identifying time intervals. In particular, they will be used later as a trigger for delimiting time intervals. These times will be monitored because they are associated with a significant machine phase which adds information to the characterization of a cycle.

The raw data provides information for each session, and each session is composed of different cycles carried out during the day. For each of the cycles, there have been calculated different

features for each of the following selected signals:

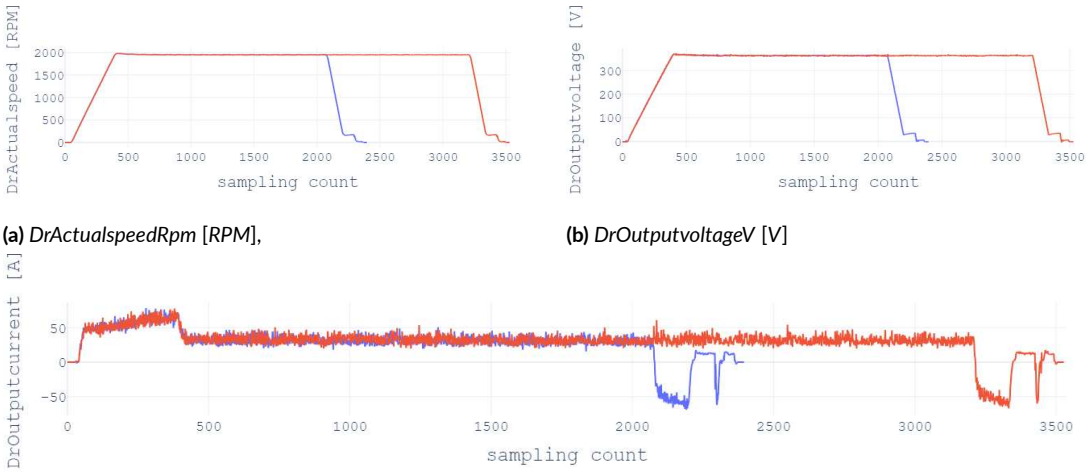
- ☐ *DrActualspeedRpm*
- ☐ *DrOutputcurrentA*
- ☐ *DrActualvoltageV*

The features have been calculated considering a restriction on the signals, that is, the condition that both binary sensors *Actuator1Locked* and *Actuator2Locked* assume a high logical value. As mentioned in Section 3.1, these binary sensors are at a high logic level when both locking pins are fully inserted. Only after this safety measure, the main motor is powered. For this reason, in the features' calculation it would not be informative to take into account the whole signal since the additional acquired values of the affected signal would be always zero.
After analyzing the signal behavior in chapter 2, the extracted features (see Figures 4.2, 4.3) for each of the signals are listed below. The features obtained are independent on the type of cycle adopted by the machine.

- *Max*: represents the maximum value assumed by the signal.
- *Min*: represents the minimum value assumed by the signal.
- *Peak-to-Peak*: is the difference between the maximum positive and the maximum negative amplitudes of the signal.

For future reference, it is worth mentioning that it is necessary to extract some features that are independent on the cycle's duration. Figure 4.1 illustrates a comparison between two cycles in which the only operativity difference is the duration of the cycles themselves. As it is possible to notice, it seems that in this case the only part of the signal that is scaled in terms of duration is the one such that the speed oscillates around its maximum value. For instance, due to this, some statistical properties such as the mean are quite different between these two cycles despite the same operating conditions and being the same cycle type. Accordingly, during the feature extraction, these considerations should be taken into account.

From *Zamperla*'s domain knowledge and from the considerations of Section 3.2 it emerged that the most variable part of the driver's signals (during different cycles) is the interval in which the speed oscillates around its maximum value. For this reason, further features based on this interval have been calculated.



(a) *DrActualspeedRpm* [*RPM*], **(b)** *DrOutputvoltageV* [*V*]

(c) *DrOutputcurrentA* [*A*]

Figure 4.1: Cycles 0011 (——), 0015 (——), type 2A, December 27th. Restriction on *Actuators1Locked=Actuators2Locked*=1.

Up to now, there is no binary signal capable of identifying the aforementioned range, thus the interval has been identified by exploiting the condition that the moving average of the signal is almost constant. In this case, the features collected (see Figures 4.2, 4.3) are:

- *Mean*: represents the mean value of the signal.
- *RMS*: is the square root of the mean square.
- *SD*: represents the standard deviation of the signal.
- *Osc*: is the difference between the maximum positive and the maximum negative amplitudes of the signal.

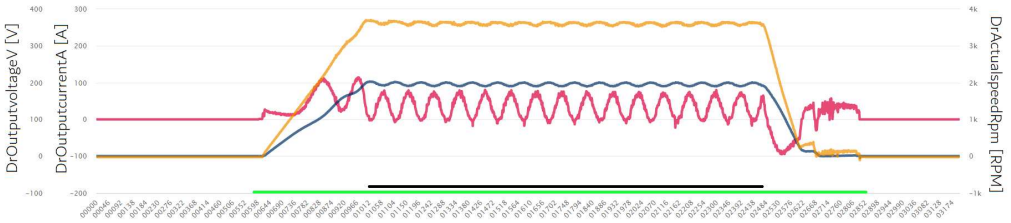


Figure 4.2: *DrActualspeedRpm* (——), *DrOutputcurrentA* (——), *DrOutputvoltageV* (——). Cycle type 2A.
Restriction (——), extracted features: *Max*, *Min*, *Peak-to-Peak*
Restriction (——), extracted features: *Mean*, *RMS*, *SD*, *Osc*

Figure 4.3: *DrActualspeedRpm* (——), *DrOutputcurrentA* (——), *DrOutputvoltageV* (——). Cycle type 1A.
Restriction (——), extracted features: *Max*, *Min*, *Peak-to-Peak*
Restriction (——), extracted features: *Mean*, *RMS*, *SD*, *Osc*

4.1.2 Time and Weather related features

As anticipated, other features have been extracted concerning the time that the machine takes to cover some salient phases. Domain knowledge has suggested that these timings could be variable over the course of several cycles. The following features were then added to highlight the characterization of the cycles.

- *Time before rise*: it concerns the time that elapses from the start of the cycle until the machine center leaves the home position. It is calculated as the difference between the instant in which the binary sensor *HomesensClm* commutates to zero and the cycle start time.
- *Rise time*: it concerns the time that elapses from the moment the machine center leaves the home position to when it reaches the top position. In the perspective of general operation, the raising of the central tower is carried out to allow the fully rotations of the arms. The aforementioned feature is calculated as the difference between the instant in which the binary sensor *TopsensClm* commutates to one and the instant in which the binary sensor *HomesensClm* commutates to zero.
- *Descent time*: it concerns the time that elapses from the moment the machine center leaves the top position to when it returns to the home position. It is calculated as the difference between the moment in which the binary sensor *HomesensClm* commutates to one and the moment in which the binary sensor *TopsensClm* commutates to zero.

Regarding the weather signals, some considerations were made after a meeting with domain experts. In particular the fruitlessness related to the extraction of features concerning some of the weather conditions emerged. First of all, in principle, the machine can be installed in a park outside or inside. Specifically, as already mentioned in Section 3.1, in the available dataset there are data derived from machines that were located outside and inside the factory building. For this reason, and generally because it was considered unreliable and unpredictable, the wind signal was rejected. Ultimately, it was decided to include only the features related to *humidity*

and $temperature$. Indeed, there could be an effect of the latter on the grease of the fifth wheel in the sense that as the temperature increases, the grease becomes less viscous. The experts mentioned that in the first machine cycles, when temperatures are low, higher currents delivered by the driver can be noted. In the context of this thesis, the aforementioned meteorological features have been adopted for all the machines. Although some machines are located in the factory shed, they are still relatively exposed to changes in temperature and humidity as they are not completely isolated from the outside.

4.2 Feature Selection

When building a machine learning model in real-life, it's usually rare that all the features in the dataset are useful to build a model. Adding redundant variables reduces the generalization capability of the model. Furthermore, adding more and more features to a model increases (in general terms) the overall complexity of the model. As per the Law of Parsimony of 'Occam's Razor', the best explanation to a problem is that which involves the fewest possible assumptions. Thus, feature selection becomes a fundamental part of building machine learning models. The feature selection techniques essentially analyze and evaluate the various features to determine which are irrelevant or redundant and can therefore be removed and which ones are more useful for the model and must therefore be prioritized. Feature selection is one essential method for multiple objectives: improving the prediction accuracy by eliminating irrelevant features, accelerating the model training and prediction speed, reducing the monitoring and maintenance workload for feature data pipeline, and providing better model interpretation and diagnosis capability. In the case study the data is obtained by observing and collecting, thus we deal with unlabeled data. It follows that the feature selection process needs to be contextualized to that effect. An important distinction to be made in feature selection is that of supervised and unsupervised methods. The difference has to do with whether features are selected based on the target variable or not. Unsupervised feature selection techniques ignore the target variable, such as methods that remove redundant variables using correlation. Supervised feature selection techniques use the target variable, such as methods that remove irrelevant variables. In general terms, as mentioned in [7]:

> "The goal of feature selection for unsupervised learning is to find the smallest feature subset that best uncovers "interesting natural" groupings (clusters) from data according to the chosen criterion."

Hence, we need to define what "interesting" and "natural" mean. These are usually represented in the form of criterion functions. Recall that our goal is to find the feature subset that best discovers "interesting" groupings from data, a criterion function is what generally allows to assess the cluster quality and therefore to select an optimal feature subset. For instance, a property typically desired among groupings is cluster separation and with Scatter Separability Criterium we can quantify the fact that we are interested in features that can group the data into clusters that are unimodal and separable.

That being said, as for the case study, we mainly relied on domain knowledge to overcome the difficulties in the choice of features given by the unsupervised context. A correlation analysis between features can be helpful to remove redundant variables, but the domain knowledge played a bigger role in understanding whether it is significant to include or exclude features because they can provide feedback, moreover, the domain experts can evaluate if the correlation of the feature is coherent with what they expect from the machine.

4.2.1 Elimination of redundant features

Feature vectors constructed with the procedure described so far in Section 4.1 are found to have dimension equal to $N_{feat} = 26$. However, by virtue of the feature selection procedure, some of these are eliminated in order to obtain the definitive inputs to be provided to the AD algorithm. Some of the extracted features are in fact redundant, therefore, a correlation analysis is used to eliminate them. Regarding correlation analysis, it can help with getting some insights and spot patterns within the dataset. A positive correlation result means that both variables increase in relation to each other, while a negative correlation means that as one variable decreases, the other increases. The correlation matrix contains the correlation coefficients between each variable and thus it can be used to investigate the dependence between multiple features at the same time. Among the several different measures for the degree of correlation in data, we will adopt one of the measures of linear correlation, namely, Pearson Correlation Coefficient (PCC). Given a pair of random variables (X, Y), the PCC, represented by ρ, is the ratio between the covariance of the variables and the product of their standard deviation. We have [8]:

$$\rho_{X,Y} = \frac{cov(X, Y)}{\sigma_X \sigma_Y} \tag{4.1}$$

Of course, in our case we can obtain a practical formula for equation 4.1 by substituting the estimates of the covariance and variances based on samples.
An interpretation of the absolute value of the PCC can be:

- Exactly $1 \rightarrow$ A perfect downhill (or uphill) linear relationship
- $0.70 \rightarrow$ A strong downhill (or uphill) linear relationship
- $0.50 \rightarrow$ A moderate downhill (or uphill) relationship
- $0.30 \rightarrow$ A weak downhill (or uphill) linear relationship
- $0 \rightarrow$ No linear relationship

It was chosen to consider a feature as redundant, and therefore to exclude it, in the case that the absolute value of PCC is greater than or equal to 0.99. This threshold has been voluntarily set very high because, as already mentioned, in the unsupervised context it is not trivial to evaluate the contribution of the features, consequently we wanted to eliminate only the very redundant features. As mentioned in Section 3.1 the data comes from three different rides which have been tested in different environments. In view of the fact that we would like a set of features that can be applied to any other *NebulaZ* and by virtue of wanting to reduce the loss of information, only the features that are redundant in all three cases have been discarded.

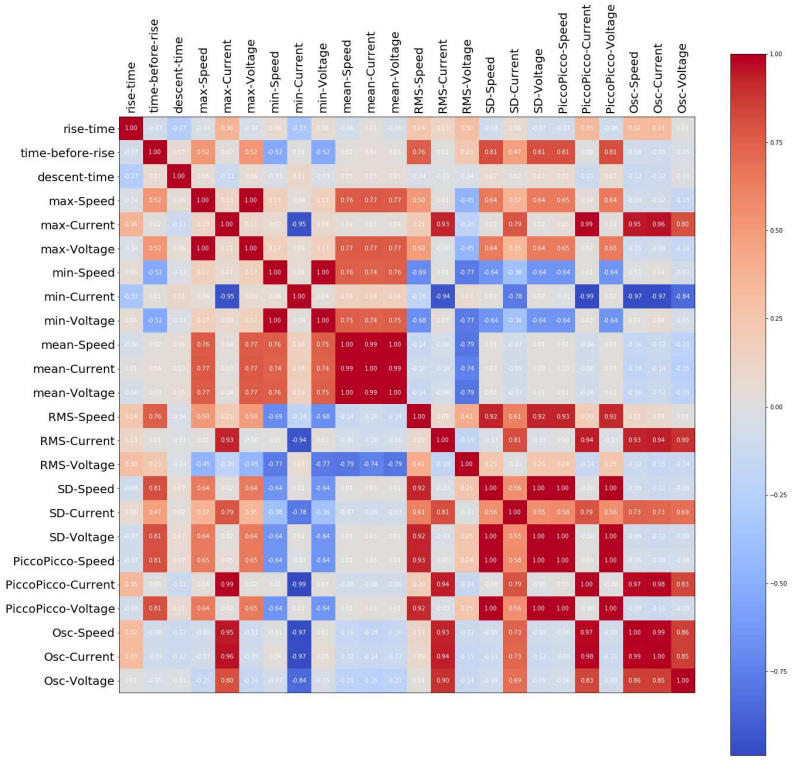


Figure 4.4: Correlation matrix: driver's signals related features and time related features, *NebulaZ*_C21111.

Correlation matrices about driver's signals related features and time related features of *NebulaZ*_C21111, *NebulaZ*_C20174, *NebulaZ*_C21148 are available, respectively, in Figure 4.4, 4.5 and 4.6. Recall Section 3.1, the voltage signal has many similarities in terms of waveform with the speed signal. This redundancy has repercussions in a high correlation between the features of the two signals. In conclusion, according to the chosen threshold, the following features have been excluded: *Max Voltage*, *Min Voltage*, *Mean Voltage*, *SD Voltage*, *Peak-to-Peak Speed*, *Peak-to-Peak Voltage*, *Osc Speed*.

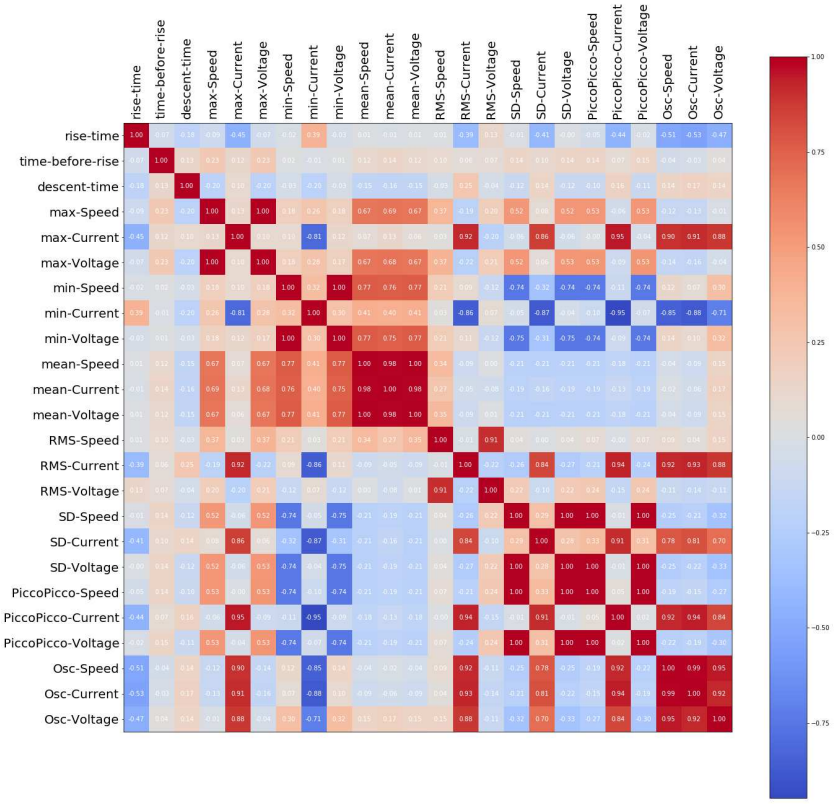


Figure 4.5: Correlation matrix: driver's signals related features, and time related features *NebulaZ*_C20174.

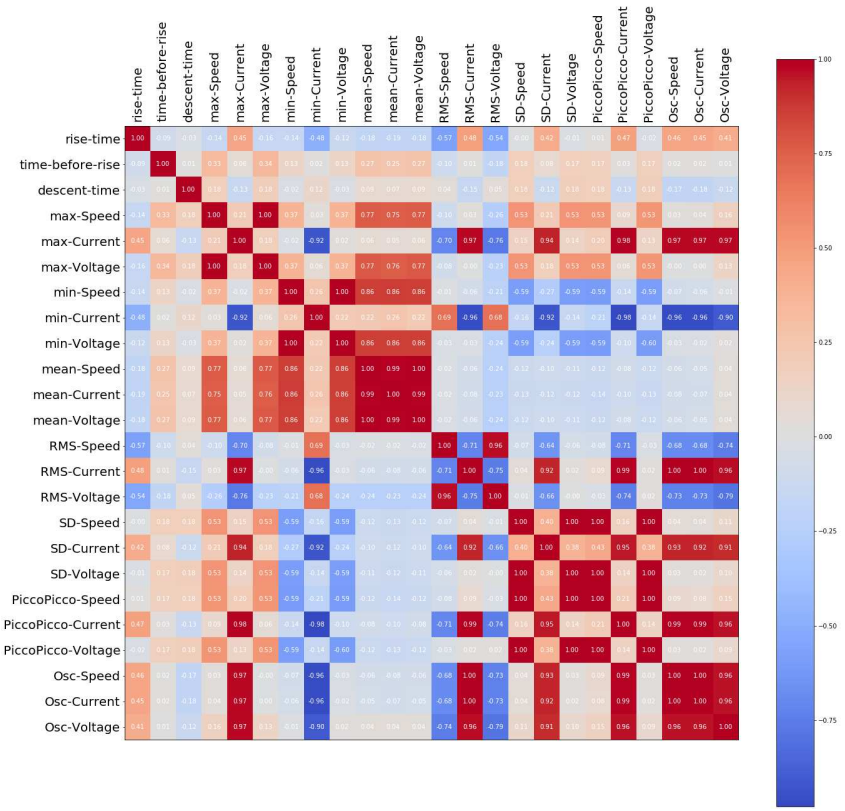


Figure 4.6: Correlation matrix: driver's signals related features, and time related features *NebulaZ*_C21148.

4.3 Features Visualization

At the end of the extraction and subsequent selection of the features, it is important to provide a summary of the quantities obtained to outline a precise scenario for the continuation of the work. The aim of this part is to provide some insights on the features calculated for each cycle in the previous section. This mainly concerns frequency distribution analysis with histograms or possibly other analysis that might be significant in interpreting the behavior of the features. Another covered technique is Principal component Analysis (PCA), which will be introduced below in Subsection 4.3.2. Perhaps the most popular use of principal component analysis is dimensionality reduction, but besides this, we will mainly use it to help visualize data. A final summary including the definitive features is available in Table 4.1. The final dataset consists of 1103 cycles which include all three rides. The feature selection procedure led to a number of features equal to $N_{feat} = 19$.

#	Feature	Signal
(1)	*Max* - Maximum value	*DrOutputcurrentA* - restriction: *ActuatorLocked1=ActuatorLocked2=1*.
(2)	*Min* - Minimum value	⋮
(3)	*Peak-to-Peak* value	
(4)	*Mean*	*DrOutputcurrentA* - restriction: constant Moving Average.
(5)	*RMS* - Root Mean Square	⋮
(6)	*SD* - Standard Deviation	
(7)	*Osc = Max - Min*	
(8)	*Max* - Maximum value	*DrActualspeedRpm* - restriction: *ActuatorLocked1=ActuatorLocked2=1*.
(9)	*Min* - Minimum value	⋮
(10)	*Mean*	*DrActualspeedRpm* - restriction: constant Moving Average.
(11)	*RMS* - Root Mean Square	⋮
(12)	*SD* - Standard Deviation	
(13)	*RMS* - Root Mean Square	*DrOutputvoltageV* - restriction: constant Moving Average.
(14)	*Osc = Max - Min*	⋮
(15)	*Temperature*	Metereological
(16)	*Humidity*	⋮
(17)	*Time before rise*	*HomesensClm*
(18)	*Rise time*	*HomesensClm*, *TopsensClm*
(19)	*Descent time*	⋮

Table 4.1: Final extracted features.

4.3.1 Features Distirbution

Since the data was collected from different machines, it is reasonable to take this difference into account when viewing the data. It is also significant to discriminate between the different types of cycles that can be adopted by the machine to analyze the behavior of the features as a function of the latter. The histograms of the majority of the extracted features are shown below.

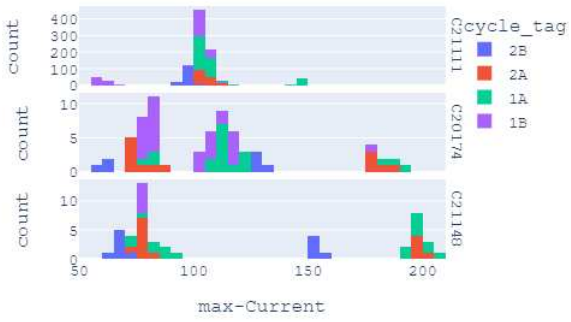


Figure 4.7: (1) *Max* Current

Figure 4.8: (2) *Min* Current

Figure 4.9: (3) *Peak-to-Peak* Current

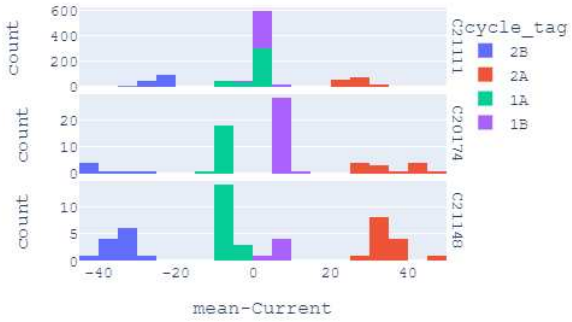


Figure 4.10: (4) *Mean* Current

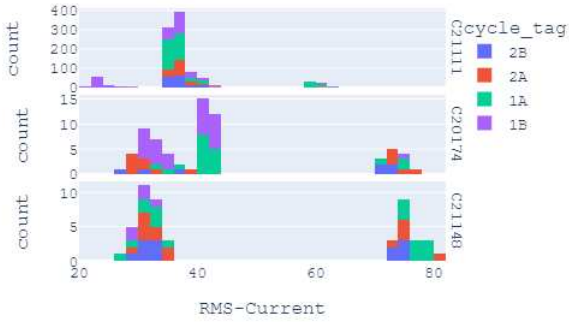


Figure 4.11: (5) *RMS* Current

Figure 4.12: (6) *SD* Current

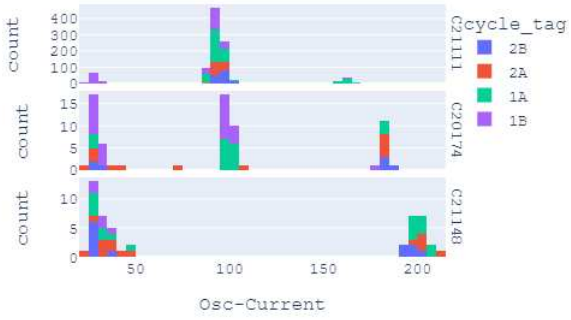


Figure 4.13: (7) *Osc* Current

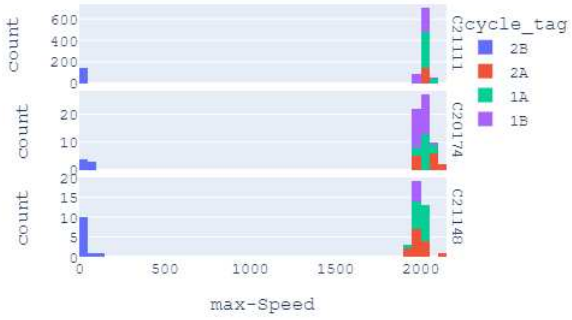


Figure 4.14: (8) *Max* Speed

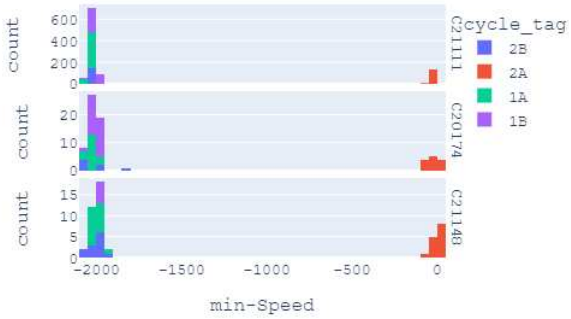


Figure 4.15: (9) *Min* Speed

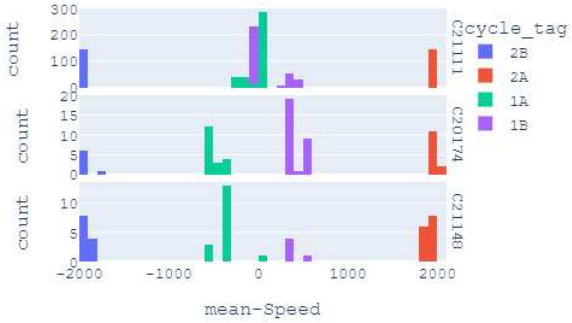


Figure 4.16: (10) *Mean* Speed

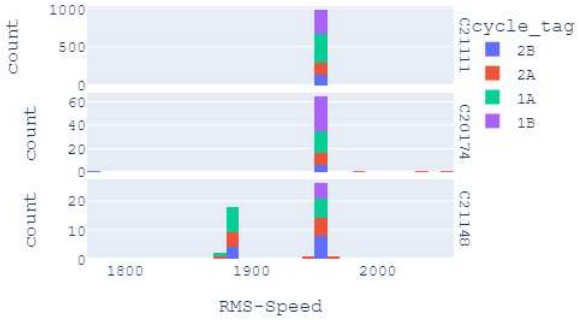


Figure 4.17: (11) *RMS* Speed

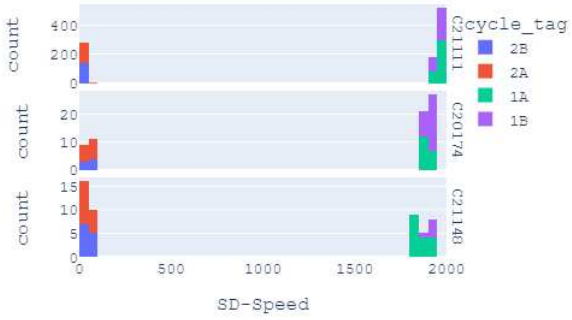


Figure 4.18: (12) *SD* Speed

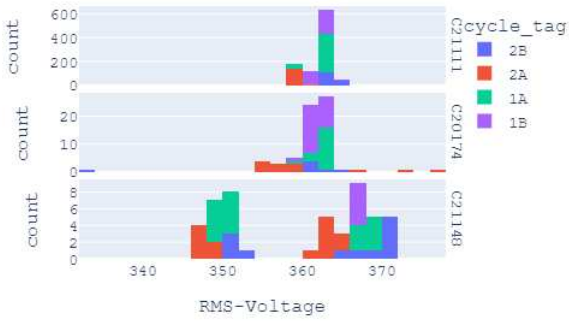


Figure 4.19: (13) *RMS* Voltage

Figure 4.20: (14) *Osc* Voltage

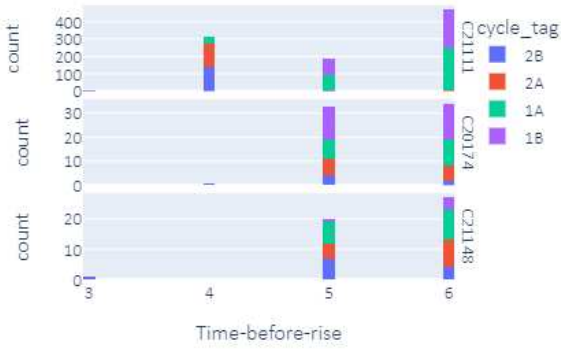


Figure 4.21: (17) *Time before rise*

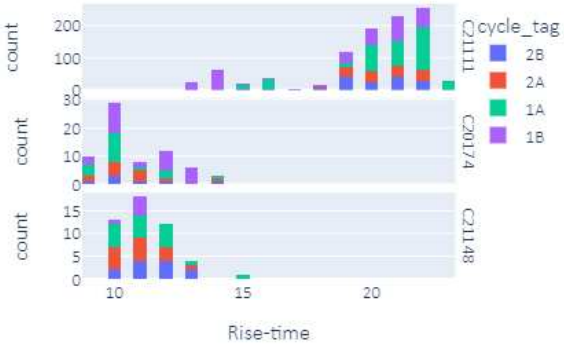


Figure 4.22: (18) *Rise time*

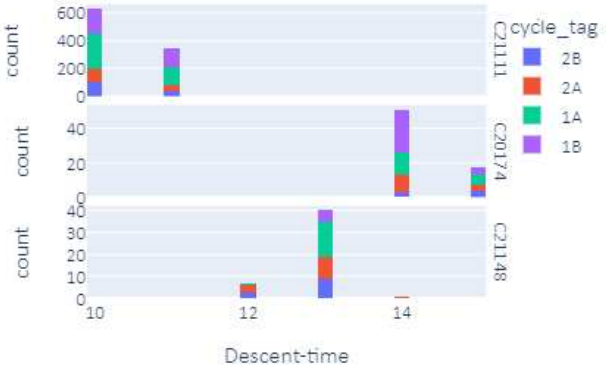


Figure 4.23: (13) *Descent time*

In some of the histograms presented, the count of some values assumed by the features are not perfectly appreciable due to the scarcity of data whose value is within the range. In this case, it is better to refer to the ranges of values constituting the horizontal axis of each histogram to outline the range of variability of features. In conclusion, the trend of *Temperature* and *Humidity* acquired over the course of the cycles is illustrated in Figure 4.24. We remind that the weather data have been associated with the cycles on an hourly basis.

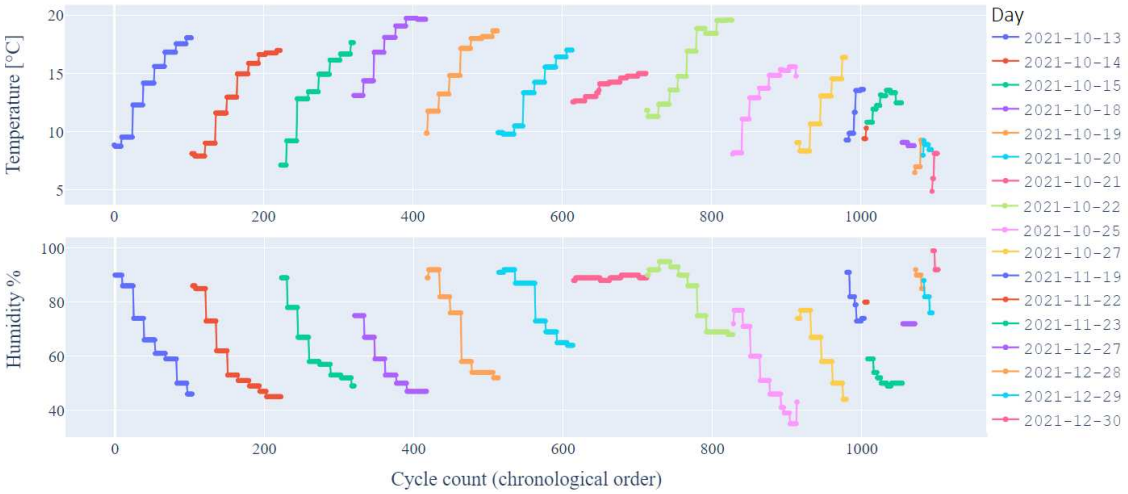


Figure 4.24: (15) *Temperature* and (16) *Humidity* over the course of the acquired cycles.

4.3.2 Dimensionality Reduction: PCA Method

Principal Component Analysis (PCA) is an unsupervised technique for reducing the dimensionality of a dataset, increasing interpretability but at the same time minimizing information loss. It does so by creating new uncorrelated variables that successively maximize variance. Finding such new variables, the principal components, reduces to solving an eigenvalue/eigenvector problem. The principal components are obtained as a linear combination of the starting features. The number of these PCs are either equal to or less than the original features present in the dataset. In this work, the use of PCA is related to the visualization of the acquired cycles of the *NebulaZ* in terms of the first two principal components. We denote the latter as PC1 and PC2. For further information on the PCA technique see [9]. Figure 4.25 shows the loadings and the explained variance obtained from the available data. The loadings can be interpreted as the coefficients of the linear combination of the initial variables from which the

principal components are constructed, while the explained variance represents the information explained using a particular principal components.

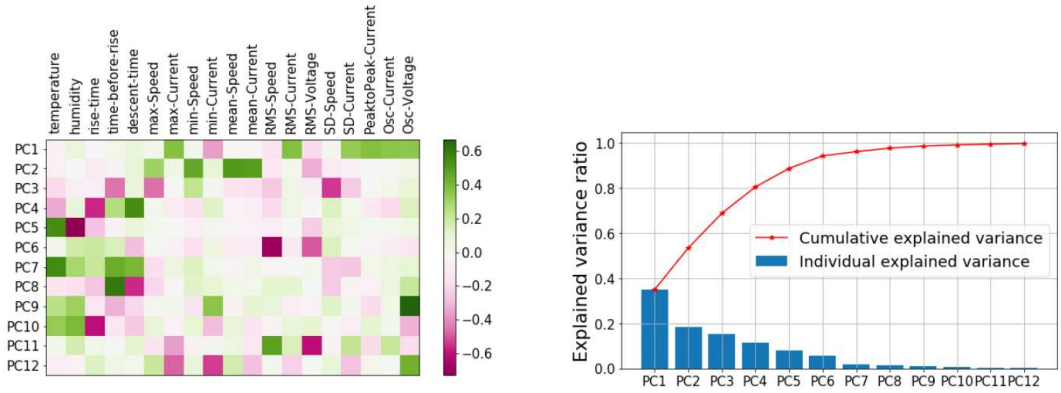


Figure 4.25: PCA: Loadings and Explained Variance

A graphical representation of the cycles by means of the first two main components is provided in Figure 4.26. In the figure, the cycles are characterized by color according to the ride to which they belong. The predominance of blue dots suggests that most of the data comes from the first machine, the *NebulaZ_C21111*. Figure 4.27 provides an illustration in which the cycles are characterized by color according to the type of cycle to which they belong. Based on the loadings in Figure 4.25, the *Mean* Current and *Mean* Speed features play an important role in determining PC2. Moreover, according to Figures 4.10 and 4.16, these features assume very different values depending on the type of cycle. It follows that along PC2 the cycles are kind of divided into three clusters corresponding to the type of cycles 2A, 2B, and 1. As regards the characterization of PC1 it can be seen from Figure 4.25 that the greatest contribution is due to the features *Osc* Current/Voltage, *Min*, *Max*, *RMS*, *SD* and *PeaktoPeak* Current. Given the considerations in Section 3.2, it follows that the aforementioned features are strongly influenced by the variation of the load. The first main component therefore takes into account the variability between cycles due to the different load configurations. In this regards, Figure 4.28 provides an illustration in which the cycles are characterized by color according to the value of the feature *Osc* Current. For the purpose of a final consideration, consider Figure 4.29. The latter characterizes the cycles according to the overall load value. It can be verified that, according to Section 3.2, the variability in terms of oscillations introduced in the driver's signals is not necessarily proportional to the overall load value of the machine.

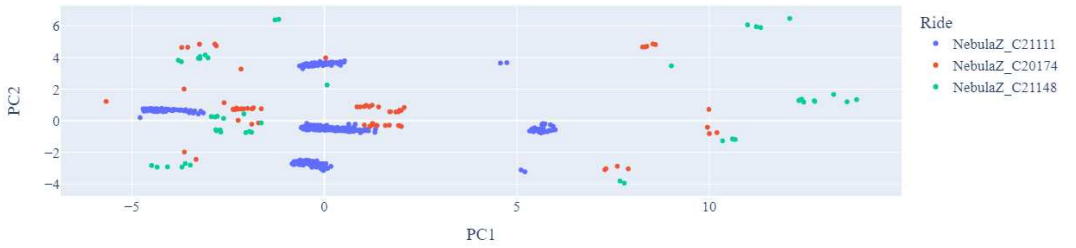


Figure 4.26: PCA: Ride.

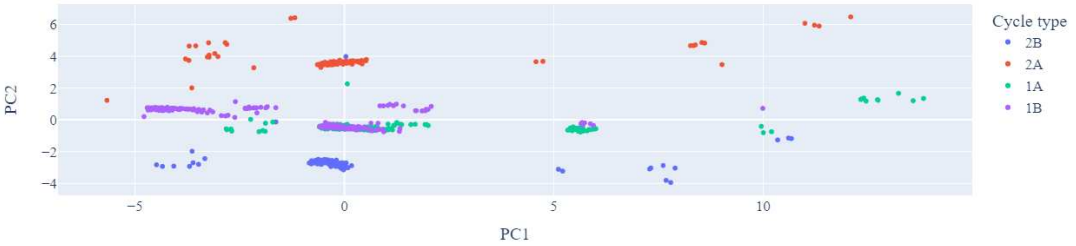


Figure 4.27: PCA: Cycle type.

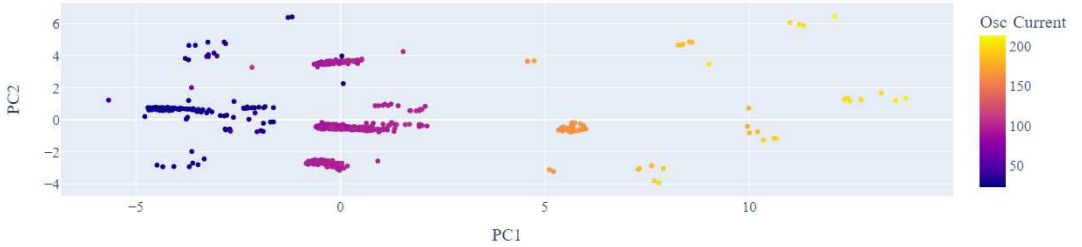


Figure 4.28: PCA: *Osc* Current.

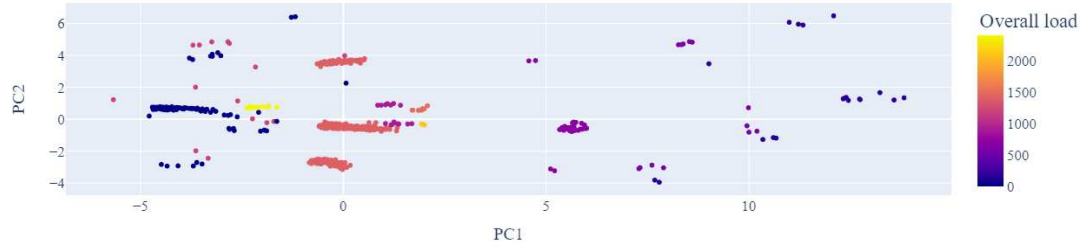


Figure 4.29: PCA: Overall load.

5

Anomaly Detection: Multivariate Approach

This chapter includes Section 5.1 and Section 5.2.
In Section 5.1 some basic Anomaly Detection (AD) concepts are covered. Here it is chosen not to list any particular state-of-the-art AD methods except what will be used in the case study. The reason for this is that Section 5.1 is not a review of all the AD techniques, rather we want to provide the minimum concepts to understand the choices made for the *NebulaZ* vehicle of *Antonio Zamperla S.p.A.*
Section 5.2 is dedicated to explain the anomaly detection method that has been adopted, namely, the Isolation Forest [2]. The main concepts and properties of the Isolation Forest's algorithm will be addressed.

5.1 Anomaly Detection Problem

When providing an anomaly definition, we often find ourselves in a condition that implies the possibility of different interpretations of the term [10]. This ambiguity may be due to the area of application in which we focus. Areas of different domains contain different types of anomalies and it often happens that an anomaly in one domain may be a normal behavior in another and vice versa. For example, in healthcare, a small deviation from normal conditions (think of variations inbody temperature) may indicate an anomaly, where in financial markets the more or less significant fluctuations in prices can be considered normal. Despite the vagueness and complexity in defining an anomaly (outlier), the latter can generally be described as:

> "Observation which deviates so much from other observations as to arouse suspicion it was generated by a different mechanism".[11]

Even if the definition does not mention the numerousness, it can be understood that an anomaly must rarely occur, otherwise it could be traced back to normal behavior. Preliminarily, if desired to carry out an attempt to reveal anomalies, it seems to be necessary to define one region representing normal behavior and declare as an outlier each observation that differs from it. Clearly, several factors make this approach delicate and complex. Define a region as normal and such that it contains every possible ordinary behavior is evidently difficult. The real world is indeed not stationary. The boundary between normal and anomalous is never clear and predeterminable with certainty. Patterns indicating normal behavior or trends are usually constantly evolving, therefore the current representation of normal behavior may no longer be valid or reliable in the future [12]. Detecting outliers translates into significant actionable information in a wide variety of applications. Some examples from the literature may be fraud detection [13, 14, 15, 16], intrusion detection in cybersecurity [17], health diagnosis [18], defect detection from behavioral patterns of industrial machines [19] and many others [20, 21]. There are quite a lot of different issues that cause outliers. Some of the most common causes of outliers are a result of mechanical failure, changes in system behavior, fraudulent behavior, malicious activity, human error, instrument error, setup error, sampling errors, data entry error, and environmental changes. For instance, outliers from data errors are usually the result of human error, such as in data collection. Before listing the main categories of methods to be used to identify outliers, a first distinction can be made with respect to the nature of the data under consideration [21].

- Unsupervised AD. In the unsupervised approach, the available data are not labeled and it is not required knowledge or classification of normal behavior or anomalies that may arise. In this approach, it is assumed that normal data are much more frequent than anomalies. These techniques are the most applicable because to be performed they only need a database of data to be analyzed. In fact, it is not always easy to obtain labeled data as the labeling is usually done manually by an expert operator and therefore the effort to obtain a complete dataset is considerable. It is difficult to get a complete overview of all the anomalous behaviors that the system may incur, moreover the behavior of the anomalies is by its nature dynamic and new types of anomalies can always arise.

- Supervised AD. Supervised techniques are based on the assumption that the data are labeled, i.e., that both anomalous and normal data are classified with an appropriate label. This approach has the advantage of being able to exploit the a priori information of

anomalies during the construction phase of the classifier. When we have a new data, it is compared through statistical techniques with the two classes (normal data and anomalies) to determine which one it belongs to. The problems that arise with the use of this approach are of two types. The first is related to the difference in the number of samples available for the two classes; typically there are more normal than anomalous data available. The second concerns the difficulty of obtaining labels for anomalies and the difficulty of obtaining the dataset since anomalous data are rare.

- Semi-supervised AD. In the semi-supervised case, it is assumed that only the labels of normal data are available and that there is no information available on the type of anomalies that may arise. The fact that this technique does not require knowledge of the anomalies that may arise makes it more easily usable in practical applications than supervised techniques. The typical approach used for these kinds of techniques is to build a model for normal behavior and use the information obtained from normal data to recognize anomalies.

Another distinction of AD techniques based on the number of features being examined can be as follows:

- Univariate techniques look for anomalies in each individual metric or feature. Univariate methods are simpler, so they are easier to scale to many metrics and large datasets.
- Multivariate techniques consider two or more features. This case takes into account that an anomaly can occur as combinations of variables that, if considered individually, do not denote anomalous behavior. Usually, this case is of difficult interpretation because all the metrics are inputs that generate a single output from the anomaly detection system.

Finally, it is possible to categorize the outlier identification methods into the following [20]:

- Ensemble-based methods. Ensemble methods focus on the idea of combining the results of dissimilar models to produce more robust models to detect outliers efficiently. Since it will be used in the case study, it is worth mentioning the Isolation Forest method [2].
- Statistical-based methods. In statistical-based AD methods, the data points are sometimes modeled using a stochastic distribution, and some data points can be labeled as outliers depending on the relationship with the distribution model. These methods are usually classified into two main groups - the parametric and non-parametric methods. Some of the methods adopted for outlier detection are the Gaussian Mixture model, Regression model, Kernel Density Estimation Methods, histogram and other statistical tests.

- Distance-based methods. The underlying principle of distance-based detection algorithms focuses on the computation of the distance between observations. A data point that is at a far distance from its nearest neighbor is regarded as an outlier.

- Density-based methods. The core principle of the density-based outlier detection methods is that an outlier can be found in a low-density region, whereas non-outliers (inliers) are assumed to appear in dense neighborhoods. The objects that differ considerably from their nearest neighbors, i.e., those that occur far from their nearest neighbors, are flagged and always treated as outliers.

- Clustering-based methods. Clustering-based techniques generally rely on the use of clustering methods to describe the behavior of the data. To do this, the smaller-sized clusters that comprise significantly fewer data points than other clusters are labeled as outliers. It is important to note that the clustering methods are different from the outlier detection process. The main aim of clustering methods is to recognize the clusters, while outlier detection is to detect outliers.

- Model-based methods. It refers to the adoption of models that learn the characterization of a normal behavior and consequently label as anomalous the data that deviate from this representation. This category of methods, although much more general, is partially superimposed on the statistical methods. For instance, deep learning techniques for the purpose of AD are part of it. Deep Anomaly Detection (DAD) methods can be based on supervised, semi-supervised, and unsupervised approaches. These techniques learn hierarchical discriminative features from data. In the context of unsupervised DAD models, it is worth mentioning autoencoders since they play a central role.

Among the various categories of algorithms available, it is necessary to keep in mind some of the possible limitations that may arise. First of all, even though some density-based methods are shown to have improved performance, they are in general computationally expensive. They are sensitive to parameter settings such as in determining the size of the neighbors. They need to cautiously take into consideration several factors, which consequently results in expensive computations. Statistical-based approaches, due to their dependency and the assumptions of a distribution model in parametric models, produce results that are mostly unreliable for practical situations and applications due to the lack of preceding knowledge regarding the underlying distribution. Distance-based methods share some similar drawbacks as statistical and density-based approaches in terms of high dimensional space, as their performance declines due to the curse of dimensionality. In clustering settings, outliers are binary; that is, there is no quantitative indication of the object's outlierness. Regarding ensemble techniques, difficulties in evaluating the features of the ensembles arise.

To conclude this subsection, it is worth mentioning the output of the anomaly detection method, that is, the way in which anomalies are detected. Typically, the outputs are of two types:

- Labels. It provides a binary output by assigning a label "normal / anomaly" to each data to indicate if it is an outlier or not.
- Scores. These techniques assign a weight that indicates how much a given data is an outlier. The higher the weight, the more abnormal the data. The weight can be calculated through considerations on the sparsity of the region, considerations on the distances to the neighbors, or the match with a certain distribution of data.

Score-based AD techniques allow the analyst to use a domain-specific threshold to select the most relevant anomalies. Techniques that provide binary labels to test instances do not directly allow analysts to make such a choice, although this can be indirectly controlled through parameter choices within each technique.

5.2 Anomaly Detection: Isolation Forest

In the case study, it has been chosen an algorithm that is able to work without particular assumptions in an unsupervised scenario and multidimensional data with possibly many irrelevant attributes. In particular, we deal with the isolation forest (IF) method. Thanks to its linear time complexity, low memory requirements and capacity in handling high volume databases, IF is highly desirable for real-life applications. Some basic concepts on the functioning of the algorithm will now be addressed in order to better understand the choices made by the latter as soon as the results are presented. For further information, please refer to the original article [2].

5.2.1 Description of the Algorithm

Isolation Forest (IF) [2], similar to Random Forest, is built based on decision trees. Since there are no predefined labels here, it is an unsupervised model. The IF is founded on the fact that anomalies are data points that are few and different. The innovation of this algorithm is that it isolates anomalies rather than profiling normal points. In an isolation forest, randomly subsampled data are processed in a tree structure based on randomly selected features. The samples that travel deeper into the tree are less likely to be anomalies as they require more cuts to isolate them. Similarly, the samples which end up in shorter branches indicate anomalies as it was

easier for the tree to separate them from other observations. As mentioned earlier, Isolation Forests are nothing but an ensemble of binary decision trees. And each tree in an Isolation Forest is called an Isolation Tree (iTree). The algorithm starts with the training of the data by generating Isolation Trees. The training phase can be summarized as follows.

1. Given a dataset, a random sub-sample of the data is selected and assigned to the root node of an iTree.
2. Branching of the tree starts by selecting a random feature (in our case from the set of all N_{feat} features) and a random threshold (any value in the range of minimum and maximum values of the selected feature).
3. If the value of a data point is less than the selected threshold, it goes to the left branch else to the right. And thus a node is split into left and right branches.
4. This process from step 2 is continued recursively till each data point is completely isolated or till max depth, if defined, is reached.
5. The above steps are repeated to construct random Isolation Trees.

After creating an ensemble of iTrees (Isolation Forest), model training is complete. During scoring a single path length $h(x)$ is derived by counting the number of edges e from the root node to a terminating node as instance x traverses through an iTree. When x is terminated at an external node which is associated with more than one data point, the length of the path $h(x)$ is given by e plus an adjustment that accounts for an unbuilt subtree beyond the tree height limit. When $h(x)$ is obtained for each tree in the ensemble, an Anomaly Score (AS) is produced by computing $s(x, n)$ as:

$$s(x, n) = 2^{-\mathbf{E}(h(x))/c(n)} \tag{5.1}$$

In equation (5.1), $\mathbf{E}(h(x))$ denotes the expected path length for each test instance and it is derived by passing instances through each iTree in an iForest, $c(n)$ is the average path length of unsuccessful search in Binary Search Tree and n is the number of external nodes.

5.2.2 Characteristics

It is worth mentioning some characteristics of the IF algorithm. Among these is the fact that the presence of anomalies in the dataset is quite irrelevant to the performance of the algorithm, i.e., it is reasonable to train the model using normal instances only. Moreover, the unique characteristic of isolation trees allows iForest to build a partial model by sub-sampling which incidentally alleviates the effects of swamping and masking. It is because: 1) sub-sampling controls

data size, which helps iForest better isolate examples of anomalies and 2) each isolation tree can be specialized, as each sub-sample includes a different set of anomalies or even no anomaly. As already highlighted, the IF algorithm also has excellent performance in terms of time complexity and memory usage, as it does not need to perform distance or density calculations between data. Furthermore, hyperparameter tuning is often unnecessary as it usually provides good performance when working with predefined hyperparameters.

However, one of the main limitations of the Isolation Forest is related to its interpretability, given that the algorithm provides indications regarding the anomaly levels of the data under examination, but is unable to provide information to enable root cause analysis. In fact, given its random nature, it is very difficult to understand the choices of the algorithm. From this scenario follows Chapter 6 on eXplainable Artifical Intelligence (XAI) which aims to understand why a point has been labeled as anomalous.

6

eXplainable Artifical Intelligence (XAI)

The goal of this chapter is to give an introduction to interpretability in ML and to introduce the model-agnostic state-of-the-art interpretability methods adopted for the *NebulaZ* ride. The motivation that led to the use of interpretability is that detecting an anomaly is becoming no longer enough and providing a reason why a cycle has been labeled as anomalous is getting more and more importance. In the context of AD the explainability of the model allows to identify the underlying causes of an anomaly so that the most effective solutions can be identified and implemented. Indeed, in this scenario, evaluating feature importance is fundamental to enable Root Cause Analysis (RCA).
In Section 6.1 tha main concepts related to the interpretability/explainability of a model will be discussed. Section 6.2 covers a well-known state-of-the-art method, namely SHapley Additive exPlanations (SHAP) [4], while Section 6.3 deals with a more recent method, namely Accelerated Model-agnostic Explanations (AcME) [5].

6.1 Introduction

As it has been in introduced in Chapter 1 of this work, due to the current digital revolution, the use of Artificial Intelligence (AI) in the modern world continues to grow. As a consequence of this, also the topic of XAI becomes increasingly important. Explainable AI and Interpretable ML are all about making our models more transparent and interpretable, helping us answer important questions such as:

Why should I trust the model? How does the model make predictions?
An intuitive definition of interpretability can be stated as:

> "Interpretability is the degree to which a human can understand the cause of a decision.".[22]

Modern AI or ML methods can be used to build sophisticated models that obtain fantastic prediction performance or classification accuracy in a wide range of challenging domains. However, they typically have a complex, black-box nature. There is no definitive threshold for when a model becomes a black box. Generally, simple models with easy-to-understand structures and a limited number of parameters, such as Linear Regression or Decision Trees, usually can be interpreted without requiring additional explanation algorithms. In contrast, complex models, such as Deep Neural Networks with thousands or even millions of parameters (weights), are considered black boxes because the model's behavior cannot be comprehended, even when one is able to see its structure and weights. Indeed there is a sort of a trade off between the performance of the model and its interpretability. Unfortunately, the superior performance of the more complex models often comes at the cost of model transparency and interpretability. The more complex a model is, the more difficult it is to understand what is important to the model and why it behaves the way it does. For these reasons, improving model transparency and interpretability not only helps us build safer, explainable models, but can also help build confidence and trust in the model and its output in the eyes of technical and non-technical stakeholders. This is especially important in environments where a poorly understood model could take actions that bring financial or reputational risks.

Researchers have developed a lot of different types of model interpretability technics over the years. These technics can be classified according to various criteria [23]. The methods used to provide explanations are either model-specific or model-agnostic:

- Model-specific. Model-specific methods work by inspecting or having access to the model internals. Interpreting regression coefficient weights in a linear model is an example of model-specific method. The main challenge of model-specific interpretability is to come up with models that are simple enough to be easily understood by the audience, while maintaining high predictive accuracy.
- Model-agnostic. Model-agnostic methods work by investigating the relationship between input-output pairs of trained models. They do not depend on the internal structure of the model. These methods are very useful for when we have no theory or other mechanism to interpret what is happening inside the model.

The types of "explanations" can typically be grouped into:

- Global explanations - A global explanation of a ML model details what features are important to the model overall. This can be measured by looking at effect sizes or determining which features have the biggest impact on model accuracy. Global explanations are helpful for finding evidence or rejecting a hypothesis that a particular feature is important.
- Local explanations - A local explanation details how a ML model arrived at a specific prediction.

6.2 SHAP

Shapley values are a concept of the *Cooperative Game Theory* field, whose objective is to measure each player's contribution to the game. Shapley values emerge from the context where m players participate collectively obtaining a reward r which is intended to be fairly distributed at each one of the m players according to the individual contribution. Such contribution is a Shapley value. Among explainability techniques that can provide both global and local interpretation, Shapley Additive Explanations (SHAP) [4], is a method introduced by Lundberg and Lee in 2017 based on Shapley values for interpreting machine learning predictions, both in unsupervised and supervised tasks. In this case, SHAP quantifies the contribution that each feature brings to the prediction made by the model, so the players of this cooperative game are replaced by the features of the ML model and the payoff by the model output itself. Follows that this contribution ϕ_i of feature i is defined as:

$$\phi_i = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(|N| - |S| - 1)!}{|N|!} [f(S \cup \{i\}) - f(S)] \tag{6.1}$$

where $f(S)$ corresponds to the output of the machine learning model to be explained using a set S of features, and N is the complete set of all features.

One innovation that SHAP brings to the table is that the Shapley value explanation is represented as an additive feature attribution method, a linear model. The advantage of this form of explanation is that it is really easy to interpret; we can see the exact contribution and importance of each feature. Lundberg and Lee argued that only SHAP satisfies a set of three desirable properties. In particular, if the feature attributions in the additive explanatory model are chosen to

be the Shapley values of those features, then all three properties are upheld. These properties are:

- *Local Accuracy*

$$f(x) = g(x') = \phi_0 + \sum_{i=1}^{M} \phi_i x'_i, \tag{6.2}$$

where M is the number of simplified input features and ϕ_0 represents the model output with all simplified inputs toggled off. We recall that the simplified input x' is a binary vector that represents whether or not we want to include the contribution of that feature to the overall prediction. When approximating the original model f for a specific input x, local accuracy requires the explanation model to at least match the output of f for the simplified input x'. In other words, if the input x and the simplified input x' are roughly the same, then the actual model f and the explanation model g should produce roughly the same output. Recall that by hypothesis ϕ_i are the Shapley values.

- *Missingness*

If the simplified inputs represent feature presence, then missingness requires features missing in the original input to have no impact. Missingness constrains features where $x'_i = 0$ to have no attributed impact.

$$x'_i = 0 \Rightarrow \phi_i = 0 \tag{6.3}$$

- *Consistency*

Consistency states that if a model changes so that some simplified input's contribution increases or stays the same regardless of the other inputs, that input's attribution should not decrease. This is a more important property that essentially says: if we have two (point-wise) models (f, f') and feature i consistently contributes more to the output in f' compared to f, we would want the coefficient of our explanation model for f' to be bigger than f (i.e. $\phi_i(f', x) \geq \phi_i(f, x)$). It is a sensible requirement that allows us to fairly compare different models using the same explainability techniques. We recall that the mapping function $h_x(x') = x$ converts a binary vector of interpretable inputs into the original input space. Note that this mapping function $h_x(\cdot)$ is specific to the data point x.

In particular, let $f_x(z') = f(h_x(z'))$ and $z'\backslash i$ denote setting $z'_i = 0$. For any two models f and f', if

$$f'_x(z') - f'_x(z'\backslash i) \geq f_x(z') - f_x(z'\backslash i) \tag{6.4}$$

for all inputs $z' \in \{0,1\}^M$, then:

$$\phi_i(f', x) \geq \phi_i(f, x) \tag{6.5}$$

The problem with this, however, is that computing Shapley values means you have to sample the coalition values for each possible feature permutation, which in a model explainability setting means we have to evaluate our model that number of times. For example, for 32 Features it is over 17.1 billion. To get around this, Lundberg and Lee devise the Shapley Kernel, a means of approximating Shapley values through much fewer samples. Shapley values can be obtained by tracing the problem back into a case of linear regression [4]. There are a lot of other forms of SHAP that are presented in the original paper, ones that make use of model-specific assumptions to speed up the algorithm and the sampling process, but Kernel SHAP is the one among them that is universal and can be applied to any type of machine learning model. It is worth mentioning TreeSHAP [24], a variant of SHAP for tree-based machine learning models such as decision trees, random forests, and gradient-boosted trees.

Results Visualization

Before introducing AcME, it is reasonable to introduce the visualizations of SHAP results which will be used later. As for global interpretation, the summary plot combines feature importance with feature effects. Each point on the summary plot is a Shapley value for a feature and an instance. The position on the y-axis is determined by the feature and on the x-axis by the Shapley value. The color map, from blue to red, represents the value of the feature from low to high. Overlapping points are jittered in the y-axis direction. In the summary plot, we see first indications of the relationship between the value of a feature and the impact on the prediction. We can also display a bar plot in which the features are sorted by decreasing global importance. The global importance of each feature is calculated as the mean absolute value of that feature overall. As for local interpretation, we will adopt the waterfall plot. The prediction starts from a baseline, that is, the average of all predictions. In this local importance plot, each positive Shapley value is an arrow that increases the prediction, while negative values decrease it. Balancing each other, the arrows point to the actual prediction for the selected observation.

6.3 AcME

In a scenario like the case study, interpretability insights must be provided to the user in a reasonable amount of time to avoid losing money. Most state-of-the-art interpretability approaches require time-consuming procedures for computation that do not allow for 'on-the-fly' operation. Accelerated Model-agnostic Explanations (AcME) [5] is an interpretability approach that, among the various aspects, quickly provides feature importance scores both at the global and the local level. Indeed, similarly to SHAP [4], AcME produces an effective data visualization for global interpretability. Moreover, it brings the same effectiveness to visualization for local interpretability. The importance scores provided by AcME rely on perturbations of the data based on quantiles of the empirical distribution of each feature. These perturbations are performed w.r.t. a reference point in the input space, the baseline vector ($\mathbf{x^b}$).

6.3.1 Global Interpretability

In this case, we consider $\mathbf{x^b}$ as the mean vector $\bar{\mathbf{x}}$, that is, the p-dimensional vector whose components are the mean values of the features. Based on the original paper, the whole procedure for computing the global importance score for the feature j can be summarized as follows.

1. $\mathbf{x^b} = \bar{\mathbf{x}} = \begin{bmatrix} \bar{x}_1, & \dots & \bar{x}_{j-1}, & \bar{x}_j, & \bar{x}_{j+1}, & \dots & \bar{x}_p \end{bmatrix}^T$
2. For each $q \in \{0, 1/(Q-1), 2/(Q-1), \dots, 1\}$, create $\mathbf{z}_{j,q} \in \mathbb{R}^p$ by substituting $\bar{x}_j$ with $x_{j,q}$ i.e., the value of quantile q for the j-th variable:
   $\mathbf{z}_{j,q} = \begin{bmatrix} \bar{x}_1, & \dots & \bar{x}_{j-1}, & x_{j,q}, & \bar{x}_{j+1}, & \dots & \bar{x}_p \end{bmatrix}$
3. Create the variable-quantile matrix for feature $j \in \{1, \dots, p\}$:
   $$\mathbf{Z_j} = \begin{bmatrix} \mathbf{z}_{j,0} \\ \mathbf{z}_{j,1/(Q-1)} \\ \vdots \\ \mathbf{z}_{j,1} \end{bmatrix} = \begin{bmatrix} \bar{x}_1 & \bar{x}_2 & \dots & x_{j,0} & \dots & \bar{x}_p \\ \bar{x}_1 & \bar{x}_2 & \dots & x_{j,1/(Q-1)} & \dots & \bar{x}_p \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \bar{x}_1 & \bar{x}_2 & \dots & x_{j,1} & \dots & \bar{x}_p \end{bmatrix}$$
4. Compute predictions associated with the variable-quantile matrix rows:
   $$\hat{\mathbf{y}}_j = \begin{bmatrix} \hat{y}_{j,0} \\ \hat{y}_{j,1/(Q-1)} \\ \vdots \\ \hat{y}_{j,1} \end{bmatrix} = \begin{bmatrix} f(\mathbf{z}_{j,0}) \\ f(\mathbf{z}_{j,1/(Q-1)}) \\ \vdots \\ f(\mathbf{z}_{j,1}) \end{bmatrix}$$

5. Calculate the standardized effect:
   $\boldsymbol{\Delta}_{j,q} = \frac{\hat{y}_{j,q} - f(\mathbf{x^b})}{\sqrt{var(\hat{\mathbf{y}}_j)}}(max(\hat{\mathbf{y}}_j) - min(\hat{\mathbf{y}}_j))$
6. The global feature importance score for the generic feature j can be computed by averaging the magnitude of standardized effects over the quantiles:
   $I_j = \frac{1}{Q}\sum_{q=1}^{Q} |\boldsymbol{\Delta}_{j,q}|$

AcME has in general lower computational complexity than KernelSHAP. Indeed, AcME only needs to apply the model on $Q \times p$ observations, corresponding to the vectors $\mathbf{z}_{j,q}$. It is worth mentioning the results visualization that AcME provides. For the visualization of global feature importance scores, the authors propose two different kinds of plots. The first is just a bar plot that shows the feature scores in decreasing order computed according to point 6 of the previous list. As for the second visualization on the y axis, the features are sorted in decreasing order of importance according to point 6 of the previous list, while the standardized effects for each element of the variable-quantile matrix are plotted along the x axis. Moreover, the ACME visualization provides a black dashed line, corresponding to the prediction for the base point, to separate positive effects, i.e. those pushing the prediction to higher values, from negative effects, i.e. those pushing the prediction to lower values.

6.3.2 Local Interpretability

When the scope of the analysis is the interpretation of individual predictions, we set the baseline vector $\mathbf{x^b}$ equal to the specific data point to be explained $\mathbf{x}^*$, instead of setting $\mathbf{x^b} = \bar{\mathbf{x}}$ as in the global interpretability scenario. The procedure to get importance scores is similar to the one just reported, but in the local case it only serves to order features in the displayed plot, which is meant to convey a different kind of information. The visualization for local interpretability resembles the global one of AcME. Specifically, in the local case the algorithm does not display standardized effects but the actual predictions associated with the perturbed data points (based on the selected Q quantiles). A dashed line is placed in correspondence with the prediction associated with the original observation $\mathbf{x}^*$, so that it is clear which variables are pushing to increase (or decrease) the prediction.

7

Experiments and Results

Up to now, in the previous chapters, the data and theoretical foundations of the multivariate approach used for AD have been introduced. This chapter is structured in three sections and its aim is to present some results obtained from the data acquired from the *NebulaZ* case study. Section 7.1 plays an introductory role in laying out the main hypotheses and the main considerations with respect to the results. In Section 7.2 a focus is devoted to the evaluation of the interpretability methods in a global fashion in order to certify the consistency of the model. In this context, global explanations are helpful for finding evidence or rejecting a hypothesis that a particular feature is important. Section 7.3 proposes the results obtained by applying the IF algorithm and related interpretability methods (SHAP [4] and AcME [5]) as an actual tool to detect anomalies and enable Root Cause Analysis. In this case some of the data deemed as more anomalous by the IF algorithm are examined locally.

7.1 Introduction

Data Distribution

Before presenting the results obtained, it is necessary to specify some aspects that affect the application of the proposed approach. As already mentioned, the data that make up the final dataset come from different rides. First, the diversity of the rides must be considered. This diversity should not be understood in a general sense of functioning but rather in the change

of the distribution underlying the data. The different rides can be under different operating conditions that are not quantifiable, such as different wear of the components, different type or degree of lubrication, etc. It follows that a cycle of a ride evaluated by an AD method that has been trained for a different ride could be labeled as anomalous despite the same configuration between the two machines in terms of quantifiable operating conditions. Considering the net minority of data coming from *NebulaZ*_C20174 - *NebulaZ*_C21148 and considering what has just been mentioned, it would be that the cycles coming from these last two machines would be more likely to be labeled as anomalous by IF since they require fewer cuts to be isolated. The same considerations can be extended regarding the cycles of the same ride, which represent a clear minority in terms of the operating conditions adopted. For instance, consider the case of very few cycles that have been performed in a certain load configuration which has never been performed until then. Given the considerations on the variability of the signals of Section 3.2, we have that in such cycles the values of the affected features are much more marginal than the general trend. This leads to a greater likelihood of being labeled as an anomaly.

How to handle outliers

Now it is worth making some brief considerations on what to do when an anomaly is detected. So far, in Chapter 5, some considerations have been made regarding what could give rise to outliers and how to identify outliers. It was found that providing a definition of anomaly is a task that requires domain knowledge. Similarly, dealing with outliers is dependent on the application domain. For instance, in cases where the influence of outliers might cause serious issues such as critical environment safety scenarios, or in real-time situations (fraud detection/intrusion detection), an alarm could be set up. While, in a no cause for alarm scenario, in a case like in a population census survey where few people stand out in some features, these outliers can be noted and verified since they are just naturally occurring outliers. Moreover, in the context of AD, we are faced with a trade-off between false positives and false negatives. Depending on the application, it has to be decided how abnormal a point must be for it to be labeled as anomalous. In terms of Anomaly Score, it is a question of defining an anomaly threshold. In applications where errors cannot be accepted, this threshold will tend to favor a higher rate of false positives. In most cases, to answer the question about how to handle outliers, one has to use intuition, analytic argument through some experiments and also thoughtful deliberation before making decisions. In a scenario such as the case study, the role played by the algorithms can find application within a Decision Support System (DSS) to support judgments and actions. A DSS should provide comprehensive information that can be used to help hu-

man operators in decision making. The operator can consider and weigh the importance of the algorithm outcomes for the purpose of an optimal choice in line with the company's objectives.

Experimental Setup

Some practical aspects have been adopted by virtue of data distribution considerations. A first empirical choice led to the exclusion of *NebulaZ*_C20174 - *NebulaZ*_C21148 from the application of the anomaly detection approach. The data available for these two rides are, in fact, too few to outline any reliable consideration. In conclusion, a last consideration can be about the initialization of the IF algorithm in terms of its hyperparameters. In this case, the default values recommended in the original paper [2] have been adopted. These include an isolation forest consisting of 100 iTrees, each obtained using 256 randomly extracted data. The adopted notation foresees that a negative AS is associated with an anomalous point. In contrast, the more positive the AS, the less anomalous the point. This AS assignment is a trivial review of the formulation in Equation 5.1 in terms of an affine transformation. Labels -1, $+1$ are assigned, respectively, to an anomalous cycle or not. It is important to reiterate that the cycles are not a priori labeled as anomalous or not, therefore the evaluation of the results obtained is based on the experience gained during the data analysis, on the results provided by the interpretability method, on the visual comparison with the rest part of the available cycles, and on the comparison with domain knowledge.

7.2 *NebulaZ*_C21111: Consistency of the model

A first significant evaluation can be made by analyzing the evolution of the AS provided by the Isolation Forest: Figure 7.1 shows the trend of the score over the course of the cycles acquired from the *NebulaZ*_C21111. The cycles are sorted in chronological order. The trend of the AS obtained during the cycles of different sessions could be relevant to determine if there is a global pattern for which the cycles are labeled as anomalous. An interesting evaluation of the result provided in Figure 7.1 can be made by observing the tendency of the data related to the first cycles of every acquisition to be more anomalous than the following ones. In simplistic terms, this tendency to sub-optimality could be due to a lower heating state of the machine. Domain experts confirmed the trend of a general increase in energy expenditure in the first few cycles of a session, especially on cold days.

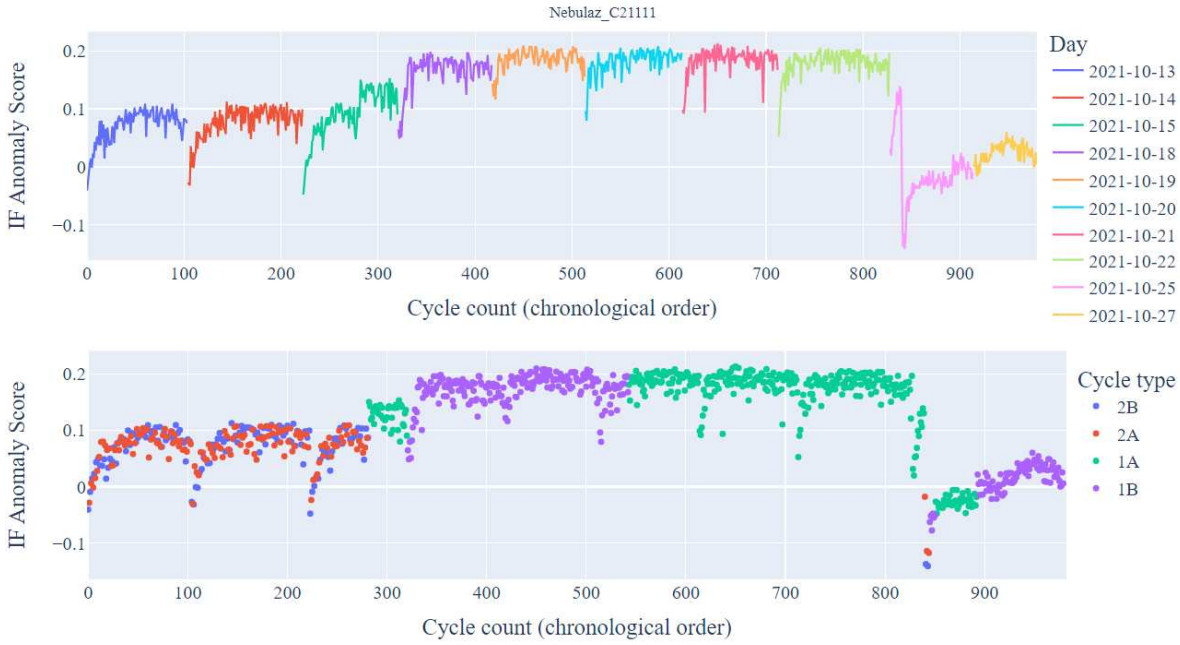


Figure 7.1: IF Anomaly Score, *NebulaZ*_C21111.

As mentioned previously, given the intrinsic nature of the IF, it follows that cycles that represent a clear minority in terms of operating conditions adopted are more likely to be associated with a higher AS. In the specific case, during the acquisitions of October 25th and October 27th, the majority of the cycles were performed with load configurations that had never been adopted until then. For this reason, the load variation can be considered as one of the main aspects that has influenced the AS of these cycles the most. Looking ahead, more data should favor a data set that is homogeneously distributed among the most adopted operating conditions. This should avoid the bias intrinsically introduced by the IF in judging the cycles as anomalous. However in this section, as a first approach, it is interesting to apply and evaluate the interpretability methods to all available data relating to the *NebulaZ*_C21111 machine. This approach can be useful for testing the reliability of the model. Subsequently, in Section 7.3 the focus will be specifically on a fixed operating condition. In the latter case, the IF algorithm and related interpretability methods are proposed as a tool to detect anomalies and enable Root Cause Analysis.

A first global evaluation can be made in terms of the importance of the features provided by the AcME and SHAP methods. Figure 7.2 (←) shows the bar graph of the importance score of the global AcME. The features are ordered from the most important to the ones that

have the least impact on the model. It can be seen that some of the features have a relatively lower importance score. In retrospect, this global classification of features can be interesting for understanding whether the extracted features are significant or whether their presence is irrelevant to the model. This tool can aid in the feature evaluation and selection process in an unsupervised context. Among the features considered less significant by the model are the features related to the weather signals of *Temperature*, *Humidity* and the features related to *Descent time*, *Time before rise*. During the acquisitions made on *NebulaZ_C21111*, the aforementioned features, if compared to the rest, are not very variable between cycles, so they are not very informative in characterizing a cycle. In hindsight, the feature *Time before rise* should be determined by the software choices related to the logic of the PLC. Consequently, this feature should assume scarcely variable values over the course of different cycles and therefore is not considered important in influencing the output of the model. As far as weather features are concerned, it is more than legitimate to expect their little influence as all tests were carried out with stable atmospheric conditions during the various days.

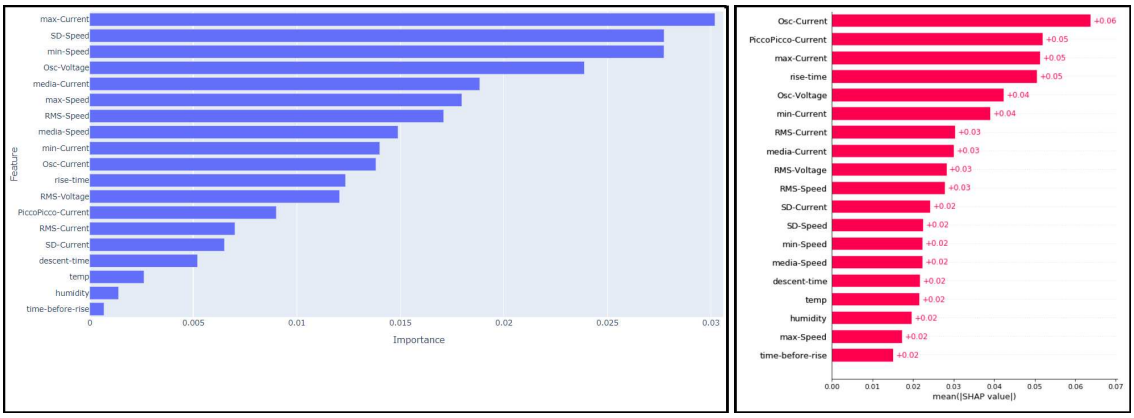


Figure 7.2: Global AcME (←) and SHAP (→) : Bar plot of the importance of the features.

These considerations also seem to be shared by the global interpretability result provided by SHAP. Figure 7.2 (→) illustrates the bar plot of the global SHAP designed to display a summary of the features that the model considers the most important in influencing the output itself. The global importance of each feature is calculated as the mean absolute value of the Shapley values for that feature. In particular, it can be seen that the feature *Time before rise* assumes the lowest score in terms of global importance. Even for SHAP this feature is not very relevant in discriminating the level of anomaly of the cycles.

In a more detailed picture, Figure 7.3 shows the summary graphs of the global SHAP and AcME, designed to display an information-dense summary of how the features impact the model's output. In this case, interpretability is useful as a tool for investigating the behavior of the model and the degree to which it is consistent with expectations. First of all, it is possible to notice a first characterization of the features, which, independently of the type of cycle, are dedicated to quantify the oscillations of the driver's signals. Consider, for example, the features *Osc*, *Peak-to-Peak*. Both methods of interpretability share the same results on the fact that very high or low values of the aforementioned features tend to characterize a cycle as anomalous. This is justified in retrospect by the fact that the acquisitions of 25 and 27 October, which tend to be considered the most anomalous, are the only ones performed with different load configurations. In particular, on October 25th, the load was changed so that a higher level of imbalance occurred in the machine. Consequently, as mentioned in Section 3.2, this induces higher oscillations in the driver's signals. However, on October 27th the tests were carried out without using the load. In this case, since the machine is fully balanced, not much variability is appreciated in terms of oscillations of the driver's signals.

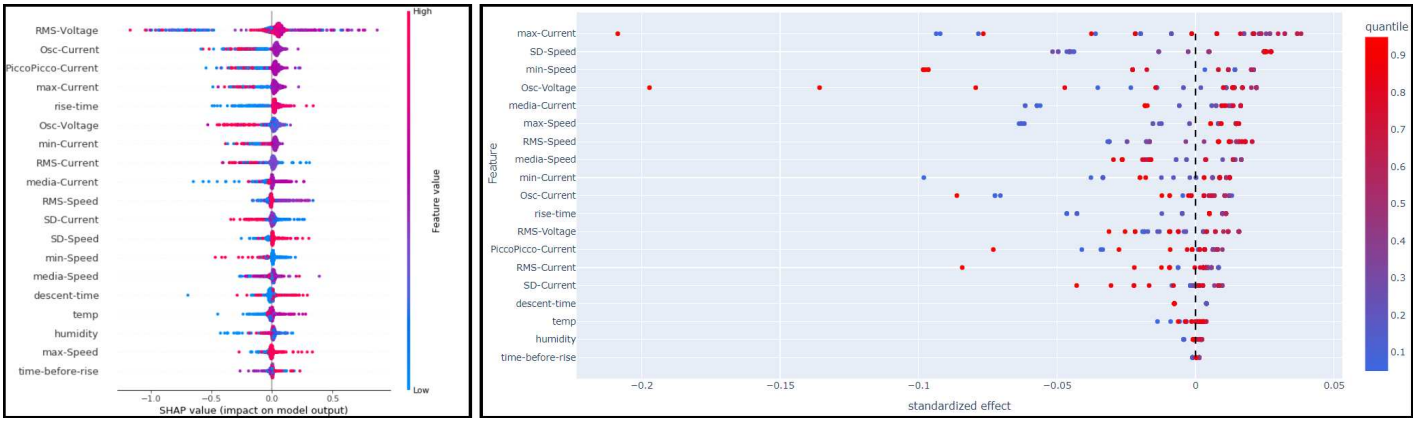


Figure 7.3: Global SHAP (←) and AcME (→): Summary plot.

Another interesting consideration can be seen from the importance assigned by both methods to the feature *Rise time*. In general terms, shorter rise times appear to be associated with abnormal behavior. This can be validated considering that on October 25th the overall load value was half that of the previous days. Also, as already mentioned, on October 27th there was no load at all. Through correlation analysis, a positive correlation (Figure 7.4) is observed between the overall load value and the rise time. Intuitively, if the weight is less, it is reasonable to think that slightly less time is needed to lift it.

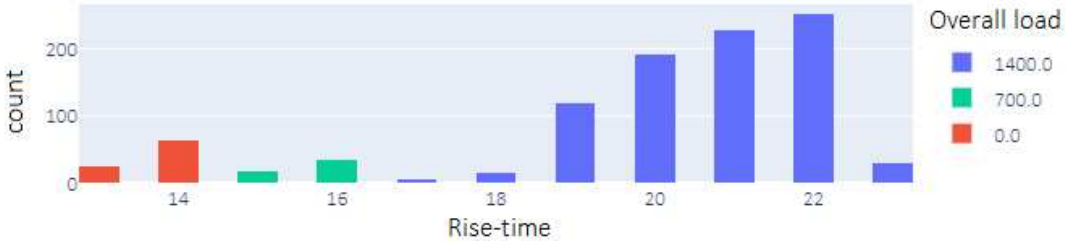


Figure 7.4: NebulaZ_C21111. Rise-time distribution; focus on the overall load value.

7.3 NebulaZ_C21111: Results

As anticipated, let us now consider the analysis at a fixed operating point so as not to be conditioned by the scarcity of data relating to the different operating conditions. In particular, the acquisitions of *NebulaZ*_C21111 which were carried out with the same load are considered. These acquisitions include the days from October 18 to October 22, 2021. In this context, anomaly detection by IF and the related methods of interpretability can be significant in detecting sub-optimal conditions and providing information on why a point is labeled as anomalous. Even in this case, a first significant evaluation can be made by analyzing the evolution of the AS provided by the IF: Figure 7.5 shows the trend of the latter over the course of the cycles. The cycles are sorted in chronological order.

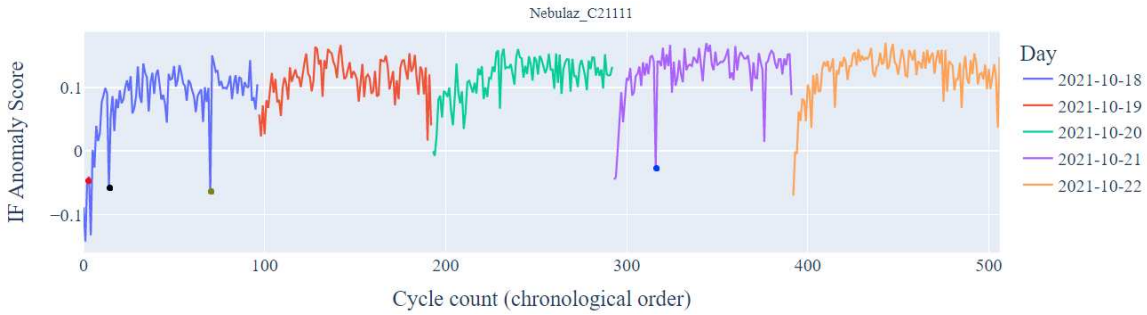


Figure 7.5: IF Anomaly Score, *NebulaZ*_C21111, October 18 . . . 22, 2021, Contamination = 0.03.

A further global assessment can be done in a similar way to what was presented above. In fact, it is possible to assess in general terms how features impact the model output. In this regard, Figure 7.6 illustrates the summary graphs of the global SHAP and AcME. Looking at the results of SHAP and, in particular, those provided by AcME, it seems that for most of the features the

values in the highest or lowest range (represented by the red and blue colors, respectively) are positively affecting the model output in having a higher AS. Indeed, values that differ from a possible steady-state behavior can be a symptom of anomaly. A further evaluation can be made considering features such as *RMS* Current. The latter tends to provide information in terms of a positive correlation with the current draw. As mentioned by domain experts, early cycles tend to suffer from a slight increase in consumption, especially on cold days. Given the fact that the first cycles of each acquisition are considered among the most anomalous by IF, we could assess that explainability methods tend to confirm what experts report. In fact, according to Figure 7.6 higher values of *RMS* Current push the prediction towards abnormal behavior.

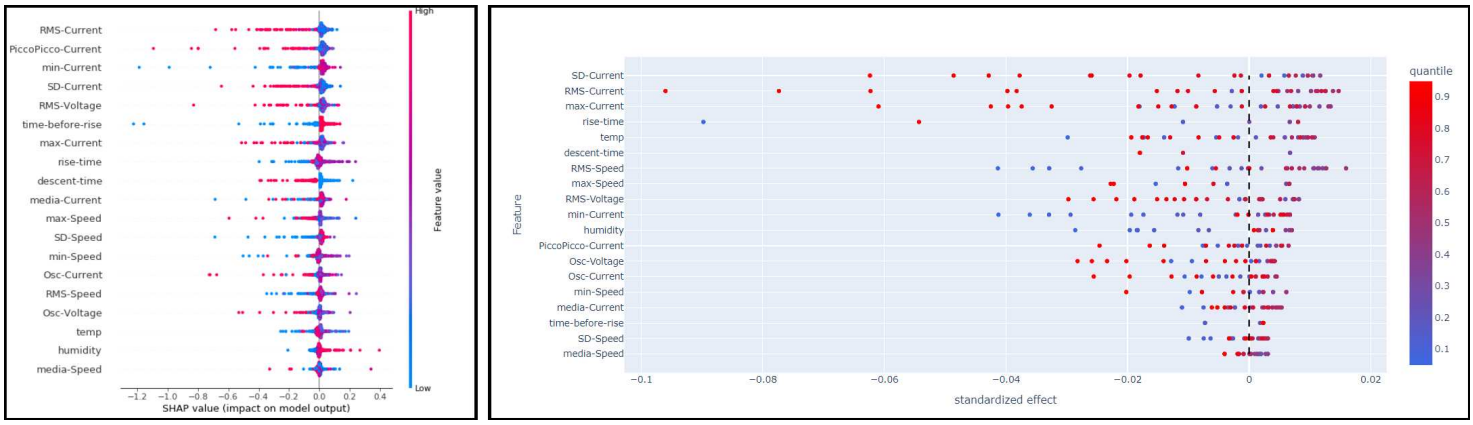


Figure 7.6: Global SHAP (←) and AcME (→): Summary plot.

Beyond the behavior on a global level, it is more interesting to evaluate the anomalous cycles from a local point of view. In this way it is possible to delineate case-by-case considerations of which features have most influenced the outcome in order to motivate the predictions provided by the IF.

7.3.1 Anomalies Detected

The analysis procedure presented in the following is adopted to analyze some of the anomalies detected by the multivariate approach in the case study *NebulaZ*_C21111. To avoid redundancies, reference is made to a subset of all the anomalies detected, i.e., those caused by different reasons. In particular, the cycles marked with a circle in Figure 7.5 are treated. These cycles are reported in Table 7.1.

Acquisition	Cycle	Start hour:minute:second	Anomaly Score	Color
2021-10-21	1521	07:24:52	−0.026	●
2021-10-18	1203	08:22:20	−0.04	●
2021-10-18	1214	09:08:02	−0.057	●
2021-10-18	1270	13:00:48	−0.066	●

Table 7.1: Anomalies subject of analysis.

The first cycle analyzed is the one corresponding to the first row of Table 7.1, namely the cycle ● 1521 of October 21 . To understand and justify the AS assigned to this cycle, it is significant to rely on interpretability tools. Figure 7.7 illustrates the SHAP local waterfall plot which creates a local feature importance plot where the bars are the Shapley values for each feature. The feature values are shown in gray to the left of the feature names. The waterfall plot powerfully shows why a case receives its prediction given its variable values. From Figure 7.7 it seems that a great influence in labeling the cycle as anomalous is due to the features *Peak-to-Peak* and *Min* Current.

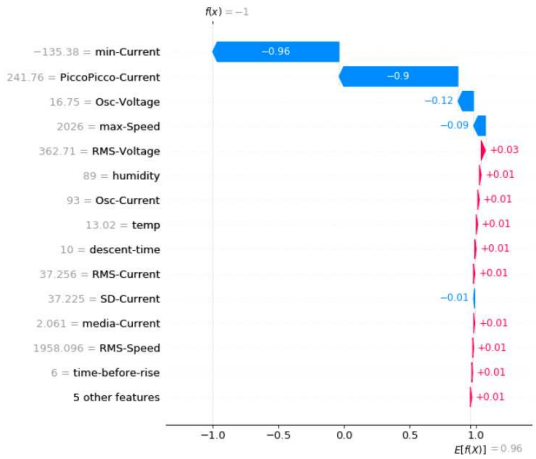


Figure 7.7: Local SHAP: Cycle ● 1521, October 21th.

A complementary analysis can be performed by evaluating the local interpretability result provided by AcME. In this case, Figure 7.8 illustrates the results of the local AcME on the cycle under examination. The larger dots represent the quantile value of each feature in this specific observation. This view can provide a *what-if* analysis, that is, information about what will happen with the observation if we change the quantile value of a specific feature while keeping fixed all the other features. It can be seen that the current observation presents a high *Peak-to-Peak*

Current value (highest quantile) and a low *Min* Current value (lowest quantile). Furthermore, it can be observed that higher *Min* Current values or lower *Peak-to-Peak* Current values (than those currently assumed by the observation) would lead the observation to be labeled normal.

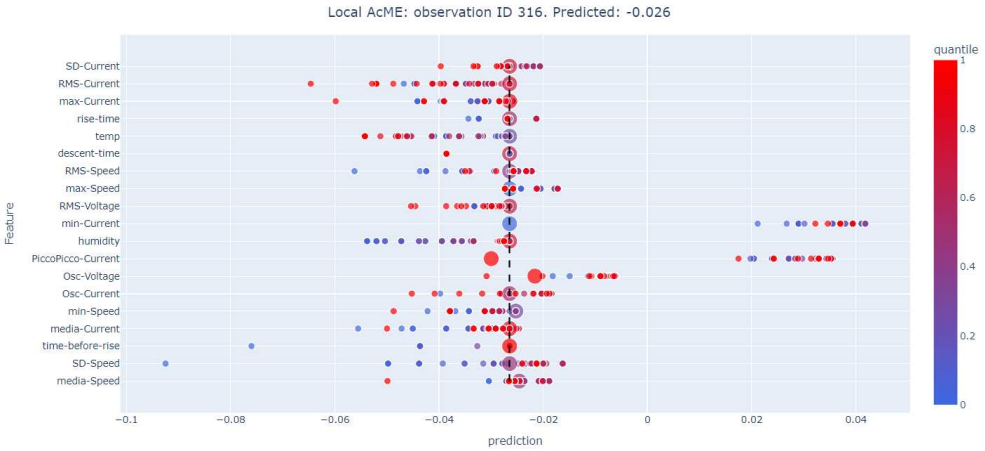


Figure 7.8: Local AcME: Cycle ● 1521, October 21th.

To verify the anomalous behavior of this cycle *a posteriori*, it is significant to look at the driver's signals of the previous/subsequent cycles of the same type that are part of the session of October 21. Figure 7.9 illustrates the comparison just mentioned. A blue mark is intended for cycle 1521. The current and voltage signals do not appear to have an observable difference, while a negative current pulse is observed in proximity to the moment in which the machine reverses the direction of rotation.

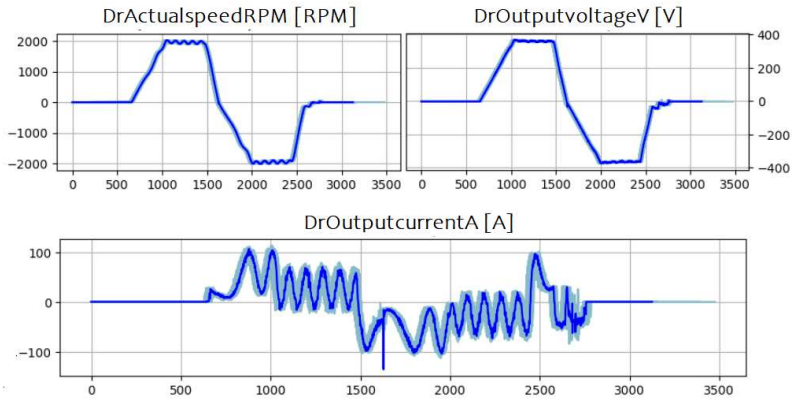


Figure 7.9: Driver's signals: A comparison between cycle 1521 (——) and the previous/subsequent cycles of the acquisition of October 21.

The second case analyzed concerns the cycle ● 1203 of October 18th. In analogy to the previous case, it is useful to see what is proposed by the local SHAP method. Figure 7.10 illustrates the assignment of the local importance of the features. In this case, it appears that the feature *RMS* Current plays a major role in categorizing this cycle as anomalous. As mentioned above, this feature is positively influenced by a general increase in current consumption. In any case, it can be observed that different features synergistically tend to contribute to the attribution of an anomalous score. The cycle under review highlights the multivariate perspective of anomaly detection. From a temporal point of view, this cycle ranks as one of the first to be performed during a session. The considerations made with respect to a general increase in consumption in the first cycles seem to be verified by the local importance that has been assigned.

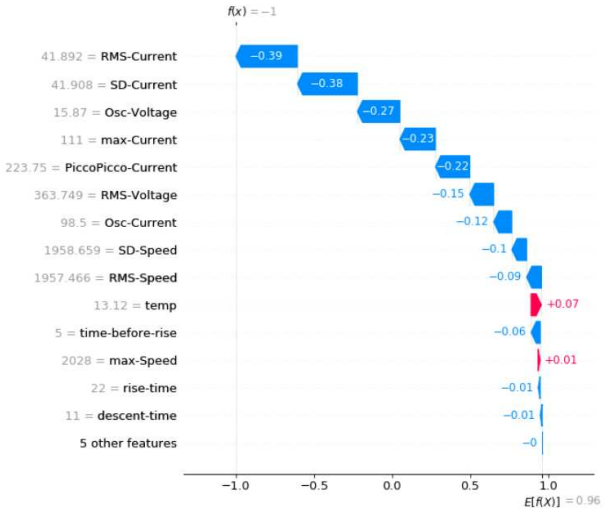


Figure 7.10: Local SHAP: Cycle ● 1203, October 18th.

A concomitant analysis can also be provided by the AcME local interpretability tool. Figure 7.11 provides the AcME local importance of features. It can be seen that the feature *RMS* Current of the actual observation assumes a value relative to the highest quantile. In this case, regardless of whether we increase or decrease the quantile value of any feature, the prediction score of the cycle will still be negative.

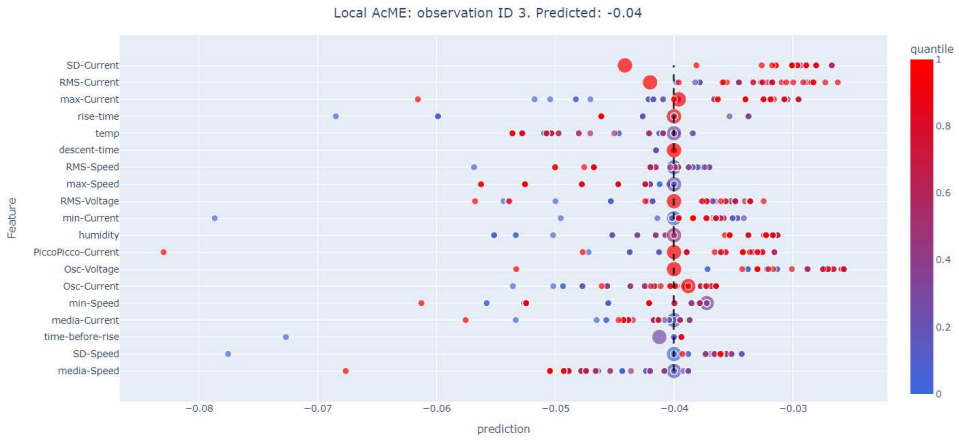


Figure 7.11: Local AcME : Cycle ● 1203, October 18th.

Another case to examine is represented by the cycle ● 1214 of October 18th. Similarly to the previous cases, Figure 7.12 illustrates the local importance of the features assigned by SHAP. The feature *Peak-to-Peak* Current is of fundamental importance in classifying this cycle as anomalous. From the local interpretability of AcME in Figure 7.13, it can be seen how the value of the feature *Peak-to-Peak* Current is relative to the highest quantile. Furthermore, through the *what-if* analysis it can be deduced that, by changing the *Peak-to-Peak* Current value towards lower quantiles, the cycle would be classified as normal.

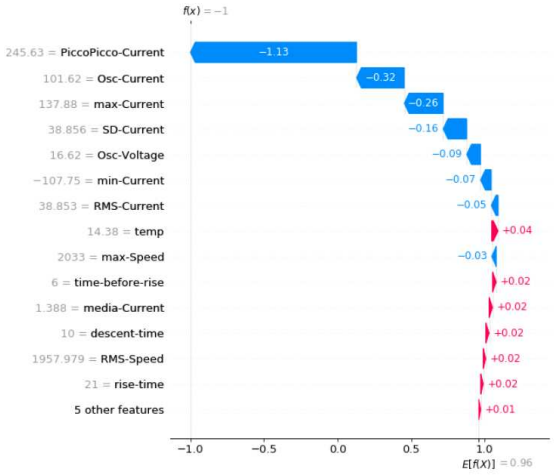


Figure 7.12: Local SHAP: Cycle ● 1214, October 18th.

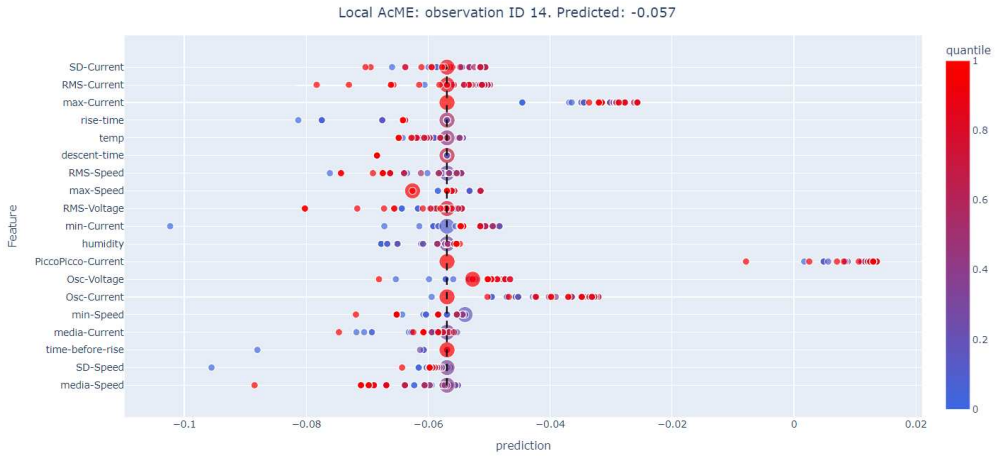


Figure 7.13: Local AcME: Cycle ● 1214, October 18th.

Similarly to the first case, an evaluation can be made by comparing the signals of the driver. Figure 7.14 illustrates the driver's signals of the previous/subsequent cycles of the same type that are part of the session of October 18. A black mark is intended for cycle 1214. It is possible to observe a positive current pulse in proximity of the end of the cycle.

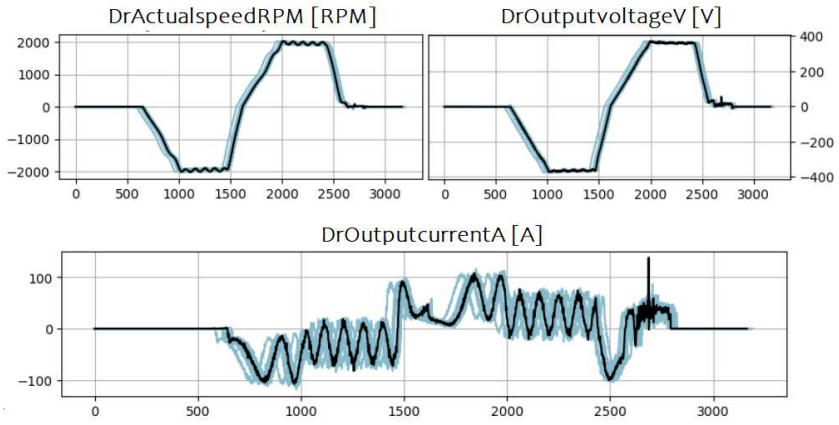


Figure 7.14: Driver's signals: A comparison between cycle 1214 (——) and the previous/subsequent cycles of the acquisition of October 18.

The last case subject to analysis concerns cycle ● 1270 of October 18th. In this case, the cycle was detected as anomalous because it does not represent a normal behavior with respect to the hypothesis of operation, that is, with respect to the types of cycles that the machine can perform. More in detail, the causes can be guessed by exploiting the results of the SHAP and AcME local interpretability methods, respectively, in Figure 7.15 and Figure 7.16.

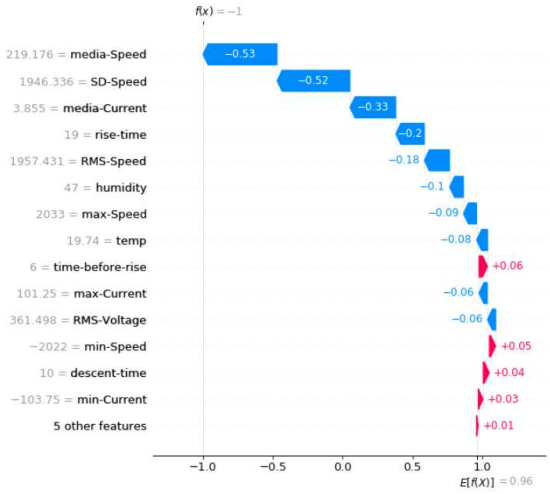


Figure 7.15: Local SHAP: Cycle ● 1270, October 18th.

The role of the most significant features in classifying this cycle as anomalous can be traced back to the features *Mean* Speed, *Mean* Current, *SD* Speed. With the introduction in Section 2.1 of the type of cycle 1, the latter has been defined as a cycle in which the motor changes the direction of rotation at half the duration. As for the current cycle under examination, the features identified as most important are representative of the fact that one direction of rotation has been employed more than another. To verify this, Figure 7.17 illustrates the driver's signals of the previous/subsequent cycles of the same type that are part of the session of October 18th. An olive color mark is intended for cycle 1270. We recall that, contrary to this, the features are independent of the cycle duration as the latter is a tunable parameter.

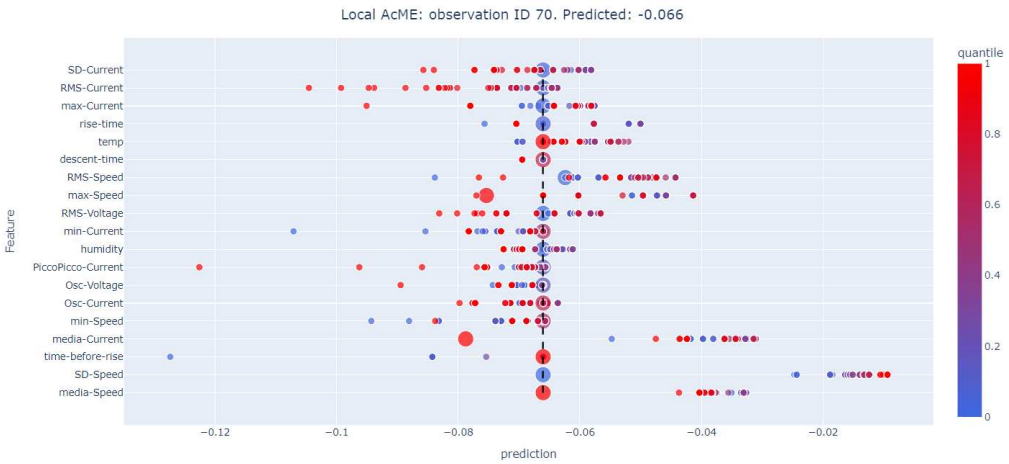


Figure 7.16: Local AcME: Cycle ● 1270, October 18th.

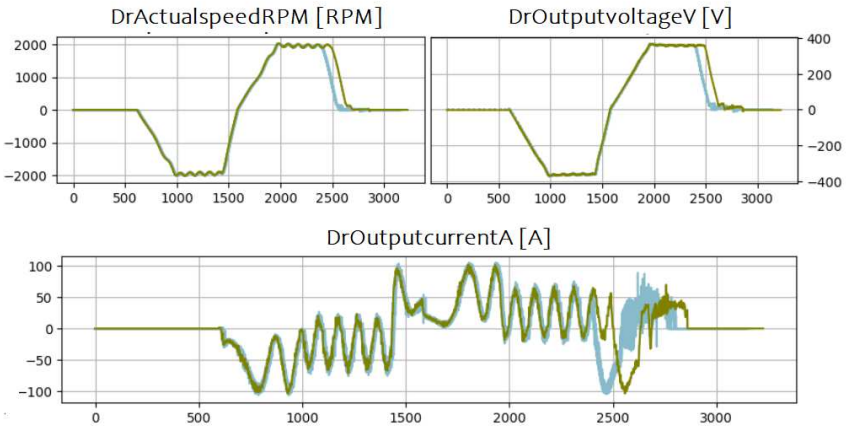


Figure 7.17: Driver's signals: A comparison between cycle 1270 (——) and the previous/subsequent cycles of the acquisition of October 18.

A conclusive analysis can be made by viewing the values of the features in the anomalous cycles. To this end, Figure 7.18 shows the distribution histograms of some of the features considered among the most important by the previous local interpretability results. Each of these highlights the value of the feature for each of the four anomalous cycles examined.

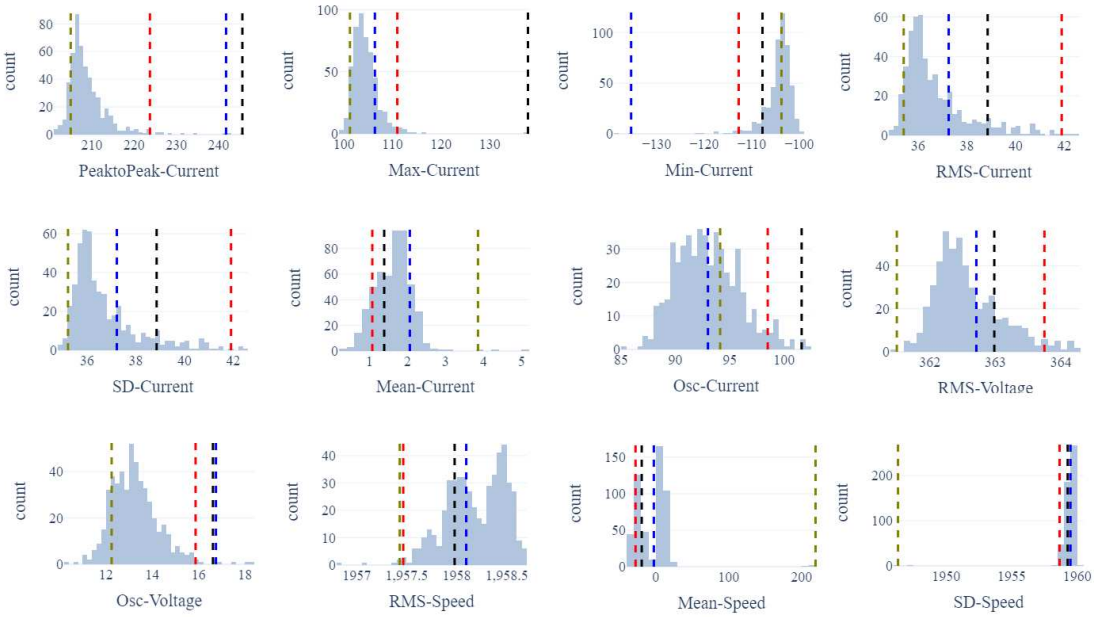


Figure 7.18: Features distribution: cycle 1521 (——) October 21th, cycle 1203 (——) October 18th, cycle 1214 (——) October 18th, cycle 1270 (——) October 18th.

8

Conclusion and Future Work

In this thesis a problem of AD related to a real case study of *Antonio Zamperla S.p.A.* has been addressed. The proposed approach includes a multivariate feature-based AD method in synergy with the interpretability tools. Initially, as data were acquired during the testing phases, the data cleanup and preparation procedure played a significant role. In particular, by interpolating the domain knowledge inputs with the data analyzes, it was possible to evaluate with critical thinking any inconsistencies in the data tagging. By virtue of the AD approach, a preliminary analysis was performed in order to identify and assess the causes of possible variability of the signals. The feature extraction procedure was focused on the analysis of the most variable traits of the signals most relevant to machine monitoring. In this case, the concomitant discussion with the domain experts allowed the validation of some considerations and the filling of the gaps due to the unsupervised settings. The application of the IF algorithm has highlighted an aspect related to the lack of data due to the fact that we are in a preliminary phase of data acquisition and it is therefore reasonable to expect that the variety and quantity of the tests carried out is rather limited. In spite of this, it was possible to obtain significative results which confirmed the observations of the domain experts and highlighted behaviors not in line with the hypotheses of normal functioning. In this work, we proposed two approaches to model explainability, AcME and SHAP, aimed at analyzing the role played by each feature (at both global and local scale) in order to support RCA. The rationale behind AcME allows for a substantial reduction in computational time, while retaining a quality of explanations comparable to state-of-the-art interpretability method SHAP. Being able to quickly identify the roots cause

of an anomalous behavior is particularly important in the scenario under examination. This capability can lead to important savings both in terms of time and costs. Although SHAP is supported by a refined theoretical basis, the specific context of use does not always allow its use due to the relative calculation times. The adoption of both interpreatibility methods has provided consistent results which have proved to be considerably informative when, in retrospect, we wanted to investigate the reason why the problem arose.

In conclusion, a more informative data collection could achieve an improvement of the current approach. First of all, in general terms, it is necessary to have a greater amount of data regarding the different operating conditions carried out. To this end, the set of measured variables could also be extended, including, for example, the lubrication state of the machine. This could provide information to discriminate in a more distinctive way the main aspects that influence the state of the machine. Second, a substantial improvement may arise with respect to the current unsupervised nature of data collection. Stimulating the data tagging of some of the anomalous and normal cases could make it possible to evaluate the performance of the algorithm in terms of an accuracy metric. Furthermore, the labeling of the data could lead to the use of more advanced supervised or semi supervised approaches.

References

[1] M. Berno, M. Canil, N. Chiarello, L. Piazzon, F. Berti, F. Ferrari, A. Zaupa, N. Ferro, M. Rossi, and G. A. Susto, "A machine learning-based approach for advanced monitoring of automated equipment for the entertainment industry," in 2021 IEEE International Workshop on Metrology for Industry 4.0 & IoT (MetroInd4. 0&IoT). IEEE, 2021, pp. 386–391.

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

[3] D. Gunning, "Explainable artificial intelligence," Defense Advanced Research Projects Agency (DARPA), p. 2, 2017.

[4] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," Advances in neural information processing systems, vol. 30, 2017.

[5] D. Dandolo, C. Masiero, M. Carletti, D. D. Pezze, and G. A. Susto, "Acme–accelerated model-agnostic explanations: Fast whitening of the machine-learning black box," arXiv preprint arXiv:2112.12635, 2021.

[6] OpenWeather. [Online]. Available: https://openweathermap.org/

[7] J. G. Dy and C. E. Brodley, "Feature selection for unsupervised learning," Journal of machine learning research, vol. 5, no. Aug, pp. 845–889, 2004.

[8] D. Freedman, R. Pisani, and R. Purves, "Statistics (international student edition)," Pisani, R. Purves, 4th edn. WW Norton & Company, New York, 2007.

[9] K. Pearson, "Liii. on lines and planes of closest fit to systems of points in space," The London, Edinburgh, and Dublin philosophical magazine and journal of science, vol. 2, no. 11, pp. 559–572, 1901.

[10] A. Ayadi, O. Ghorbel, A. M. Obeid, and M. Abid, "Outlier detection approaches for wireless sensor networks: A survey," Computer Networks, vol. 129, pp. 319–333, 2017.

[11] D. M. Hawkins, Identification of outliers. Springer, 1980, vol. 11.

[12] A. Tsymbal, "The problem of concept drift: definitions and related work," Computer Science Department, Trinity College Dublin, vol. 106, no. 2, p. 58, 2004.

[13] E. L. Paula, M. Ladeira, R. N. Carvalho, and T. Marzagao, "Deep learning anomaly detection as support fraud investigation in brazilian exports and anti-money laundering," in 2016 15th ieee international conference on machine learning and applications (icmla). IEEE, 2016, pp. 954–960.

[14] U. Porwal and S. Mukund, "Credit card fraud detection in e-commerce: An outlier detection approach," arXiv preprint arXiv:1811.02196, 2018.

[15] R. Ghevariya, R. Desai, M. H. Bohara, and D. Garg, "Credit card fraud detection using local outlier factor & isolation forest algorithms: A complete analysis," in 2021 5th International Conference on Electronics, Communication and Aerospace Technology (ICECA). IEEE, 2021, pp. 1679–1685.

[16] H. John and S. Naaz, "Credit card fraud detection using local outlier factor and isolation forest," Int. J. Comput. Sci. Eng, vol. 7, no. 4, pp. 1060–1064, 2019.

[17] K. Alrawashdeh and C. Purdy, "Toward an online anomaly intrusion detection system based on deep learning," in 2016 15th IEEE international conference on machine learning and applications (ICMLA). IEEE, 2016, pp. 195–200.

[18] G. B. Gebremeskel, C. Yi, Z. He, and D. Haile, "Combined data mining techniques based patient data outlier detection for healthcare safety," International Journal of Intelligent Computing and Cybernetics, 2016.

[19] S. Cateni, V. Colla, M. Vannucci, J. Aramburo, and A. R. Trevino, "Outlier detection methods for industrial applications," Advances in Robotics, Automation and Control, pp. 265–282, 2008.

[20] H. Wang, M. J. Bah, and M. Hammad, "Progress in outlier detection techniques: A survey," Ieee Access, vol. 7, pp. 107 964–108 000, 2019.

[21] R. Chalapathy and S. Chawla, "Deep learning for anomaly detection: A survey," arXiv preprint arXiv:1901.03407, 2019.

[22] T. Miller, "Explanation in artificial intelligence: Insights from the social sciences," Artificial intelligence, vol. 267, pp. 1–38, 2019.

[23] C. Molnar, "A guide for making black box models explainable," URL: https://christophm. github. io/interpretable-ml-book, 2018.

[24] S. M. Lundberg, G. G. Erion, and S.-I. Lee, "Consistent individualized feature attribution for tree ensembles," arXiv preprint arXiv:1802.03888, 2018.