Aalto-yliopisto
Aalto-universitetet
Aalto University

Master's Programme in Computer, Communication, and Information Sciences

Multi-label classification of a hydraulic system using Machine Learning methods.

Cristina Molinero Ranera

Master's Thesis
Spring 2022

Copyright ©2020 Eddie Engineer

Author Cristina Molinero Ranera

Title of thesis Multi-label classification of a hydraulic system using Machine Learning methods

Programme Master's Programme in Computer, Communication and Information Sciences

Major Exchange studies in Computer Science

Thesis supervisor Alexander Jung

Thesis advisor(s) Alexander Jung

Date 05.07.2022 **Number of pages** 51

Abstract

In this project, a condition monitoring of a hydraulic system has been developed. The research consisted of a health categorization of the most relevant physical and non-physical elements of the system. The objective has been to use different ML models to classify the state of the elements in each cycle and to be able to know through the information of the features of each cycle whether an element of the system needs to be replaced or not and also find out the work efficiency of each element of study.

This research therefore follows a supervised learning analysis in which two types of classifications will be carried out. The first one will be a multiclass classification done with different ML techniques that will try to classify the categories of each label separately, getting to know for each cycle which is the state of the analyzed element.

On the other hand, a multilabel analysis will follow. In this case, all labels will be taken, and different performances will be done. The main objective in this chapter will be to elaborate different tests with different ML models in order to see which is the optimal one for this system, and which is the algorithm that should be used to monitor this type of hydraulic system.

In addition to these classification analyses, the correlation between the different data will be assessed beforehand in order to verify relationships or coincidences that may be relevant.

Keywords supervised learning, hydraulic system, multilabel classification, monitoring

Contents

Preface

I want to thank first of all my supervisor, Professor Alexander Jung, and Yu Tian and Anni Rytkönen for their guidance.

I also want to thank my computer science friends for their help and support.

Otaniemi, 5 July 2022
Cristina Molinero Ranera

Abbreviations

AI	Artificial Intelligence
LSTM	Long short-term memory
PCA	Principal Component Analysis
ML	Machine Learning
RNN	Recurrent Neural Network
RMSprop	Root Mean Square Propagation
API	Application programming interface
KNN	K- nearest neighbors

1 Introduction

1.1 Machine Learning

“Machine learning is an evolving branch of computational algorithms that are designed to emulate human intelligence by learning from the surrounding environment. They are considered the working horse in the new era of the so-called big data” [1].

Machine learning is considering a strong tool that as it is define before, analyze the surrounding area of a problem as if it were human intelligence, and this means that it is no longer standard computers that work, but intelligent computers that act as if the analysis were done by a human with a hand.

Professor Alexander Jung refers in his book "Machine Learning, The Basics" to the principle of "trial and error". He explains that "This principle consists of the continuous adaptation of a hypothesis about a phenomenon that generates data" [2]. We could consider this concept as continuous learning, since there is no fixed path from the beginning, but thanks to the previous steps of analysis, the process is optimized towards a better final solution than the one that was predefined at the beginning.

In conclusion, we have chosen to work with machine learning within the whole area of innovative technologies because of everything described above and because in the end we thought that the range that this tool can offer was sufficiently large and as Alexander Jung comments "ML methods have also become standard tools in many fields of science and engineering” [2], so it is a topic very common and popular in the scientific area.

Finally, as Arthur Samuel, one of the founders of the concept of machine learning, says, it's a "Field of study that gives computers the ability to learn without being explicitly programmed". It is considered that using machine learning is the best option to work with as there isn’t a mandatory requirement of programming skills.

1.2 Motivation

In today's world, new technologies are becoming more and more important. Technological innovation is used to address the sustainability of systems and make them more ecofriendly and that is the point this project is intended to follow.

Furthermore, our civilization is suffering the consequences of climate change, which is growing faster and faster. Many industrial processes, although they have improved in energy efficiency, continue to be highly polluting systems due to the immense atmospheric damage they generate, both in terms of gases and industrial materials.

Artificial intelligence is also used to create and develop projects that can have an impact on our society. The range of applicability of AI projects is so large that the industry is growing by leaps and bounds. The motivation comes from this eagerness to be part of this technological innovation that scientific world is facing. It is very interesting to have data extract from a system and to improve the system by analyzing it. This way of optimization seems extraordinary and that is why in this project the same dynamics are elaborated.

In conclusion, being able to use innovative information technologies, such as machine learning to optimize, change, rebuild systems and make them more sustainable, is a good way to help the industry sector to eliminate obsolete processes and elaborate strategies focused on reducing pollution, creating fully sustainable eco-systems and eradicating unnecessary waste of industrial material.

1.3 Problem Statement and Research questions

This investigation focuses on improving the hydraulic system under study by making it more eco-friendly and more sustainable thanks to monitoring the elements of the system. This monitoring is necessary as it controls the substitution of materials, preventing the unnecessary waste of the physical elements of these systems and creating a correct and healthy maintenance strategies for the industry and that above all help the entire environment.

The aim of this project is to optimise the maintenance of this type of hydraulic system by analysing the behaviour of its elements, which are characterize as labels. It is possible to characterize them thanks to the information provided by a wide variety of sensors, in this case features data.

Furthermore, these sensors distributed throughout the system capture and record characteristics variables of the system, this data is captured for each

operating cycle. In addition, our labels are composed by the health state of 5 important physical and non-physical elements of the system. Two of label type are for example the pump and the cooler, both physicals element of the system. In short, the goal is to characterize the health state of a system cycle, to label each cycle with the help of the sensors, features that are distributed over the hydraulic system.

To achieve this, the supervised learning technique will be used, also known as supervised machine learning. This technique is a subcategory of machine learning. Data of the system is labelled and will be use as features to train algorithms to classify the outputs. Our inputs are fed to the model, which adjusts the weights until the model develops automated learning (machine learning). Different ML methods have been applied to understand the behavior of the system and classify the different labels according to their health state.

According to the research questions the thesis tries to answer 3 different statements.

First of all, it would be interesting to know if does it help more to use multiclass classification as a preliminary analysis or would have been more efficient to start the investigation performing multi label classification directly?

On another hand, the project will try to discover a similarity or a common pattern between the ML methods that have the highest accuracy. Do they follow a particular path, do they have theorical similarities?

Finally, focusing on optimization of the data pre-processing. In this type of problems work with a 2-feature dimensional dataset would be better than elaborate a highest dimensional feature dataset.

1.4 Structure of the Thesis

The structure of the thesis is as below:

1. The first chapter introduces the background of our project, it raises the research questions. In addition, we also talk about motivation and why we have chosen ML as our main theme.

2. The second part of the project explains the process of creating our dataset and discusses the different tools used. Then, the final structure of the ready-to-use databases is shown, and a correlation study is carried out. Last but not least, the different machine learning methods are explained.

3. In the third chapter the results of the analysis are described in two sub chapters. In the first part, the results of the analysis of each of the labels separately is discuss, this being a multiclass classification analysis. This is followed by an analysis of the results given by the multioutput classification technique and multilabel classification results are reported.

4. The fourth chapter summarizes the analysis by drawing relevant conclusions and discusses both the limits of the project and possible recommendations for future research.

2 Literature Review

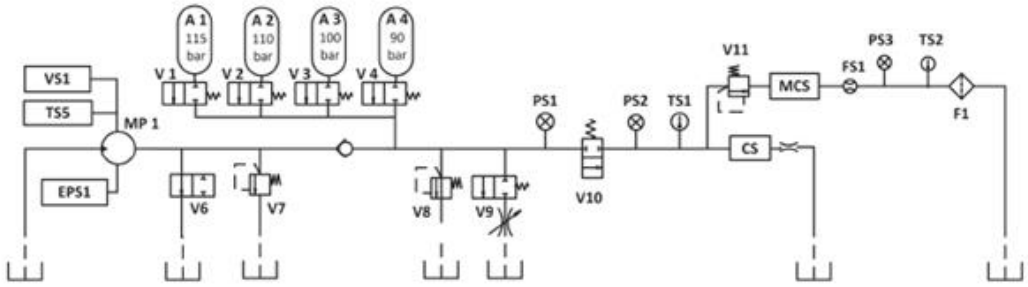
2.1 Background

The dataset is provided by The UCI Machine Learning Repository which offers “a large collection of databases that are used by the machine learning community for empirical analysis of machine learning algorithms” [3].
The dataset includes sensors and labels information, the size is around 530 MB and all of it is given in a “.txt” format.

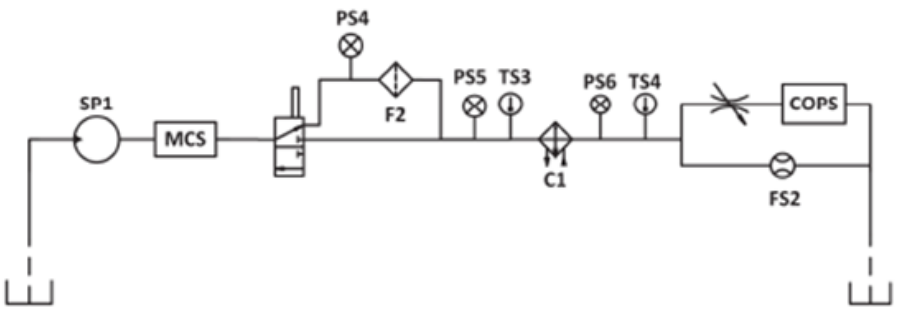
2.1.1 Hydraulic System

“In machine learning and patter recognition, a feature is an individual measurable property or characteristic of a phenomenon” [4]. This means that the features are an important element for the analysis as they characterize the system. This characterization will be carried out thanks to the ML models that will internally build an algorithmic model which will be able to associate specific information of the features with the health classification labels.

Below is a schema of the hydraulic system. All the hydraulic physical elements can be seen, those that give information of the features and the hydraulic elements associated with the labels. It would be seen that this hydraulic system has two sub circuits, working circuit and cooler refrigerator circuit.



Working circuit [3]



Cooler circuit [3]

Firstly, there is a vibration sensor (VS1) at the start of the system as well as EPS1, the motor power sensor. As the name of vibration sensor indicates, it analyses the vibration of the pump. On the other hand, the system has 4 temperature sensors (TS1,TS2,TS3,TS4) distributed throughout the system and another 6 pressure sensors (PS1,PS2,PS3,PS4,PS5,PS6) also distributed throughout the system. Pressure sensors are on either side of elements such as the valve and cooler. Having different sensors with the same characteristics distributed throughout the system helps to better classify the system status for each cycle. Additionally, 2 sensors are used to measure the volum flow of the system (FS1, FS2). It is possible to analyze how the fluctuation of the different volum-flow sensors affects the classification of the labels.

On the other hand, the most interesting sensors are those that evaluate the system as efficiency factor (SE), the cooling power (CP) and the cooling efficiency (CE). In total this project is going to use 17 features.

CE: cooler efficiency (virtual %)
CP: cooler power (virtual %)
EPS1: motor power (W)
FS1: volum flow 1 (l/min)
FS2: volum flow (l/min)
PS1: pressure sensor 1 (bar)
PS2: pressure sensor 2 (bar)
PS3: pressure sensor 3 (bar)
PS4: pressure sensor 4 (bar)
PS5: pressure sensor 5 (bar)
PS6: pressure sensor 6 (bar)
SE: system efficiency factor (%)
TS1: temperature sensor 1 (°C)
TS2: temperature sensor 2 (°C)
TS3: temperature sensor 3 (°C)
TS4: temperature sensor 4 (°C)
VS1: vibration sensor (mm/s)

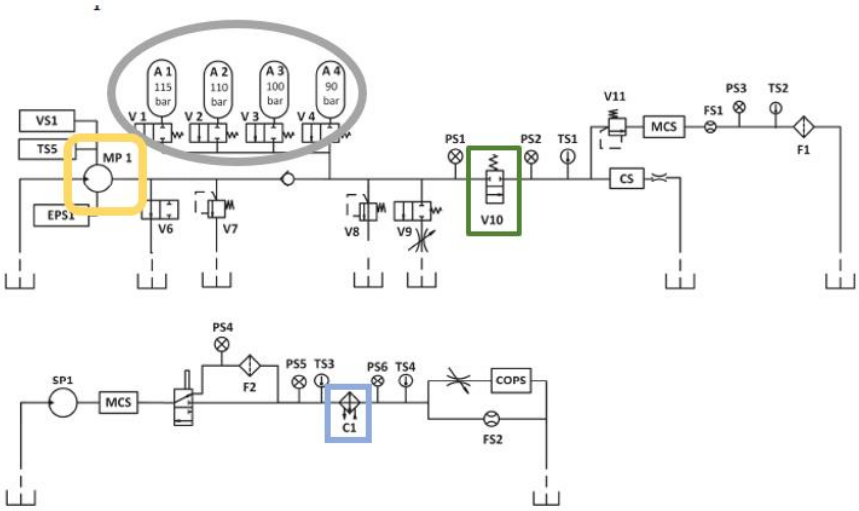
Label description [3]

CE	31.299077
CP	1.808399
EPS1	2495.509203
FS1	6.198549
FS2	9.649453
PS1	160.485315
PS2	109.379906
PS3	1.753227
PS4	2.600266
PS5	9.163320
PS6	9.079363
SE	55.287900
TS1	45.424567
TS2	50.365979
TS3	47.662121
TS4	40.735654
VS1	0.613315

Numerical mean values of features data

Features have different numerical ranges because each feature has a different measurement parameter. The motor power parameter (EPS1) has numerical values around 2500 and on the other hand parameters such as CP, PS3, VS1 are features that range between null values and 5. The rest of the values range between minimum values of 10 and maximum values of 100 and in some cases 200.

This will have to be considered and arranged before proceeding with machine learning analysis. That's why a standardization technique will be conducted.



Hydraulic system [3]

Labels are the physical or non-physical system parameters. In this study 5 types will be used. First there is a label called cooler, this label is associated with the functionality of the C1 device to cool the fluid, in the picture below, this element is colored in blue. In this type of systems, having a cooler with the highest performance is essential because otherwise the system does not

do what it should. Additionally, there is a physical element that is very characteristic of hydraulic systems, the valve, which regulates the fluid, colored in green. In this case, as it is seen in the picture above V10 valve would be label number 2 and this project will try to optimize its maintenance. Furthermore, there is the hydraulic pump (MP1 in the diagram). This is the element that allows the fluid to be correctly introduced into the system and that is why looking for a good functionality helps the system to be more efficient. Moreover, the system includes 4 accumulators, which are responsible for storing energy and manage it at a time of demand. Finally, one of the labels to study will be the stability of the system. Stability is very important to have under control as it is a guideline of the system behavior as it can alert of a future bursting of the system's pipes.

The categories of each type of study label are specified below.

1: Cooler condition / % :
3: close to total failure
20: reduced efficiency
100: full efficiency

2: Valve condition / %:
100: optimal switching behavior
90: small lag
80: severe lag
73: close to total failure

3: Internal pump leakage:
0: no leakage
1: weak leakage
2: severe leakage

4: Hydraulic accumulator / bar:
130: optimal pressure
115: slightly reduced pressure
100: severely reduced pressure
90: close to total failure

5: stable flag:
0: conditions were stable
1: static conditions might not have been reached yet

It is important to note, after having describe all the measurement parameters of the system, that this investigation works with 2205 cycles and each one of them is classified with a combination of the 5 labels detailed above. In short,

for each cycle 17 features are used and help to describe the category of each label for each cycle.

2.2 Datasets

2.2.1 Data pre-processing

2.2.1.1. Libraries

It is considered interesting to include in the report a small explanation of the different tools available in the Python environment that have been used during the process. In this case there are libraries which were being used for the first time and throughout the construction of the work have become an essential part of the process. Mostly, NumPy and pandas have been used in the development of preprocessing and later, for the visualization of results libraries such as matplotlib and seasborn have been used. Finally, scikit-learn has also been used (sklearn), a library for machine learning problems, which have help to test the different machine learning models.

The focus with NumPy has been the transformation of data into np arrays. These arrays can support a wide variability of calculations and computations which interested in the project and even more because it is needed to have the data in NumPy array format for compatibility with machine learning techniques.

On the other hand, with the most popular library in data structure analysis, pandas, we have been able to structure the data using dataframes, which has allowed better visualization of all the data as it allows to store the data in the form of a table with headers. In addition, pandas have allowed us to import csv files to our jupyter hub platform in an easy and orientate way.
Seasborn, “a Python data visualization library based on matplotlib” [5] is the tool with which we have been able to display our correlation analysis as well as the results of the multiclass analysis.
In reference to seasborn, we have used the most common visualization library in data analysis, matplotlib. This is a comprehensive library for creating static, animated, and interactive visualizations in Python. " [6] . It has been used basically to visualize graphs of the results of deep learning studies, specifically the results of the performance of our neural networks.

Scikit-learn, as mentioned at the beginning, has been the application library for machine learning methods. Thanks to its functionalities it has been possible to perform the different machine learning techniques that will be announced later. Scikit-learn “is a Python module integrating a wide range of

state-of-the-art machine learning algorithms for medium-scale supervised and unsupervised problems. This package focuses on bringing machine learning to non-specialists using a general-purpose high-level language." [7] It is therefore the most appropriate Python application to work with in this project as the investigation is going to solve a supervised problem and the sklearn platform will be very useful and beneficial.

2.2.1.2. Sampling rate

A sample rate is a parameter used in many projects where data is the main source of information. This indicator is measured in Hertz (Hz) and allows to convert analogue samples to digital format.

In our case 1 Hz is equivalent to 60 values per cycle, so 10 Hz is equivalent to 600 values and therefore 100 Hz is equivalent to 6000 catchments per cycle.

Feature	Sampling rate (Hz)
CE : cooler efficiency (virtual %)	1 Hz
CP : cooler power (virtual %)	1 Hz
EPS1 : motor power (W)	100 Hz
FS1 : volum flow 1 (l/min)	10 Hz
FS2 : volum Flow (l/min)	10 Hz
PS1 : pressure sensor 1 (bar)	100 Hz
PS2 : pressure sensor 2 (bar)	100 Hz
PS3 : pressure sensor 3 (bar)	100 Hz
PS4 : pressure sensor 4 (bar)	100 Hz
PS5 : pressure sensor 5 (bar)	100 Hz
PS6 : pressure sensor 6 (bar)	100 Hz
SE : system efficiency factor (%)	1 Hz
TS1 : temperature sensor 1 (°C)	1 Hz
TS2 : temperature sensor 2 (°C)	1 Hz
TS3 : temperature sensor 3 (°C)	1 Hz
TS4 : temperature sensor 4 (°C)	1 Hz
VS1 : vibration sensor (mm/s)	1 Hz

Sampling rate description [3]

As can be seen in the table, the pressure sensors are the ones that record the most values per cycle. This is because pressure in a hydraulic system is very important, as it must be very well monitored to avoid possible breakage of physical elements and because of the risk that this would entail.
On the contrary, temperature sensors and sensors of a more global nature of the system have a low sampling rate, as it is understood that with this width of measurements the system under study is already correctly monitored.

The process of harmonizing the sampling rate has been done in such a way that all features have 60 values per cycle. For this purpose, all the features with a sampling rate higher than 1 Hz have been compacted. The idea was to perform an arithmetic average of every x value depending on the sampling rate of each feature. The goal is to obtain, as mentioned before, 60 values for each feature for each cycle. To be clear, for example, for pressure sensor 1 we come with 6000 captures for each cycle, we have computed a mean every 100 values to obtain 1. Moreover, for the features with 600 initial captures, in each cycle we have make an average between 10 values, several time to generate 60 values per each cycle at the end.
At for the technical level, the grouby function has been used to select the number of values needed for each arithmetic mean and has help to compute the average. In addition, each final value has been added to its corresponding new dataset in order to have it all together.

It is important to note that this process has been a real challenge in this project, as complex programming code have had to be developed.

2.2.1.3. Standardization

At some points, the data had to be standardized because the features as working with different units of measurement data such as (amps, watts, bar, degrees, etc.) are not suitable. These units of measurement are very common units in the hydraulic world, and we thought it was appropriate to standardize the data in order to weight our data correctly. It has also prevented parameters such as EPS1, motor power (watts), from being weighted more heavily than they should have been because they have higher values than the rest.

Furthermore, we wanted to take advantage of the standardization process use on this project to try to understand its general benefits and in addition analyze the technique deeply. Standardization method is considered a very relevant technique, so it was thought it was necessary to focus on it in wide depth.

2.2.2 Compatibility requirement

Many compatibility problems have been encountered between ML methods and our two-dimensional dataset. Therefore, in order to cover all possible ML possibilities, two types of datasets have been developed.

Firstly, and as mentioned before in the data description section, the data has been compacted to have 60 numerical values for each feature for each cycle.

In the first dataset what has been done is to create an array per cycle where all the data of all the features have been introduced, to end up obtaining a total of 17*60=1020 array elements per cycle. This gives us a two-dimensional dataset of shape (2205,1020). It is with this first dataset that compatibility problems have arises and that is why it has been proceeded to structure the data in a different way to have a bigger compatibility arc.

Then, regarding the second dataset, we wanted to create a three-dimensional dataset, more elaborate and therefore more costly, but with the aim of being able in this case to keep the values of each feature separate in each cycle. In this case, the shape marked by this second dataset is (2205,60,17). As specified above and as can be seen in the shapes, the 2205 corresponds to the number of cycles of study, 60 to the values for each feature and 17 the amount of features.

As an additional comment, it can be said that first we have been working with the first dataset, the 2-dimensional dataset, but during the process of collecting the results, it was realized that the results could be optimized by investing time in the elaboration of a second, more complex dataset, but with which it was seen that more things could be covered, such as a greater compatibility with machine learning models.

2.3 ML Techniques

2.3.1 Logistic Regression

Logistic regression is a "classification algorithm. It is intended for datasets that have numerical input variables and a categorical target variable that has two values or classes." [8] In this case multinomial logistic regression is used because each of our label has more than 2 classes. Stability label can be exclude because it has 2 classes.

Below, the theoretical function that the multinomial logistic regression method works with is presented. This method constructs a linear prediction function by calculating weights that are combined in order to understand the relationship between the input (features) and the observation (categories of each label).

$$\text{score}(\mathbf{X}_i, k) = \boldsymbol{\beta}_k \cdot \mathbf{X}_i,$$

$\boldsymbol{\beta}_k$ is the weight vector that correspond to the k output
score($\mathbf{X}_i$, k) this score is associated with the observation i of the category k
[9]

Ultimately the prediction comes out of the maximum score generated in the analysis. Also to be considered, is that in this model the score can directly be converted to a probability value, indicating the probability of observation *i* choosing outcome k given the measured characteristics of the observation.

2.3.2 Decision tree classifier

Decision tree analysis is a "supervised machine learning method that is able to perform classification or regression analysis. At their basic level, decision trees are easily understood through their graphical representation and offer highly interpretable results" [10].

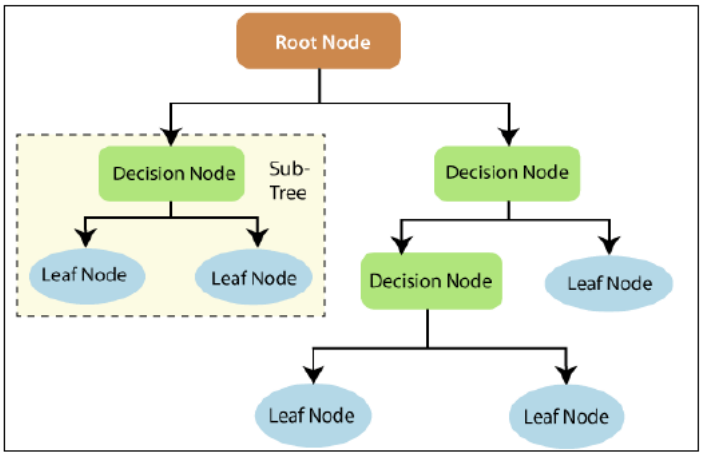


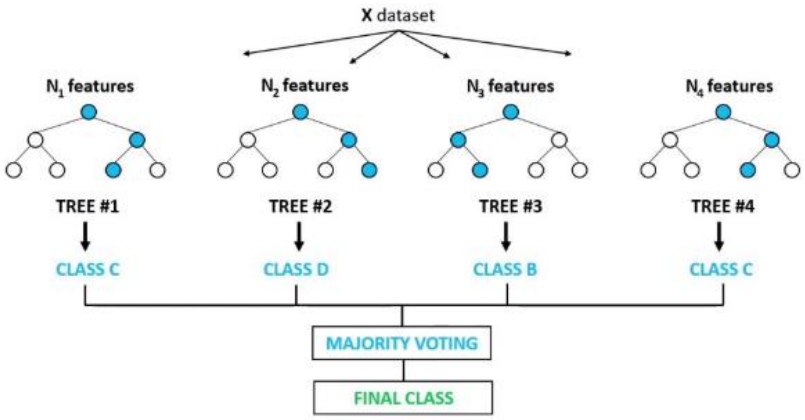
Diagram of a decision tree elements [10]

The nodes and branches are the elements of each tree. Each node represents features in a category to be classified and each subset defines a value that can be taken by the node. In more detail, the basis of a decision tree is the root node. From this initial node flows a series of decision nodes that describe decisions to be made. From the decision nodes flow the leaf nodes that represent the consequence of the decision of the decision nodes. That is, each decision node represents a point of creation of two paths and it is the leaf node that represents the answer.

2.3.3 Random forest classifier

According to Breiman, “a random forest is a classifier consisting of a collection of tree-structured classifiers h(x, k), k = 1,... where the k are independent identically distributed random vectors and each tree casts a unit vote for the most popular class at input x” [11]

Each decision tree inside the forest, is created independently from random subset of data. At the end, thanks to the process of majority voting, the trees that carry an error are remove from the equation, making the system more robust and leaves out overfitting problems.



Scheme of random forest algorithm [12]

2.3.4 Neural Networks

Recurrent Neural Network is recurrent in nature as it performs the same function for every input of data while the output of the current input depends on the past one computation. After producing the output, it is copied and sent back into the Recurrent Network.

Long Short-Term Memory (LSTM) networks are a modified version of Recurrent Neural Networks, which makes it easier to remember past data in memory. This type of neural network makes a decision of taking or not in consideration the current input and the output that it has learned from the previous input.

The different parameters that will be used to build the different neural networks with which we will work are described below.

The error function is one of the elements of the neural network, basically it shows the difference between label prediction and expected label. There are different functions to evaluate this error.

Firstly, there is the binary crossentropy function, suitable as its name suggests for binary output computations. This function uses the estimates and the actual values to calculate the error value. This function will be used when performing multilabel classification analisis, as the outputs will be binarized. In the following, the function is described mathematically.

$$L = -(y_i \; log(\hat{y}_i) + (1 - y_i) log(1 - \hat{y}_i))$$

The categorical crossentropy function, on the other hand, is ideal for multiclass studies. It multiplies the actual output with the logarithm of the model prediction for each class. In this case we will use a more complete version which is the sparse categorical crossentropy since in our multiclass analysis our outputs are in integer format, they are not binarised. The following algebraic expression describes the error function.

$$L = \sum_{j=1}^{M} y_j \log(\hat{y}_j)$$

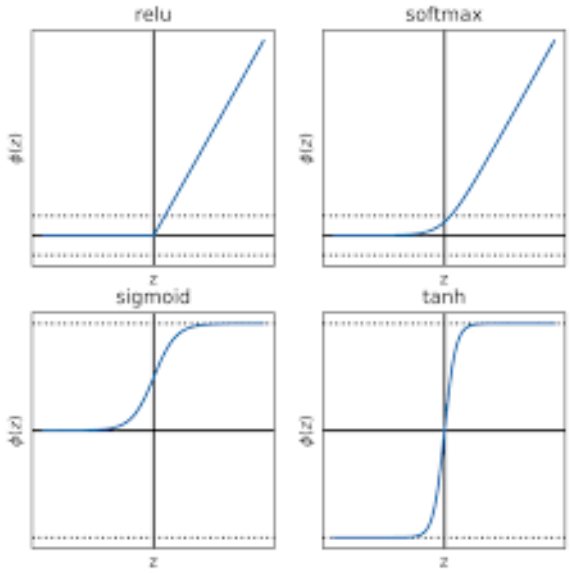
In some networks we will use the batch size parameter to limit the analysis by packages. The amount of data analyse in this project is very large, so we will choose to use the batch parameter to organise and compact the data by packages so that it can be analysed correctly.

Adam and RMSprop will be the two optimizers of neural network used in this project. The first optimizer is Adam. It efficiently computes according to stochastic gradient descent-methods. Adam combines the best properties of the AdaGrad and RMSProp algorithms to provide an optimization algorithm that can handle sparse gradients on noisy problems. This optimizer will be mainly used in multiclass classification.

For multilabel classification networks RMSprop will be used, because when dealing with complex and large amount of data RMSprop is the best option since the gradients of very complex functions like neural networks have a

tendency to either vanish or explode as the energy is propagated through the function, and with this type of optimizer, this is avoided. Basically, it uses a quadratic mean of the gradient to normalize it. It thus creates a balance and, as mentioned before, avoids gradient explosion.

Activation functions are who decide whether the neuron will be fired or not. They are used for propagating the output of nodes from one layer to the next layer.



Activation functions [13]

In this investigation the 4 activation functions plotted above will be used. The Relu function gives an output x if x is positive and 0 if it is not. The activation function softmax will be used only for the multiclass classification problem. This function converts a vector of dimension K into another of dimension k by applying the softmax function, leaving a working interval from 0 to 1. The sigmoid function rescales the inputs to (0,1) and is one of the widest functions used today. Finally, the tangent function, which will rescale the inputs this time between -1 and 1.

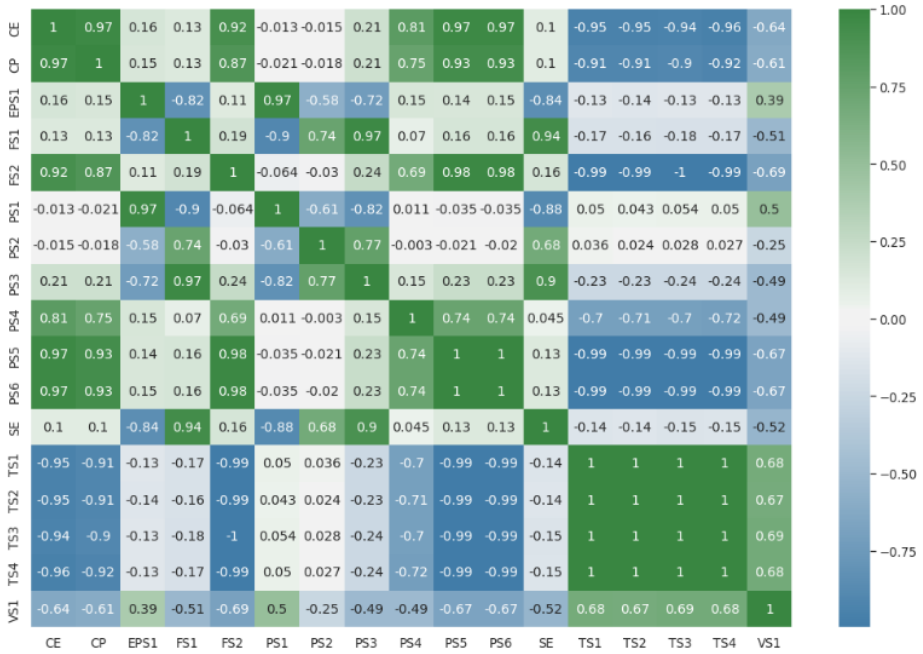
3 Data Insight

3.1 Data Correlation analysis

In this section a correlation study is presented. The aim is to show the correlation between features, between labels and between all of them. Below, it is attached different graphs in which the different correlations are shown. Also, in this case and as can be seen in the different correlation matrices, there are many positive correlations, i.e. a correlation between elements that mimic their behavior, and on the other hand there are also negative correlations, which are those that show a correlation where one element acts inversely to the other element, i.e. if one element has an upward evolution, the other will have a downward evolution.

The coefficient that has been used is the Pearson factor, which measures the strength and direction of the linear relationship between two variables.

Before starting, it is important to know that in this project there are two subsystems, the working circuit, and the cooling circuit. In each of them there are different features placed and surely the features of the first circuit will have more correlation between them than those of the second circuit. However, in this study this aspect it is avoided, and the goal is to reflect the global correlation between all the parameters, no matter which part of the system they belong to, as this analysis is considered to be very beneficial.



Correlation features matrix

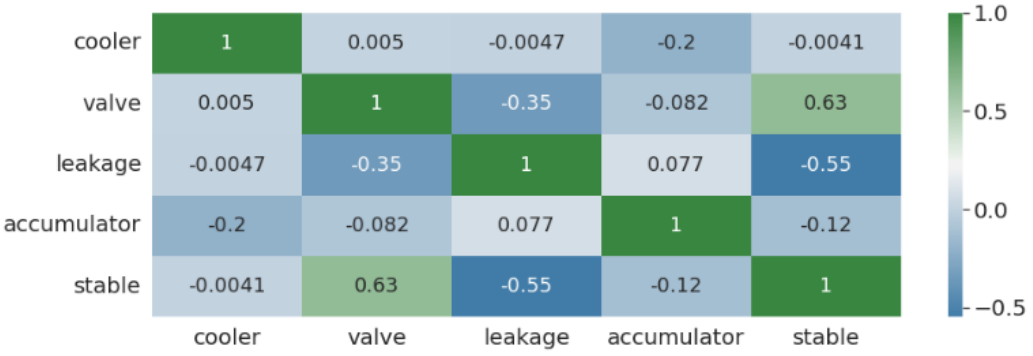
In the case of the features, as can be seen, the ratio of 1 between the temperature sensors stands out in the graph. This makes us think that it is not necessary to have 4 temperature sensors in the system if in the end the same information is received from them. However, as we will see later, none of them will be left out for the classification analysis.
Likewise, as expected, the cooler efficiency and cooler power features are highly correlated, reaching a correlation of 97% in the Pearson scale.

Moreover, our cooler efficiency (CE) feature has correlation with many features but almost no correlation with the system efficiency sensor (SE). This feature, which we consider quite important, also obviously has a bad correlation with the cooler power parameter (CP).

Another interpretation of the correlation is the one that pertains to the EPS1 parameter. This parameter, which indicates the motor power, does not have a very relevant correlation with the other features, but in this case, it does have a significant negative correlation of 84% with the SE parameter.

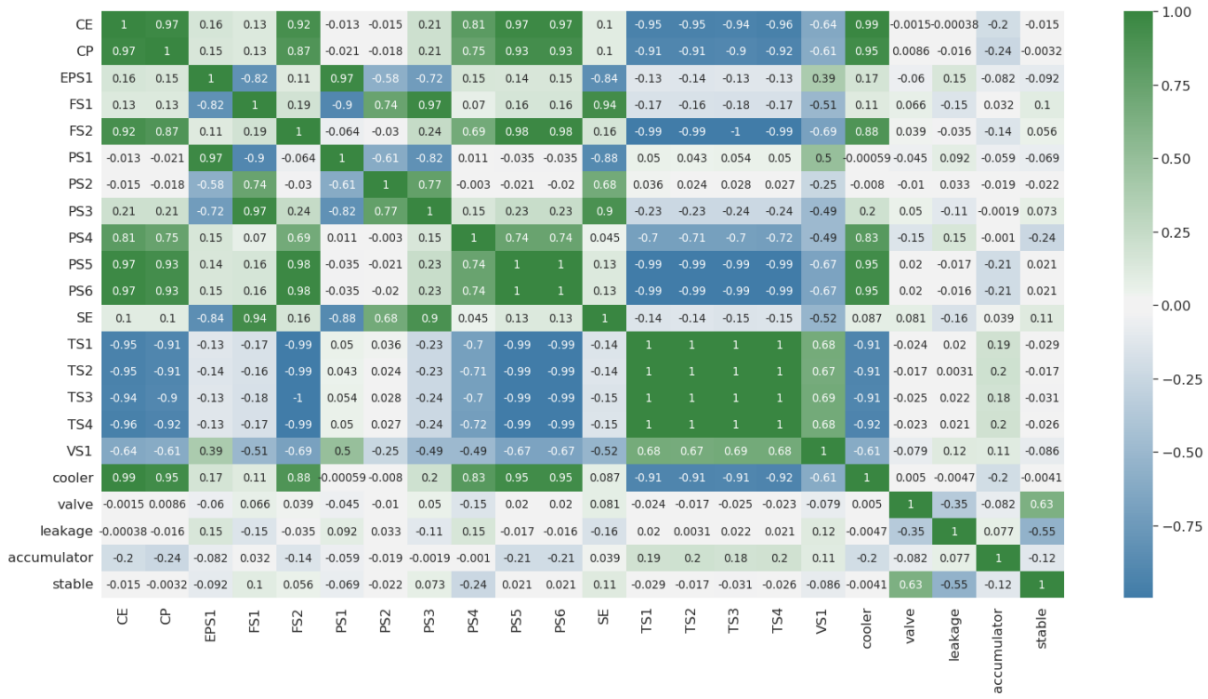
About the two volumetric flow sensors (FS1, FS2), there are differences between them, as each of them is in one part of the system. However, what is interesting is that the FS2 has a strong negative and positive proportional relationship with the 4 temperature sensors, and these sensors are not only in the cooling circuit but also in the working circuit.

In the case of the pressure's sensors, these are organized as the first 3 are in the first circuit and the other 3 in the refrigeration system so the correlations of them are divided into two groups. The first 3 pressures do not follow a very similar pattern but on the contrary, it is found that P5 and P6 follow the same pattern.



Correlation label matrix

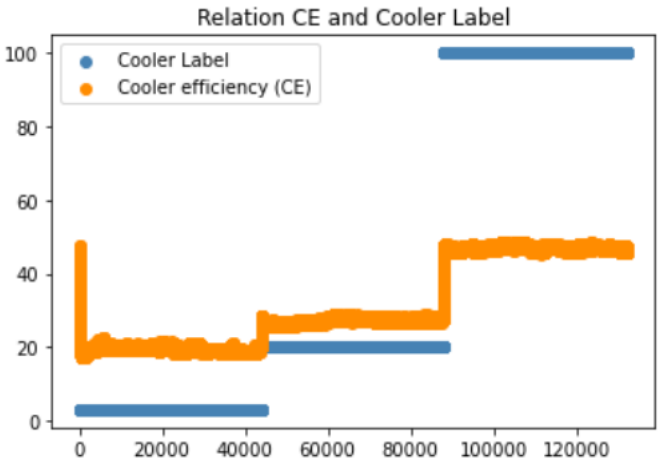
Analyzing the correlation between labels, it is realize that working with this differentiation of labels is absolutely necessary, as the overall correlation between them is practically non-existent. However, it can be seen that the valve parameter has a 63% correlation with the stability parameter and the pump label also has a negative relationship with the stability of the system.



Global correlation matrix

To end, an overall table with all correlations is shown, in order to see if the labels have significant correlations with the study features. The information is extracted from the right side of the table. It is seen how the cooler label has a high correlation with different features such as CE, CP, PS5, PS6 and temperatures in comparison with the other labels where the correlation with features is very low.

In addition, an example of a very nice correlation between the cooler label and one of the features is shown in the form of a plot.



Cooler efficiency and cooler label comparison.

It is seen how cooler label follows the same path as the CE feature.

After assessing the correlation between the study elements and seeing that there are some very marked correlations for some of the features but that overall, there are no very specific correlations, it was decided not to compact the data, i.e. no feature has been removed.

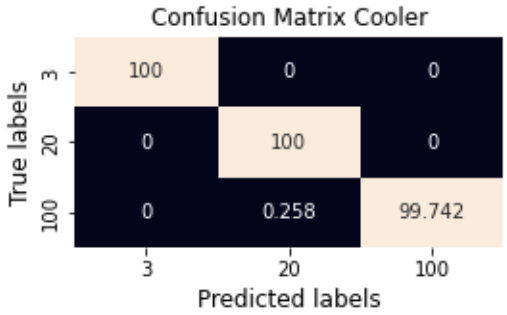
Furthermore, this analysis has allowed us to learn more about our study elements (features and labels) and has helped us to have more clearness in the dataset. Also, we are working with a fairly large number of features, and it is important to know the overall relevance of each one of them. On the other hand, this analysis prior to the classification investigation has been positively valued, as it has provided clarity about the labels and, as mentioned before, a global view of all the elements considered important in this project.

3.2 Multiclass interpretation

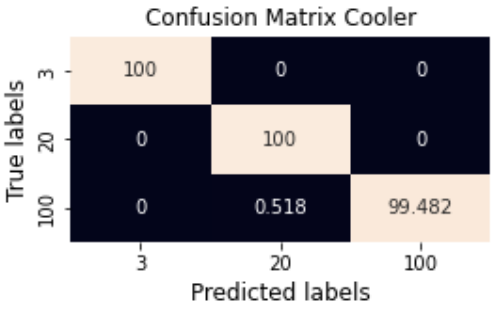
3.2.1 Multinomial logistic regression vs Decision tree classifier

In the area of machine learning, a confusion matrix is a table that describes the performance of a classification model. Each row of the matrix represents the prediction results, and the columns represent the actual outputs.
In this case, it was wanted to represent the confusion matrix by calculating the weighting of success in the classification of each label category.
The results are presented such as a comparison between performance of the multinomial logistic regression model and the decision tree model, showing a confusion matrix for each label.

With the multinomial logistic regression method, we have worked with the parameter of maximum iterations, testing different values of this parameter for each label in order to work with the most optimal system. On the other hand, with the decision tree technique, in this case the maximum depth parameter has been varied, which corresponds to the parameter that limits the depth of the decision tree. The results below correspond to the best results found for each label for each ML model.



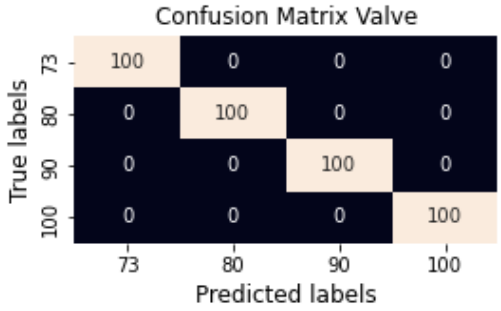
Multinomial cooler label confusion matrix



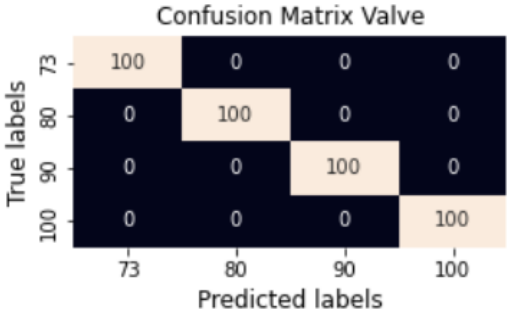
Decision tree cooler label confusion matrix

As far as the analysis of the first label is concerned, it can be seen that the multinomial logistic regression method is better than decision tree at classifying the sub-categories of this label. For categories 3 and 20 the two models compute a perfect classification, the first method obtains a 99.7% classification rate in category 100 while the second method only reaches 99.4%.

In addition, category 100 is the one where the cooler element is at its maximum functionality and as for the number of each sub-category, there are 732 elements for category 3, the same number for category 20 and 741 for category 100. This difference of approximately 10 samples may be the cause of the error in classification performance.

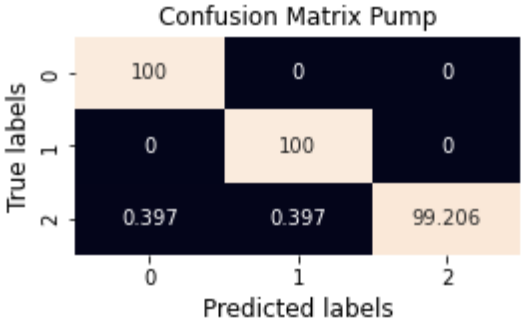


Multinomial valve label confusion matrix

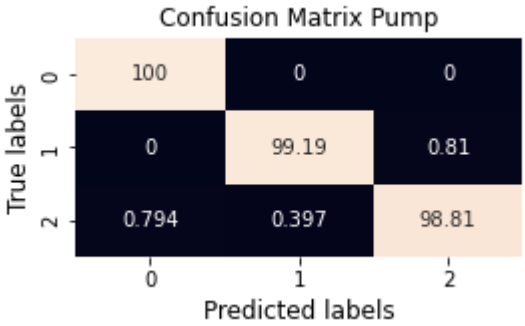


decision tree valve label confusion matrix

In the case of the valve label, it is seen that the classification made by the first and second method is perfect. It would be interesting to look with k folds cross validation the veracity of the results.

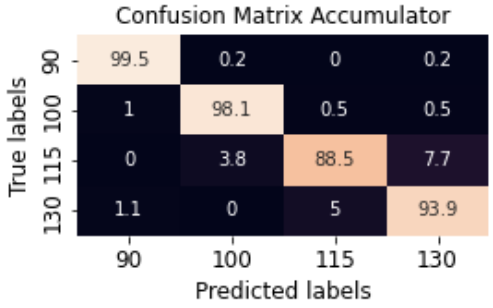


multinomial leakage label confusion matrix

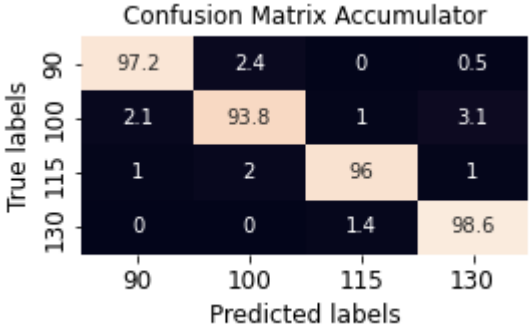


Decisión tree leakage label confusion matrix

Compared to the other labels, more classification errors are highlighted here. In this case, the first method is again the one with the best classification prediction results. In addition according to the decision tree method, it can be seen that this time there are already 2 pump categories that the method fails to classify perfectly. It is interesting to see if the weights that are distributed within the leakage label have any relationship with the classification results shown for the label. In the case of leakage, the samples are distributed in such a way that category 0 (which is equivalent to the maximum health) groups a large number of elements, 1221 to be exact, while the other two categories have the same number of elements, around 492 cycles. It can be said that the algorithm method has enough information to classify the “0” category, but for the other two other categories, maybe 492 cycles are not enough to reach a perfect classification.



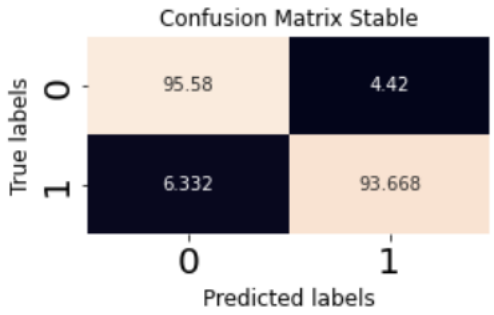
Multinomial accumulator label confusion matrix



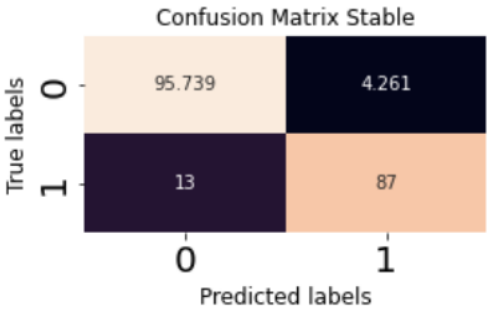
Decisión tree accumulator label confusion matrix

Accumulator label is the one that gives the worst results. If the general accuracies of the decision tree method are compared, it is seen that the label cooler, valve and leakage have very high accuracies of approximately 0.99, while the accumulator category has an accuracy of 0.967.
The only aspect that can be observed at a glance is that, in the case of the distribution of cycles for each label, for valve the distribution between the 4 categories is uniform, i.e. the high category (100) has 1125 cycles and the other 3 categories are equally attributed about 300 cycles. On the other hand, for the accumulator label, the categorical distribution is not very uniform. Firstly, the maximum category value, which in this case is 130, does not cover the majority of samples, and secondly, the other categories do not distribute the cycles equally. Another observation is that the most unfavorable category of the accumulator, label 90, which would be associated with close-to-failure accumulator elements, has the highest number of cycles. It would appear that accumulators do not perform as well as expected in general. Definitely this manifests that for this type of element our results are far from the norm. Finally, it has been observed that this label is the only one in which the performance of the first method is worse than with the second. This could be due

to the fact that, as mentioned above, the uniformity of this label is not as parametric (a characteristic of the multinomial logistic method) as in the other labels.



Multinomial stable label confusion matrix



Decision tree stable label confusion matrix

Once again, the multinomial classification model achieves better results than the decision tree model. Above all, it can be seen that for stability category 1, which corresponds to a state of instability of the system, the second method fails to classify 13% of the cycles in this category, as it categorises them as stable cycles. Compared to the multinomial logistic regression model, the error rate is halved. However, as far as category 0 is concerned, that of system stability, decision tree manages to classify more % of cycles correctly, since if the percentages of success is compared in this category it is seen that in decision tree they are 95.739% and in contrast with the other method 95.58%. It is a small difference but relevant to reflect it.

ML model	**Cooler**	**Valve**	**Leakage**	**Accumulator**	**Stable**
Multinomial Logistic Regression	0.999	1	0.998	0.957	0.949
Decision Tree classifier	0.998	1	0.995	0.967	0.925

Accuracy table

As can be seen with our accuracy results all labels have very favorable classification results as the minimum accuracy is 0.925, i.e. results in the 90% range. With this overall conclusion, it will be interesting to see if better results can be obtained with other ML methods or with other multi-label classification techniques.

On the other hand, and with the aim of validating the results, a k-fold cross validation analysis has been carried out, based on the fact that the process of splitting between training and validation data has been by means of cross-validation, we wanted to go further by carrying out a study with 7 data partitions.

The cross-validation technique is a statistical method that allows the evaluation and comparison of learning algorithms by dividing the data into two segments. One of the segments is used to learn or train a model and the other part is used to validate the model.

An interesting part of cross-validation is k-fold cross-validation, which allows for more movement of the data. In k-fold cross-validation, the data is partitioned into k segments, also called folds. The variable k is also the one that gives the number of iterations of training and validation data to be computed. The main idea is that in each iteration different data segments are placed in the validation segment. [14]

In the project we are going to develop these k-fold cross-validation studies specifically in the multiclass part for the multinomial logistic regression and decision tree methods in order to avoid possible overfitting.

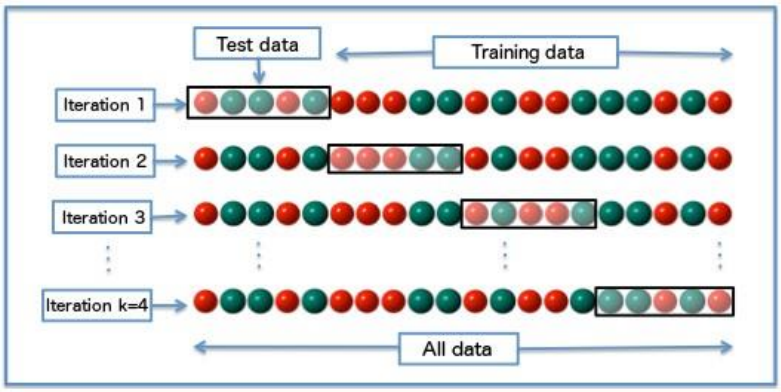


Diagram of 4-folds cross validation. [15]

In the case of the image above, we would be dealing with a case of binary classification between red and green balls. The case study of the project is more complicated but follows the same standards reflected in the image.

ML model	**Cooler**	**Valve**	**Leakage**	**Accumulator**	**Stable**
Multinomial Logistic Regression	0.979	0.982	0.971	0.825	0.866
Decision Tree classifier	0.977	0.999	0.978	0.705	0.848

K fold cross validation results.

The table shows the average accuracies resulting from the k=7 folds cross validation analysis taking our entire dataset, always separating by labels and doing the same study for each one of them. As can be seen, the results seem acceptable as the accuracies are not very far from the maximums found in the multiclass study. However, for the labels accumulator and stability it is seen that the accuracy mean of all k cross validation is very low and might be because the two labels face overfitting problems. In particular with decision tree classifier method, accumulator label present a 70,5% mean. Considering that the maximum accuracy reach for this label is around 96%, it is seen thanks to k folds cross validation, that the possibility of low accuracy of label accumulator decision tree classifier model exist, that reveal some issues according to this particular label.

Overall, all the other results leads to the veracity of the results of our first analysis.

3.2.2 Multiclass with RNN

Next, another multiclass analysis has been carried out, this time using neural networks. As the area of deep learning is very extensive due to the wide range of techniques, parameters and types of neural networks, this analysis has been focused on recurrent neural networks of the long short time memory type, as mentioned in the previous chapter. Keras, an API for implementing neural networks, has been use for analysis involving deep learning.

An optimization study of the network parameters has been done based on trial and error, in order to obtain the most effective set of recurrent neural networks possible to tackle the problem. Valve has been used as a reference label in the analysis, since, as has been verified in the first analysis of multiclass classification, it seems to be an output with good performance for its excellent classification of categories results.

On one hand, it is important to analyze what the batch_size parameter contributes to the performance of the network, as well as what is the optimal test size and also and not least the influence of the depth of LSTM parameters as to which optimizer is better. All neural networks sets have been evaluated with 100 epochs and with the same loss factor, in this case sparse categorical crossentropy.

It is important to explain that in this second part of the multiclass analysis the three-dimensional dataset will be used and that in the previous analysis the other dataset was used. All this to fulfil the maximum range of ML methods and to comply with their dimensional compatibilities.

Below is a table showing 6 types of RNN with description of parameters for each one.

	Batch_size	Test size	LSTM 1	LSTM 2	optimizer	Accuracy /val accuracy
1	64	0,50	128	128	RMSprop	0.778 / 0.448
2	64	0,50	128	128	Adam	1.0 / 0.757
3	32	0,75	128	64	Adam	1.0 / 0.642
4	32	0,75	128	64	RMSprop	0.943 / 0.539
5	64	0,75	128	128	Adam	1.0 / 0.593
6	80	0.50	264	128	Adam	1.0 / 0.628

RNN description

Networks 1 and 2 have been tested and it has been found that Adam obtains better results.

From there, in networks 3 and 4, some parameters will be changed, and the two optimizers will be compared to validate that Adam is the better optimizer overall.

Thanks to the elaboration of networks 3 and 4 it can be concluded that Adam is better than RMSprop, since Adam covers the performance of the other optimizer. Another difference between optimizers is that RMSprop uses the momentum parameter of the rescaled gradient to generate learning updates within the network, while Adam, on the other hand, produces updates directly from an estimate generated by the momentum of both the first and second gradient momentum.

On the other hand, networks 5 and 6 have been created to try to optimize neural network 2, as it is the one that has given the best accuracy results. However, it hasn't been possible to increase the efficiency, since even though we have provided it with more data in the test segment, the validation accuracy does not improve.
In short, from what the accuracies of all the sets of networks that have been elaborated show, we would be dealing with a case of overfitting. In this case, it is evaluated that the networks do not acquire enough knowledge to be able to provide high classification results from the validation test. This phenomenon is quite common in the area of machine learning because what happens is that the network in question creates such a specific learning algorithm for the traning part, as seen with the accuracies of 1, that when the network is tested with the validation part, the algorithm is not able to achieve the same results.

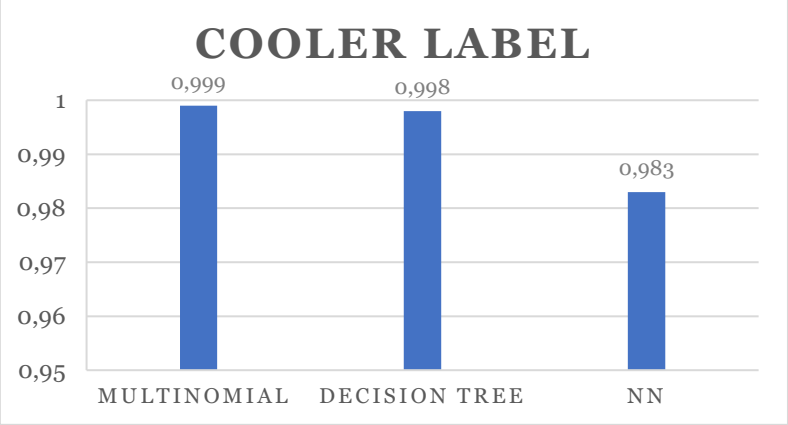
label	Accuracy of NN	Validation accuracy of NN
COOLER	1.0	0.983
VALVE	1.0	0.757
LEAKAGE	1.0	1.0
ACCUMULATOR	1.0	0.588
STABLE	0.987	0.854

The table above shows the study of the RNNs, for each label separately and, as expected, several labels with overfitting problems are revealed. Firstly, there is the valve label, whose overfitting problem has already been mentioned, followed by the accumulator label, with which the overfitting is much more severe, as it goes from a training accuracy of 1 to a very poor

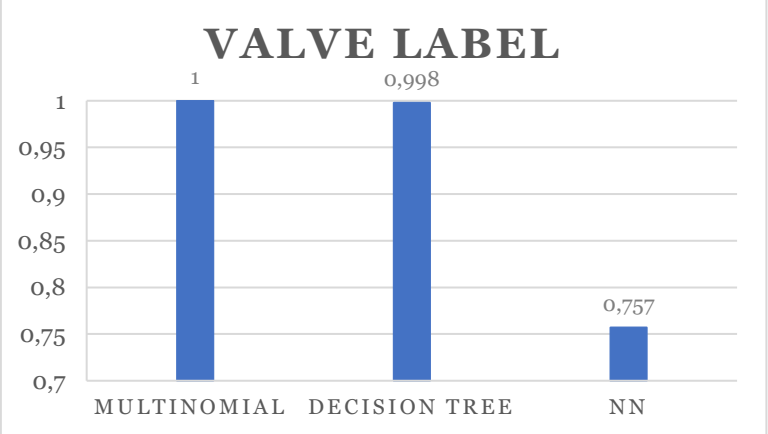
classification of the test data of 0.588, i.e. only 58% of the cycles in the test segment are correctly classified in their accumulator sub-category. As a positive point of this analysis, apart from the overfitting problems, the results for the labels cooler, leakage and stability are very positive. For example, the leakage accuracy result is a perfect classification, better than the one obtained previously with the other models.

It is important to emphasize that in this project the level of depth computationally with ML methods is not wide, which raises the question of whether a more in-depth and detailed analysis of RNNs would have given better results.

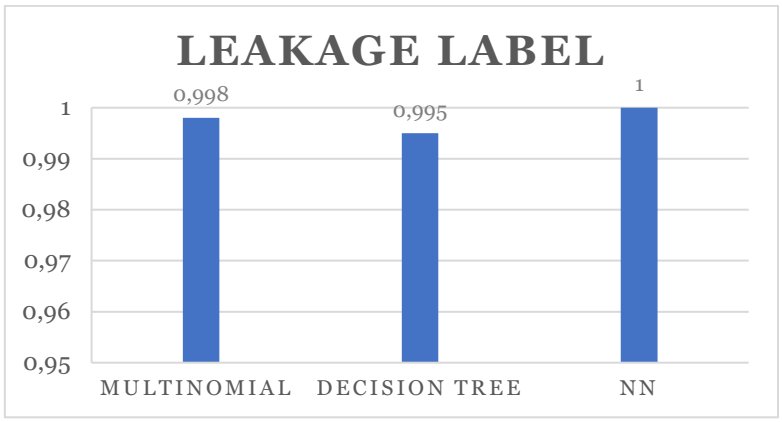
Below are graphs that help to describe the multiclass analysis results more clearly comparing ML methods used.



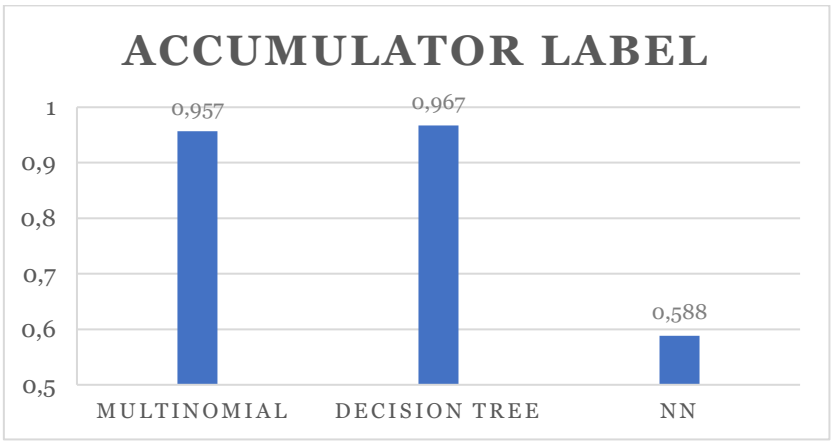
Cooler label multiclass analysis results



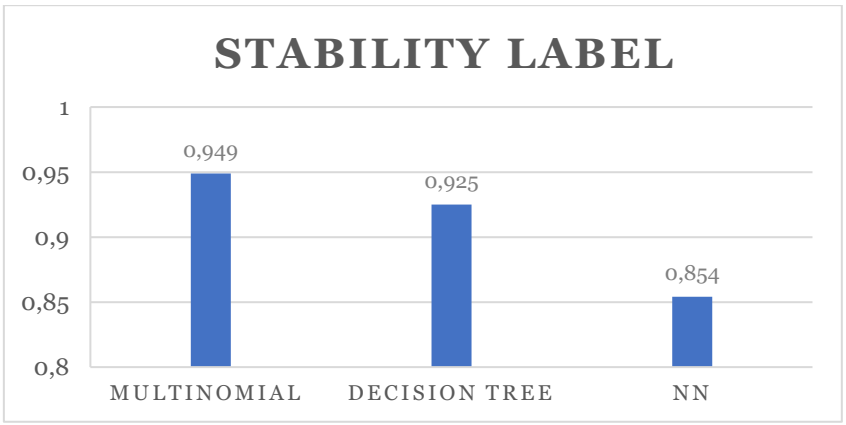
Valve label multiclass analysis results



Pump label multiclass analysis results



Accumulator label multiclass analysis results



stable label multiclass analysis results

As an overall conclusion of this section and as can be seen in the graphs of each label, it can be said that the most efficient and ultimately the best model

for multiclass analysis is the model derived from logistic regression, or rather the multinomial logistic regression model. Since the concept of overfitting makes the RNN analysis shaky and also thanks to this effect it becomes clearer how more standard methods are more suitable for the problem.

3.3 Multilabel classification

3.3.1 Multioutput classification

Once the multiclass analysis has been completed, an analysis has done using the multioutput classification technique. This method is a hybrid way between multiclass classification and multilabel classification. The objective is to predict multiple outputs simultaneously given different inputs. [16] This analysis has been carried out because the sklearn library addresses this aspect in the area of multi-label classification. Besides, it is recommended to use it for problems with large databases and in our case both the features and the number of sub labels follow this particularity.

For this analysis, the two machine learning methods that are used are the decision tree method and the random forest method. We wanted to use decision tree model again because it seems to be a good technique for multiclass classification, and we wanted to analyse whether this more advanced technique also provides relevant results. On the other hand, we opted for the random forest method, which is a broader version of the decision tree method, as in this case different trees are analysed and classified according to the majority of votes. In this section, we have sought to compare and analyse results between this two ML methods.
Furthermore, we seek to analyze the importance of standardization of features as well as the label binarization technique on labels. Do these data transformations help to achieve better classification results?

The results of test validation accuracies are represented in the following table:

X / Y	NO label binarizer	Label binarizer	Classification model
no strandarize	0,907	0,857	Decisión tree
	0,953	0,898	Random forest
standarize	0,909	0,860	Decisión tree
	0,963	0,911	Randon forest

As can be seen from the results, the random forest method is better than the decision tree method. This is due to the fact that, as mentioned before, random forest includes the decision tree method in its internal organization, and is therefore a more complex and computationally deep method.
The best result is for the random forest method when feature standardization is applied.

Performing data standardization is positive for both methods as better accuracy is achieved in all cases. However, the difference in results is more relevant with the random forest method.
We also found that binarizing the subcategories (using the label binarizer function), reduces the accuracy. This is due to the fact that by computing the binarization of labels from 5 to 15, we increase the system outputs by 10. Moreover, when working with 15 classes to categories, the number of outputs is very close to the number of features (17) and this can be an important factor that makes the results less good as the model cannot cover such a large number of outputs.

On the other hand, analyzing the best accuracy values obtained, it can be seen that the result is 96.3% correct classification. A comparison with the results of the previous chapter shows that the results of this method are very good and must be taken into account. However, with the multiclass analysis, the time and cost invested in carrying out 5 different analyses separately is not totally refundable because, as we have seen, it can lead to overfitting problems in some of the labels.
In addition, after this first preliminary analysis with the multioutput technique, an extra analysis has been carried out in order to compare this time different test sizes and different parameters between the multinomial logistic regression model and the random forest classifier. It was wanted to present this analysis because, as we have seen previously, both multinomial logistic regression gives good results in multiclass classification and random forest has been a model with which we thought it was appropriate to carry out more tests. In the following, the different accuracies of the validation data are presented, for both models and the features have been standardized. Also, label binarization has not been used since, as demonstrated above, it does not help at all.

For this analysis, we have modified the maximum iterations in the first model and the depth of the tree in the second model.

Multinomial Logistic Regression		*Training*	*Validation*
		Max Iterations	
		800	
TEST SIZE	*0.25*	0.982	0.935
	0.50	0.985	0.908
	0.75	0.989	0.865

Random Forest Classifier		Max depth		
		5	10	15
Test size	0.25	0.849	0.954	0.963
	0.50	0.863	0.948	0.953
	0.75	0.851	0.913	0.909

The multinomial logistic regression model shows worse accuracies than the Random forest method. It is also found that modifying the maximum iterations does not contribute anything since the values of training and validation of accuracy remain the same. A positive point of the multinomial analysis is that we are not in front of a case of overfitting since, as shown in the multinomial table, the training and validation results follow a normality. Moreover, it is not shown in the second table but it has been proved that the random forest training and validation results also follow a standard path and we are not in a situation of overfitting.

On the other hand, evaluating the results of the random forest classifier, it can be observed that the optimal percentage of test data is 25%, since the best results are almost always obtained with this segmentation. On the other hand, thanks to the presentation of different results of the random forest classifier varying the depth of the method, it can be seen that the deeper the system the better. It has also been tested with a depth greater than 15, giving a value of 20, and it has been seen that with 15 the best results are obtained, i.e. there is no exponential increase in accuracy as the maximum depth parameter of the method is increased.

Another observation is that if we compare the results obtained with a random forest model of depth 5 with the multinomial logistic regression model, the latter gives better results.

In short, it is concluded that modifying the parameters of the models can provide better results, it could be said that this process of testing different parameters is a way of optimizing the results.
On the other hand, as it has been proved in the first part of the multioutput classification study, it has been validated that this model works better without label binarization, but on the other hand, it improve thanks a standardization of inputs.

In short and making a comparison between the multiclass analysis where it has been seen that the best method is multinomial logistic regression and the multioutput classification technique, it can be seen that in this type of research the best way to proceed is using the multioutput technique since this technique generates very positive insights, although not as relevant as for some labels of the multiclass analysis (cooler, valve and leakage). However, even if the multiclass analysis gives extremely good results, there is still the unknown or inconvenience of being confronted with some overfitting problems. In the case of multioutput it is only necessary to perform a study that includes all 5 labels together and we are far from any kind of overfitting problem. In addition, if the objective of this project is to find an optimal strategy for the maintenance of the hydraulic system, a fast monitoring work is preferred, been that the performance of an analysis like multioutput classification technique that perform an analysis of all 5 labels elements together. This takes less time and could be more efficient than performing a separate label analysis, as multiclass classification does, because with a short analysis very good results are obtained. If , in the case, multioutput classification hasn't shown interesting results, the strategy will be to work side to side with multiclass classification using multinomial logistic regression.

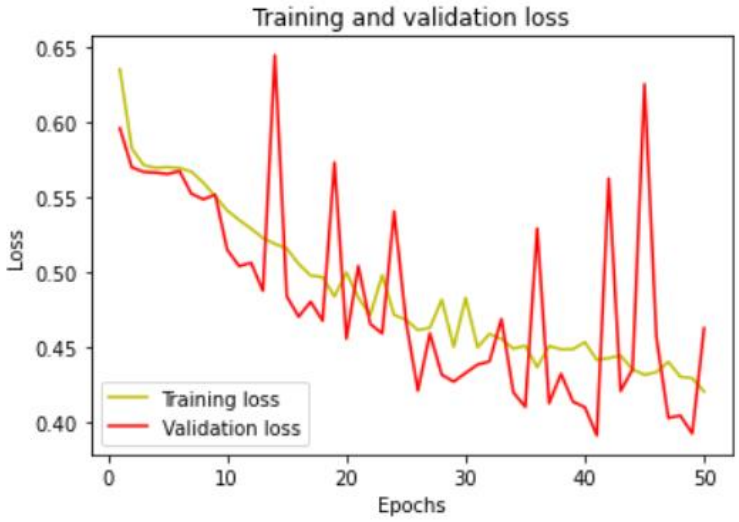
3.3.2 Multi-label classification with RNN

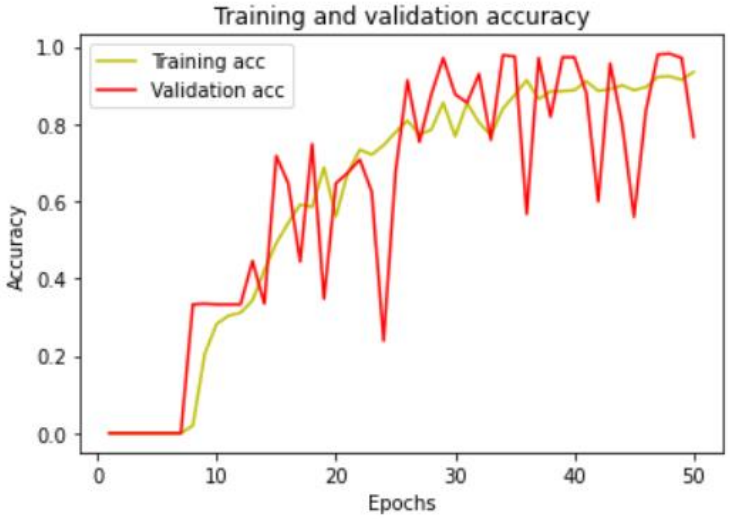
In this section, different studies have been elaborated with different neural networks in order to obtain competitive results to compare with the other methodologies used before, and to this end, innumerable adjustments have been made to the different sets of neural networks to achieve it. It should be noted that we start from a level in which the analysis of multiclass with RNNs has presented some overfitting problems, which this section will try to avoid. Although it is a challenge due to the complexity of our data, it is wanted to use this type of deep learning methodology once again as it covers many parameters and study elements that, it is believe will be favourable for tackling the problem of the project.

The two most relevant results, which it is believe can contribute to the research, are presented below. However, although only two results are presented, many more have been carried out, but it has been decided not to present them due to the lack of relevance in the results. Furthermore, these different elaborations that are not relevant, will be explained and it will be clarified why they have not been included in the project.

The first neural network shown below is composed of 2 hidden layers of LSTM category with 64 nodes each and a final hidden layer of 16 nodes with Relu. Finally the output layer has 15 units (as output number) and has sigmoid activation function. Furthermore, the RMSprop optimiser is used and the binary crossentropy as loss function. The other parameters not described here are default parameters.

Below is the loss and accuracy graph with both training and validation results:





As a first observation, there are many peaks and many irregularities. The training segment is much more stable than the validation segment. This is due to the fact that at some times the difference between the accuracy of the two data segments is an overfitting. It can also be noted that in the lost graph

the peaks are more pronounced. The only positive point is that a good level of accuracy is reached at the end of the model.

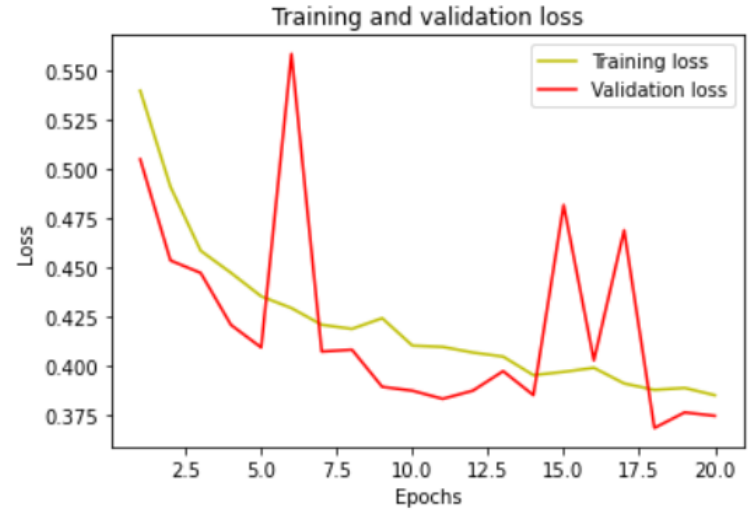
It is worth mentioning that in order to reach this kind of results, tests have been made, where several parameters have been changed, for example the optimizer to Adam, and the results in those cases were much worse, even giving 0 accuracy, due to a gradient explosion. Also the increase or reduction of epochs did not help to improve the results.

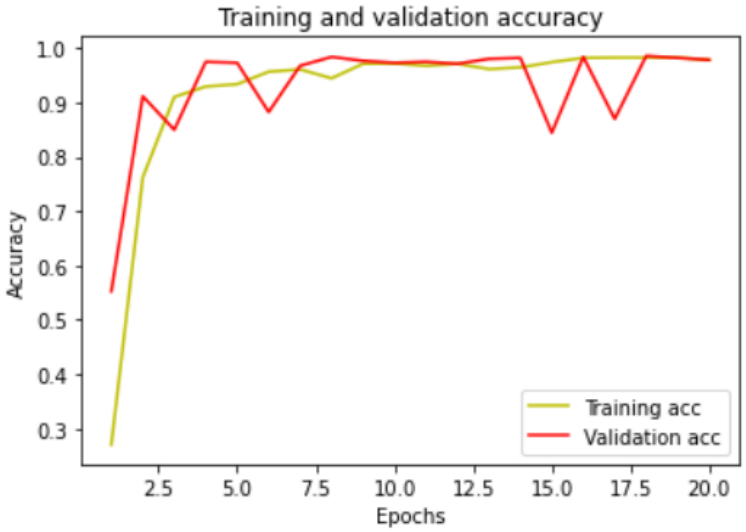
Through these results we have tried to generate another neural network with the intention of eradicating the peaks and finding better parallelism between training and validation, in short, a better model than the one presented.

After several tests where the activation functions were changed and/or the nodes in the LSTM networks and in the dense network were increased. No better results could be achieved.

This is why the shape of the feature dataset has been changed. That is to say, as has been commented during the project, two types of dataset have been elaborated, having the same information in both, but of different shapes. In the first step of the analysis of multilabel classification with RNN, the most complex network was used, given the complexity of the network method, i.e. the three dimensional dataset. However, in view of the results, we opted for the first dataset (which is two dimensional) in order to use it and see if this time, this dataset could improve the results. A reshape process what done before performing the RNN since it accepts only three dimensional datasets.

Below are the results of the analysis, where the input_shape has been 1020, because in this dataset all the features for each cycle have been put together in a single np array. In this case the batch value has been reduced, because in this case the input enters directly with 1020 elements, it has been decided to use a batch as a method of separating these data every 10 values. In addition, the hidden layer between the LSTM and the final output has been removed.





As can be seen, these results are much better than the previous ones. A close comparison of the training data shows that there is much more stability and uniformity with this second set of neural networks. As far as the validation set is concerned, it can be seen in this case that the peaks continue to persist, but in this case they are less pronounced and there are fewer of them.

On the other hand, in this second analysis we work with 20 epochs and in the previous one we reach 50 epochs. This is better in terms of cost, since the fewer the epochs, the lower the cost. In addition, greater results are achieved, with one less hidden layer. However, the duration of the computation of each epoch in this second model is much longer than in the previous model.

It can be concluded from this analysis that achieving very interesting and profitable results with RNN is not an easy task. Different tests have been carried out with different RNN, always taking the LSTM type as a reference, but

it can be seen that having so many parameters to modify, make it difficult to build a perfect neural network, which can classify training data and validation data equally good. However, although the results of this analysis are not exceptional, we have managed to start from very irrelevant results, as can be seen in our first loss and accuracy graphs, to obtain at the end, with the second performance some of the best, all of this following a trial and error methodology.

Another relevant observation of the networks analysis is that very good results are obtained for the training dataset but they do not help because the algorithm learns too much from that dataset and when it is necessary to test with other data, it is when irregularities are found.

As a conclusion of multilabel classification, it can be seen how a technique that is not so common, multioutput classification, is much simpler, faster and more solvent than what is achieved using methods such as RNN, which are very complex and in this case, have not managed to perform their job completely, as better results are achieved with multioutput.

4 Summary

4.1 Conclusion

In this project we have started using data from a hydraulic system, from which features have been extracted thanks to the catchments of different sensors such as temperature sensors, pressure sensors and other virtual sensors such as cooler efficiency. All these features have been extracted with different sampling rates, and it has been necessary to resample them in order to have a more compatible input data. On the other hand, we have worked with 5 labels, cooler condition, valve condition, accumulator condition, pump leakage and system stability, each of them marked by internal categories that describe the health state of the parameter.

With all this data, a correlation study was performed in which a fairly segmented correlation between features was observed, marked by two groups in which features were highly correlated with the cooler efficiency, covering the parameters of temperatures and cooler power, and on the other hand other parameters of pressure and flow volume and the efficiency of the system. It was found that these correlations could be related to the location of the different features in the system, although for example the 4 temperature sensors have a correlation of 1 and are distributed in different parts of the system. Furthermore, with regard to the correlation between labels, not much relevant information has been highlighted, and regarding the correlation between features and labels, a very relevant correlation between cooler efficiency and cooler label has been plotted.

Once this analysis was completed, we started to perform ML models, first by performing a multiclass classification study, where each label was tested separately, in order to try to classify its health categories. Multinomial logistic regression and decision tree classifier were used. By means of this analysis, very good results were obtained for all the labels, with results of more than 90% accuracy of the validation set. Next, another multiclass classification study was also carried out, but this time using RNN of the LSTM type. With this method, many overfitting problems have been found in valve and accumulator labels, and it has been concluded that between standard methods (multinomial logistic regression and decision tree classifier) and neural networks, for this research, it is more optimal to use simpler methods, since confronting overfitting is a complex and very costly task.

Two types of multilabel classification were then developed, one using the multioutput classification technique and the other using RNN. As a general comment on this second analysis, very promising results have been obtained with the multioutput technique, using the ML methods of the multiclass section and adding the random forest ML method with which relevant results have been obtained. However, the analysis with neural networks has been quite challenging, as it was based on multiclass results that had already presented overfitting, and in this case, as it was necessary to carry out a more extensive study, an attempt was made to eradicate overfitting and obtain better results than those obtained with multioutput classification, but this was not perfectly achieved. In short, the neural network methodology is optimal when a lot is known about it, when the network understands from the beginning how the dataset works, and in this case, although innumerable modifications have been made to optimize the performance of the neural networks, this technique has left much to be desired.

In short, and in response to the research question introduced at the beginning, in this case it has been very beneficial to carry out a separate study for each label, in the end, as has been seen in the correlation study, these labels do not have much correlation between them, and this means that they generate better insights if they are analyzed separately. However, although the multiclass study with standards ML models is the most relevant in this investigation, having carried out the multioutput classification study has provided a lot of clarity and good classifications. Although to be practical it is considered that for this project the best strategy if time and cost are not take into account is to carry out a multiclass study and with it address the maintenance and management of the system's health with a very complete monitoring. Otherwise, using multioutput classification would give an overall behavior of the hydraulic system as an entity and also could be a interesting strategy model to use if the intention is to have a global monitorization of the system.

On the other hand, what we have tried to analyze is whether our data follow a parameter model or not. In the first part, as better results have been obtained with multinomial logistic regression than with decision tree, the hypothesis that the parameters of the hydraulic system follow a parametric line has been put on the table, but this hypothesis has lost validity once the multioutput analysis between multinomial classifier and random forest classifier has been completed, as the latter has given better results. Random forest is a non-parametric model, as is decision tree classifier.

Finally, to answer the last question on whether it has been a good strategy to produce two features datasets of different dimensions. In our case, and as reflected in our last multilabel classification analysis, using different datasets has helped enormously because it has allowed us to obtain better analysis

results and at the same time it has been key since, by using different ML techniques, we have faced compatibility problems in which the only solution to solve them has been to reshape datasets with different dimensions. This has allowed us to carry out a much broader and enriching investigation.

As a last point, if one had to be practical in order to solve the problem in the most efficient and least costly way, once all the results of the different analyses were collected, it can be said that the label cooler is a very relevant label for hydraulic systems, as it has never given overfitting problems, and has always been at the top of the highest accuracy results. If one had to evaluate which of the labels follows a straight line and does not give complications, one would undoubtedly choose the cooler label. The health of all the hydraulic system could therefore be monitored by just analyzing the health state of this particular label.

4.2 Project Limitation

One of the limitations encountered is the subject matter of the project. Hydraulic circuits are not very well known and are not a very popular topic in ML analysis and specifically in multilabel classification analysis. In addition, we have worked with a rather complex database. On the one hand, the sensors, to which different processes had to be applied until their information could be used, as they had an initial sampling rate that was not compatible with ML research. On the other hand, in relation to the labels, we are dealing with a multivariate label system, i.e. it is not that there were 5 labels but that each one had its own categories, which increased the difficulty of the analysis.

On the other hand, having such a large and complex database has led to different compatibility problems with some ML methods, which was not expected and had to be corrected in order to be able to drill into the largest number of ML models. In this way, it has been realized that having a complicated database required deep computation, specifically for neural networks. It has not been possible to perform high computational developments as much as necessary due to the lack of programming skills, which have seen were necessary to cover this type of complex problem.

On the other hand, it has been seen that some problems of overfitting have appeared, but it is understood that with the complexity of the system, this type of problem are more common to appear.

4.3 Further Investigation

As possible future research, it is proposed to perform the same ML analysis but with all the data of the system. This would mean instead of compacting the features to 60 values per cycle, to make an increase and create bigger data size, having more data for each feature per cycle. This data augmentation would help especially in the RNN model since, as we have seen, the system needs much more data than the ones we have been working with in order to be able to better evaluate the hydraulic system and get better insights.

Furthermore, according to other suitable ML models to use, KNN could possibly be one model for this problematic, as it analyses the behavior of the nearest points and classify itself accordingly.
On the other hand, it also seems interesting to be able to apply the supervised learning Naives Bayes model, which in simple terms, assumes that the presence or absence of a particular feature is unrelated to the presence or absence of any other feature, given the variable class.

To end, a very interesting point of future research would be to do similar ML analysis as the ones done in this project but on this occasion analyzing each subsystem of the hydraulic system separately, that means it would be a good idea to analyses the working system, with its features and on another page proceed with a ML analysis of the cooling system, with using its features. The two analysis will be working with the same outputs / label data or also assign labels for each one of the hydraulic sub-system.

References

[1] R. L. M. J. M. Issam El Naga, Machine Learning in Radiology, 2015.

[2] A. Jung, Machine Learning. The Basics., Springer, 2022.

[3] D. Dua y C. Graff, «UCI Machine Learning Repository,» [En línea]. Available: https://achive.ics.uci.edu/ml/about.html#:~:text=The%20UCI%20Machine%20Learning%20Repository,graduate%20students%20at%20UC%20Irvine. [Último acceso: 15 February 2022].

[4] C. Bishop, «Pattern recognition and machine learning,» *Springer Berlin Heidelberg,* p. 3, 2006.

[5] M. Wakorn, «Seaborn : Stadistical Data Visualization,» Pydata, [En línea]. Available: https://seaborn.pydata.org/#:~:text=Seaborn%20is%20a%20Python%20data,introductory%20notes%20or%20the%20paper.. [Último acceso: 26 February 2022].

[6] T. M. D. Team, «Matplotlib.org,» [En línea]. Available: https://matplotlib.org/(. [Último acceso: 26 March 2022].

[7] F. Pedregos y V. Gaël, «Scikit-learn: Machine Learning in Python,» *Journal of Machine Learning Research,* p. 2, 2011.

[8] J. Browniee, «Multinomial Logistic Regression,» Machine Learning Mastery, 1 January 2021. [En línea]. Available: https://machinelearningmastery.com/multinomial-logistic-regression-with-python/#:~:text=Logistic%20regression%20is%20a%20classification,to%20as%20binary%20classification%20problems . [Último acceso: 5 March 2022].

[9] W. Foundation, «Logistic Regression,» Wikipedia, 29 July 2019. [En línea]. Available: https://en.wikipedia.org/wiki/Logistic_regression. [Último acceso: 13 May 2022].

[10] C. Z. Janikow, Fuzzy decision trees: issues and methods, vol. 28, Cybernetics, 1998, pp. 1-14.

[11] L. Breiman, Random Forest, 2001, pp. 5-32.

[12] J. A. Figueroa Márquez, «Random Forest,» Rpubs and RStudio, [En línea]. Available: RPubs - A11U1. [Último acceso: 11 June 2022].

[13] «Github,» August 2015. [En línea]. Available: http://colah.github.io/posts/2015-08-Understanding-LSTMs/. [Último acceso: 17 April 2022].

[14] P. Refaeilzadeh, L. Tang y H. Liu, «Cross-validation,» *Encyclopedia of database systems,* 2020.

[15] D. Shulga, «5 reasons why you should use Cross-validation in your Data Science Projects,» *Towards Data Science,* 2018.

[16] D. Xu y Y. Shi, «Survey on Multi-Output Learning,» *Transactions on Neural Networks and Learning Systems,* vol. 31, pp. 2409-2429, 2020.