UNIVERSITAT POLITÈCNICA DE VALÈNCIA

ESCUELA TÉCNICA SUPERIOR INGENIEROS INDUSTRIALES VALENCIA

TRABAJO FIN DE MASTER EN INGENIERÍA INDUSTRIAL

ANALYSIS OF DATA AND PATTERN RECOGNITION FOR THE IMPLEMENTATION OF AN E-MAINTENANCE SYSTEM

AUTHOR: MANUEL PÉREZ BALAGUER

SUPERVISOR: GREGORIO RIUS SOROLLA

Academic year: 2016-17

AKNOWLEDGEMENTS

First of all, I want to dedicate this work and all the hours I've spent with it to my parents, therefore I will continue in Spanish so that the can understand.

Quiero dedicarle este trabajo a mis padres, Manuel Luis Perez Peiró y Consuelo Balaguer Bosch, a quien les estaré siempre agradecido por la oportunidad que me han dado que si no fuese por ellos, nada de esto habría sido posible, ya que este trabajo significa más que lo que simplemente se ha escrito, es la culminación de una carrera académica de varios títulos universitarios y sobretodo de la increíble experiencia de haber vivido durante dos años en Italia.

<<Gracias a vosotros he llegado a ser Ingeniero Industrial, y espero que solo sea el principio del camino>>

También quiero dedicarle este trabajo a toda mi familia y amigos haciendo dos menciones más que relevantes.

A ti Ahinoa, porque desde que te conozco solo has hecho que apoyarme en todo y darme fuerzas para seguir. Y por ultimo, a mis abuelos por ser un ejemplo a seguir en la vida. Te lo dedico a ti yaya.

From now on I will continue in English. I also want to thank to the *Universidad Politécnica de Valencia* that will always be a second house for me and to the *Politecnico di Milano*.

Also thank my supervisors, Luca Fumagalli in Italy and Gregorio Rius Sorolla, that have made my work much more easier than I thought it could be, and also to all the people that in some moment have helped me during this process.

ABSTRACT

This monograph has the aim of presenting the necessary knowledge for the treatment and processing of data and its subsequent analysis for the recognition of patterns to be able to implement maintenance systems in companies beyond traditional maintenance systems, i.e. to implement E-maintenance systems. The context in which this platform is developed is the national project "Smart Manufacturing 2020" inside of which Politecnico di Milano has a role as academic partner and scientific coordinator, and thanks to which it has been possible to cooperate with the industrial partners Whirlpool Italia and Agomir S.p.A.

In the introduction it is presented the previous knowledge necessary for the complete understanding of this monographic work. Analyzing the importance of the correct implementation of intelligent maintenance systems, and also tools and methods are presented.
The work is developed using the mathematical software tool called MATLAB® through which the parameters and information necessary for the implementation of an E-maintenance system are analyzed.

In the present competitive world, work well done and a correct analysis of the available information can be an important differentiating factor for any company and organization

"Quae nos quotidie ut uirtus non sit actus, sed habitus."
Aristoteles.

Keywords: E-Maintenance, Big Data, Pattern Recognition

EXECUTIVE SUMMARY

Maintenance is defined as the combination of all technical activities, administrative and management, provided during the life cycle of an entity, intended to keep it or return it to a state where it can perform the required function. Traditionally, maintenance was regarded as a support function, not productive and not strategic, since it is difficult to attribute the added value that it made. This trend has, however, changed, especially in the last ten years, when it was observed that many industries have begun to use methods aimed at improving the effectiveness of maintenance, recognizing that the maintenance of technical equipment and machines (in general, physical assets) it is an essential part for the operation of industrial processes, and an effective maintenance strategy can significantly contribute by adding value to production activities[1]. In particular, the goal of many companies is to minimize the total cost of inspection and repair by collecting and interpreting data related to the operating conditions of critical components of an asset, by applying what is called Condition Based Maintenance (CBM).

Always during the last decade, information systems in companies are gaining a vital role not only at the level of management and planning but also at the field level involving the individual production steps and the individual machines. With the development of new trends and technologies (big data, e-commerce, the Internet of Things, etc.) software manufacturers to businesses are updating their products to allow for real-time communication with the plant and manage the vast amount of data from asset increasingly instrumented and "intelligent". In addition, strategies of intelligent maintenance are playing an increasingly widespread use even in areas less likely to digital innovation, attracted by the possibility of increasing profits by saving on maintenance costs and avoiding the loss of margins due to a production stop due to downtimes in machines.

In this context it talked about E-maintenance, seen as a major technological advancement that provides key support to innovate the maintenance management and in ensuring a better understanding of the production system, and E-maintenance.

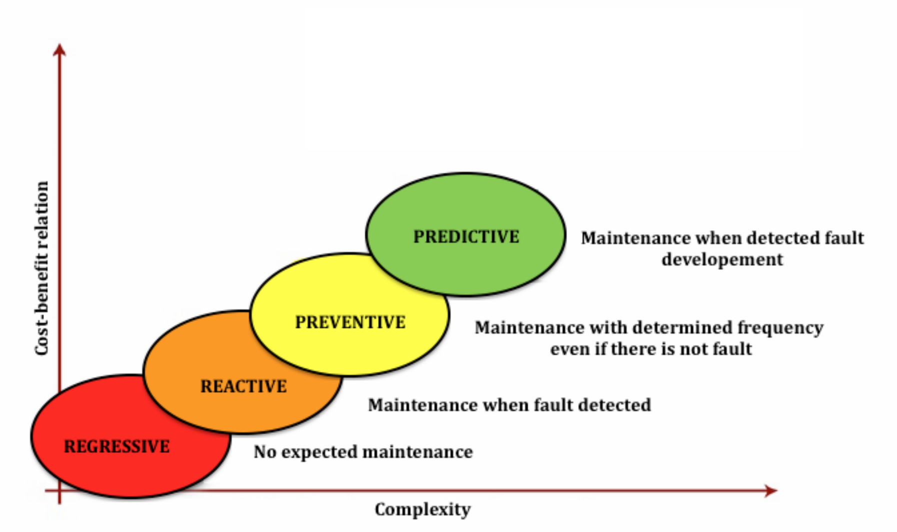


Figure ES.1

The objective of this work is to provide the necessary notions to the reader to be able to implement an E-maintenance system in addition to studying and analyzing the feasibility of this through multiple concepts of theory, and the development of techniques and methods to be able to implement it, as well as tools to serve as a means to apply the knowledge learned.

This work is developed under the idea that a series of available data must be initially processed and then analyzed for trend identification in order to be able to study the behavior of these data to predict when certain events will occur before they occur, or to be able to analyze the evolution of certain situations or behaviors of the system studied through these data. *Figure ES.1*.

With the information obtained through the analysis is intended to determine a pattern, which will be used as the basis for the development of the predictive maintenance tasks within the organization for a given system.

It will then try to analyze the feasibility of installing an E-maintenance or predictive maintenance system through a constant monitoring of the "constant vitals" of a system.

Therefore the particular objectives are established:

It is important to know techniques and methods for the treatment and cleaning of the data since these are usually of unknown nature, it is very probable that it may contain errors. This step is important for the development of an intelligent maintenance system because analyzing a system through an erroneous or falsified data system can lead to situations in which no information trends are detected or what would be worse from the point of view of company performance, to detect false patterns produced by these non-true data. A case of this kind could lead to a system totally unproductive functioning causing greater losses than if it was not implemented.

The next objective of this work is to develop techniques to proceed to the analysis of processed data. This is an iterative analysis of the data, searching for a pattern that is fulfilled in the majority of cases in which the event that wants to be predicted occurs.

It is not critical that a pattern does not meet exactly 100% of the cases (without taking into account critical cases in terms of safety, large economic volumes, or dangerous to the environment), since from the industrial point of view, in the long run, it is preferable to obtain fairly good predictions of behavior or evolution for applying predictive maintenance than applying preventive maintenance systems through statistical studies, which will have as a consequence the considerable waste of resources.

In relation to this topic are presented some useful tools for the study of trends and patterns, where Artificial Neural Network is highlighted.

Artificial neural networks are connectionist systems with a computational approach based on the large collection of neural units, where it is intended to create a system that can solve problems in the same way as the human brain. Each neural unit is connected to the others and consists of a system of self-learning and training. It stand out in situations where programming or determining a model is complicated, through this system they it is possible to determine models that may be very difficult to determine through the direct analysis.

This type of tools do not ensure the success when working with data, since it must be trained, therefore can be given complex situations in which would be needed Neural Networks mayors, i.e., the simulation of more neurons for implementation, in this research it will be worked with the utilization of 10 neurons, since higher values need very high computation times, in many cases to give similar results.

The last objective of this monograph is to use the results obtained through the analysis of the data to design a platform and an interface that informs the user about the conditions in which the system studied is, to make decisions with respect to maintenance of this system based on the real-time information.

Both for the development of the data processing, as well as for its subsequent analysis, and finally the design of the platform and the interface, it will be developed using the mathematical software called MATLAB®.

All items exposed had been applied to a particular work, the context in which the work has been developed is - as mentioned above - the national project "Smart Manufacturing 2020 'within which the Politecnico di Milano has the role of academic and scientific coordinator partner.

Asset object of monitoring of the developed modules is a machine that fills the coil of refrigerators with the fluid coolant, called FRIGUS. The working cycle is composed of several cycles, each of which has the pressure thresholds which, if not

observed, indicating that the cycle is failed and that the work piece is to be marked as defective. Defective parts are then sent to the repair department.
Through a set of historical data corresponding to the period 2014/2015 has proceeded to perform an investigation of the behavior of these data to try to determine some trends that helps predict a failure of the system. (See Annex I and II).

First of all, as it was established in the objectives, a detailed study of the data that has been done. Data is analyzed in a general way to detect errors in it, due to the fact that with large volumes of data (approx. 3 million datum in this case) it is impossible to analyze one by one. By studying the data, some types of data were determined that should be deleted in order not to contaminate the future results, these types of data corresponded for example, to those that as detected as not well introduced or impossible to be true. Those which had unusual values of, for example, temperatures above 10000 degrees Celsius, or pressures of millions of kilopascals. When analyzed more deeply, were detected trend that showed it was possible that some of the data could be false, (something common when working with large volumes of data). Detecting this type of fallacies is complicated, but vital if it is wanted to get a reliable and accurate model.

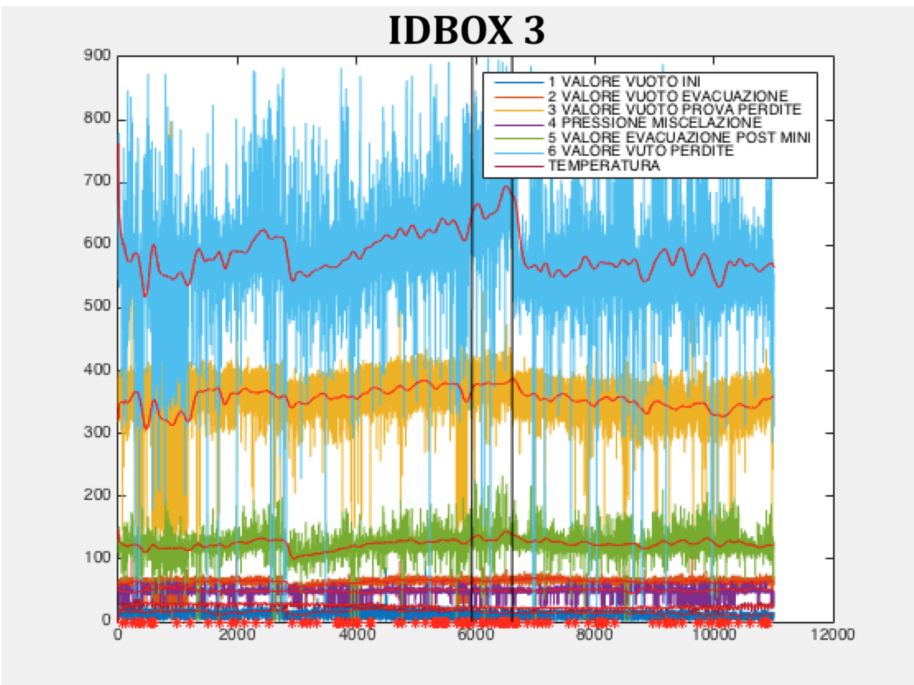


Figure ES.2

With the processed data, *Figure ES.2*, it proceeds to its analysis. In this monograph it is not possible to a comment over all the different analysis strategies that were attempted, as has been said previously this part is the most arduous part of an investigation of this type, it is an iterative process, many ways must be explored when searching for a pattern that can help determine the system's failure. The data were analyzed from different points of view until finally it was found a pattern that seemed to work in the majority of cases where there were situations in which it was necessary to apply maintenance labors. This pattern depended on a function called as "***Density of Failure***" *Figure ES.3*.

This function consists in understanding the provided detected failures of the system not as faults but as moments of malfunction, therefore with that vision, it is observed how the frequency with which these moments were detected was not constant, but was variable, and in some situations reached high amounts of detection in few moments. *(Figure ES.3)* Observing this behavior it was possible to determine that when the frequency of occurrence of these moments of malfunction exceeded certain levels, frequency of occurrence continued increasing without decreasing in any moment until after some maintenance work was carried out.

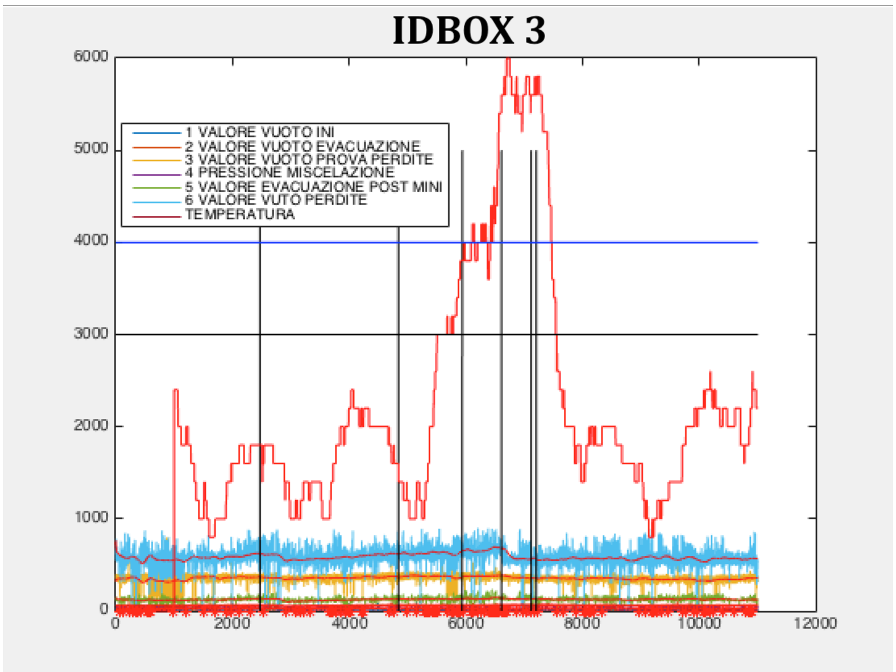


Figure ES.3

Subsequently, the validity of the idea that the system could be controlled by monitoring *Density of Failure*, *Figure ES.3,* was checked using the artificial neural network tool, verifying if there was any trend or pattern in the data obtained, where the results

Obtained were the same results comparing it to those obtained from the analysis of the system with the created function of *Density of Failure*.

Finally, a platform was developed to control in real time the evolution of the behavior of each analyzed machine. Based on the existence of a relationship between the developed function of *Density of failure* and the current situation of the machine, the platform after the introduction of the fault detection parameter at each control instant (it can be only fault detected or no-fault detected), will show through the interface the real conditions in which each machine is. The system can be in one of the three possible levels of behavior *(Figure ES.4)*; green level if the machine is working in normal conditions (low amount of failures detected in last moments), yellow level if the machine, still being in normal conditions, can derive into system's failure (medium amount of failures detected in the last moments), and the red level which refers to an imminent constant malfunctioning of the machine, away form the ideal situation, due to the high frequency by which faults are detected, (high amount of failures detected in the last moments).

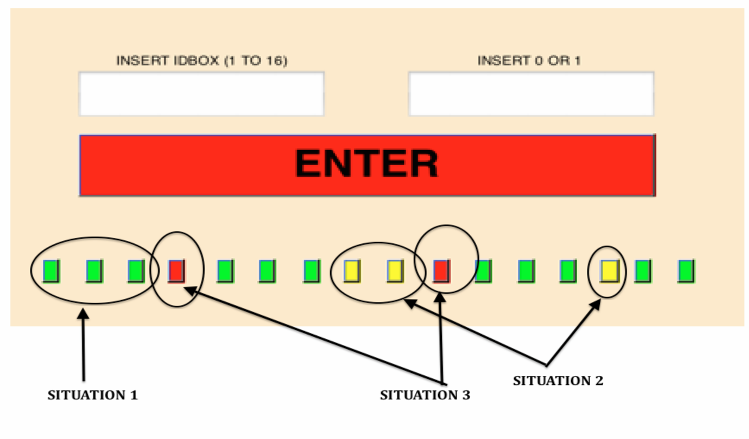


Figure ES.4

To achieve an increase of the quality in the processes of a company or organization is recommended the implementation of systems of intelligent maintenance. The implementation of these systems is not easy, since it entails the need to measure and take data of all possible items within the industry. Also these data must be processed and analyzed using advanced techniques in order to be able to find trends and patterns about behavior of machines and use these results to develop systems that help to prevent failures and permit to address the problem of failures before it occur reducing downtimes in the system, therefore also saving the resources in the company making it more efficient and sustainable.

INDEX

INDEX OF REPORT

INDEX OF ANNEX

INDEX OF FIGURES

REPORT

ANALYSIS OF DATA AND PATTERN RECOGNITION FOR THE IMPLEMENTATION OF AN E-MAINTENANCE SYSTEM

CHAPTER 1. INTRODUCTION

The present monograph tries to analyze some different ways of theoretical and practical evaluation and control of the parameters' evolution conditions of machinery that can be applied to any production process in industry.

It also tries to introduce and develop several aspects of maintenance that can make more than only costs saving in industry (which is considered as very important issue), produced by a better control of resources and decrease in uncertainty in normal working and in forecasts, it also gives added value to the final product or service trough the automation of some parts of the production process.

First of all it will be presented some important concepts needed for the complete understanding not only of this monograph but also of the present and much more of the future of industry and organizations. These concepts are:

- *QUALITY AND PROCESS MANAGEMENT IN ORGANIZATIONS*

  As the theoretical bases of this monograph is the idea of making a good maintenance of manufacturing processes, that at the end it will have a positive effect over the results, therefore over the client's satisfaction and the organization direction.

- *BIG DATA*

  Nowadays it is possible to get information of nearly every parameter inside and out of an organization, but this large amount of information and data is difficult to manage and to understand.

- *E-MAINTENANCE*

  The importance of innovation in maintenance strategies further than mere preventive and corrective methods will help industries to develop in an each time more competitive and globalized marketplace.

Further on, it would be described the application of this ideas to a real system, as said before, this being a machine that it fills a coil of refrigerators with the refrigerant fluid called FRIGUS, where a large amount of data is provided together with the original and real results of traditional corrective maintenance.

1.1. QUALITY AND PROCESS MANAGEMENT IN ORGANIZATIONS

1.1.1. DEFINITION AND DEVELOPMENT

Nowadays organizations compete in a globalized market what makes obligatory for industries to establish strategic plans based on the idea of continuous quality improvement of its products and services through continuously improving productive processes increasing productivity and efficiency[2].

Quality implies to always improve the efficiency and effectiveness of an organization and of its activities, and always being aware of clients needs. A client not only can be the final user of a service or product but also is the internal user of any output of an activity or process inside the organization.
If processes are planned, performance of the organization will increase.

Quality that is perceived by a client depends on the way the organization realizes activities that affect the clients, like recruitment, outsourcing, maintenance, service control, documentation, detection and correction of failures or incidents in time.

Client satisfaction is each time more difficult due to the fact of continuous dynamism of the market, so quality management is evolving in a way that each time are more important the factors that permit the better knowledge and fast adaptation to changing conditions in the market. Some of these factors can be strategic planning, or design of key processes of the business by the measurement and analysis of everything inside the organization and its continuous improvement[3] [4].

It have been identified eight Principles of Quality Management that can be used by the organization to drive it to an improvement in performance[5].

- Focus on the Client
  Organization depends on its clients and has to understand its actual and future needs to satisfy them.

- Leadership
  Leaders establish the unity in the proposals and the direction the organization will take.

- Personnel Compromise
  If all the personnel is compromised, its complete abilities can be used for the benefit of the organization.

- Focus on Processes
  A result that is wanted is accomplished when activities and resources are related and managed as a process

- Continuous improvement

The continuous improvement of organization's performance has to be a permanent objective.

- Focus on management

  Identifying, understanding and managing the processes, contributes to efficiency and effectiveness of a organization.

- Decisions taken by facts

  Analyzing data and information

- Partnership with suppliers

  A good relationship between suppliers and the organization helps to create value for the final product or service.

1.1.1.1. FOCUS ON PROCESSES

Any activity or set of activities linked together that uses resources and is controlled to transform input elements in results can be considered as a process. The results of the process have to have an added value with respect to the input *Figure 1.1.*

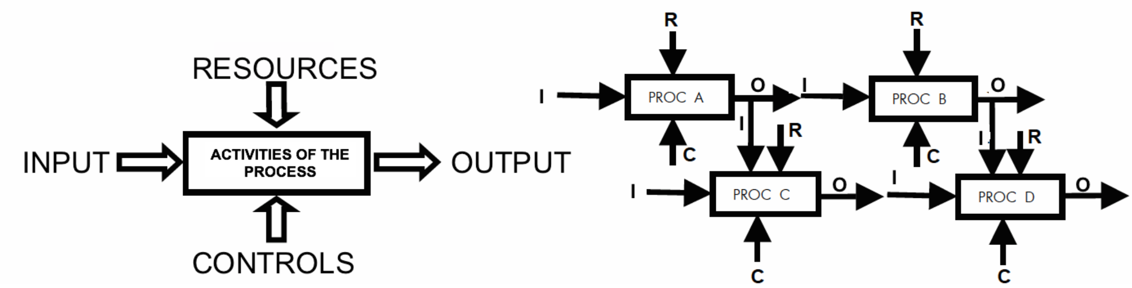


Figure 1.1. Definition of Processes

All the activities that occur in an organization have to be focused as a process and have to be centered in[2]:

- For every process, implement and accomplish the requests of the client.
- Plan the process in a way that gives value for the client
- Control, measure and obtain results of performance and effectiveness of the process.
- Continuous improvement based in facts (measurements).

It is very important to establish that the process management is not focused in error detection but in studying every process to correct the trend before a fail occurs producing a defective result.

So that a set of activities linked together lead to a certain result is necessary to define and control the process to which it belongs. The importance of directing and defining a process is that is not possible to act directly over the results, because the process drives to it, but to control the result it must be acted over the cause.

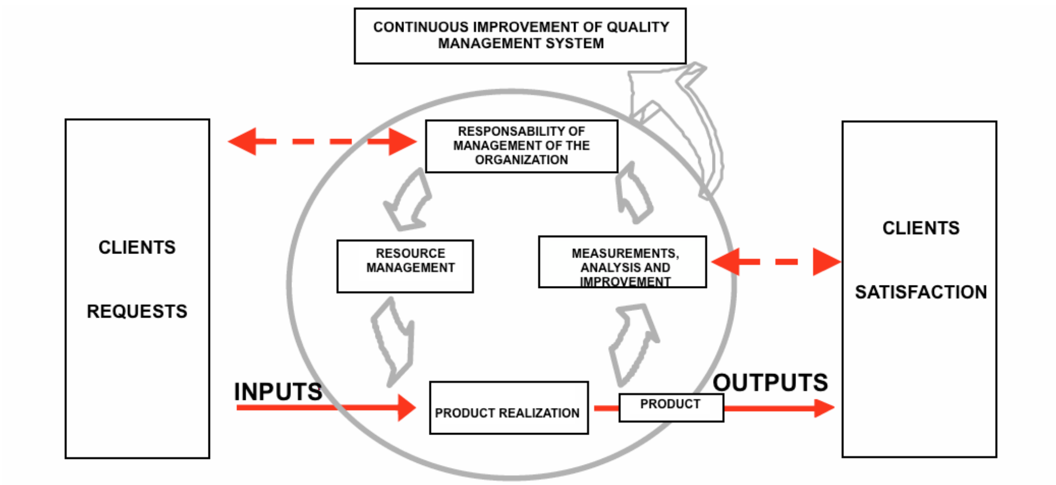


Figure 1.2. Quality Management Based in a Process

For example, imagining a process in which a car is assembled, if the final result is the car and it is defective, it is not convenient to act over the car to repair the defects, the idea proposed is to act over the process that produced the defective car as a result, so that this failure doesn't occur any more or at least to reduce the occurring probability.

The next figure, *Figure 1.2,* shows a model of a system of quality management based in a process where it can be highlighted the importance of measurements, analysis and improvement in a loop mode.

When, even though that activities are well done during the process, problems can occur, it is needed to apply the improvement cycle, this improvement cycle consists in making any action that can change the development of the process. At the end this improvement in the process produces benefits for the clients (internal or external), some of these benefits can be:

- Decreasing the amount of resources used, thinking as resources: materials, people, money, labor… having as a result an increase in efficiency.
- Time needed is decreased, increasing productivity.
- Errors are reduced, helping it to be prevented.
- It provides a systematic view of the activities of the organization.

The improvement in the process can be made in two ways, in continuous way, or by reengineering. In a continuous way it optimizes the existing process reducing errors and defects, while by reengineering it has as objective to make an important change in the process without respecting the existing parameters.

1.1.1.2. CONTINUOUS IMPROVEMENT

The process of continuous improvement, *Figure 1.3.* is a concept originated in the middle of the 20th century that as described before pretends to introduce improvements in products, services and processes.

Some of the most important elements that are used to achieve the continuous improvement are the corrective, preventive and predictive acions[3].

Figure 1.3. PDCA Diagram

There exist some techniques for data analysis that can be useful in a continuous improvement process. Usually most common problem is variability, and one of the proposals of the organizations should be to control it.

Statistical techniques as graphs, correlations analysis, etc., can help measure, analyze, describe and interpret the variability with a limited quantity of data.

The statistical analysis of this data can help to solve and prevent problems resulting form variability.

As it has been seen, the continuous improvement is not something of one day, but it is a progressive process. So the continuous improvement implies the implantation of a system of "learning" and control.

To support this process it exists some tools, as it can be the Pareto Charts, also known as the 20-80 Rule, where basically the idea is that a few causes, produce most of the problems[4].

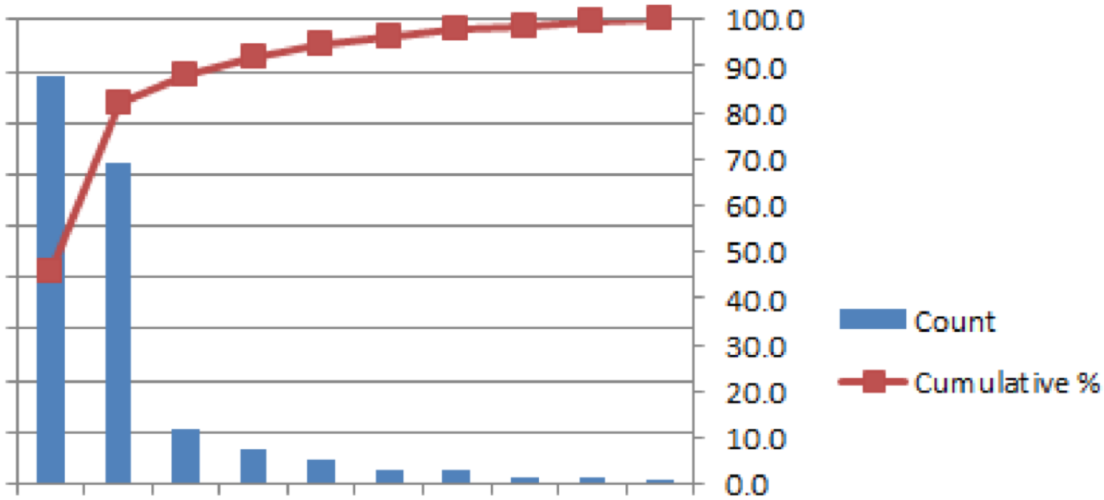


Figure 1.4. Pareto

Its origin it's due to the studies made by *Wilfred Pareto* at the beginning of the 20th century *Figure 1.4.*

This type of analysis is a way of identifying and differentiating the few "vital" of the high amount of "important", or in other words, give a priority to some aspects that are present with more frequency, or that have a higher weight in the result.

It is used to establish where have to be focused the study to determine the causes of a problem.

One of the most famous methods used to improve processes is six-sigma method. The idea is to reduce the variability by reducing to the minimum as possible the errors that are produced in a process.

This method has 5 steps, *Figure 1.5,* which are Define, Measure, Analyze, Improve and Control. This method tries to reduce the number of defects until levels of 3,4 over a million[6].

Figure 1.5. DMAIC – Six-Sigma Analysis

1.1.2. ADVANTAGES

The advantages derivated of introducing methods of continues improvements in an organization are multiple.

First of all, this type of management exposed before gives stability to the organization, what has as a consequence the possibility of making a better prevision and planning of resources, being this, material resources and also human (labor) resources.

In situations in which is possible to predict when a fail can occur in real time, it is possible to have an action plan prepared for the moment in which the failure is going to occur, having available the materials needed for the reparation as such as specialized human resources that can make the reparation.

Also with the reiterative study and analysis of the systems it can occur that it is arrived to a situation where the probability of having a failure is very low. Due to all of these, important decreases in production cost can be reached, even being able of decreasing final`s product or service price having as a result an increase of the competitiveness of the organization. Also it can be given the situation in which is produced an increase of the productivity due to the fact of having working machines/processes with lower amount of failures and problems.

Enabling prevention and continuous improvement is also achieved the adaptation of the organization to the dynamism of the market, making it easier to install new methods for productive processes.

In particular, the use of improvement techniques and analysis techniques such as Pareto Charts allows the non-waste of energy when analyzing the system, giving value to those causes of failure that are more important, because it prioritizes those few causes that have much effect, or a more frequent effect in the failure of the result.

1.2. BIG DATA

1.2.1. DEFINITION AND DEVELOPMENT

Due to the fact of the great advance made in the last years with respect to technology of information, the organizations have had to face new challenges that permits them to analyze, discover and understand further than before.

Big data analysis is the analysis of large volumes of data that before was impossible to analyze. More than its uses to propose solutions for strategies of the organization and marketing, big data also inspires new ways of transforming processes inside the organizations.

It is possible to see four dimensions inside big data, this dimensions are Volume, Variety, Velocity and Veracity[7], *Figure 1.6*.

- Volume
  Is the most recognized characteristic about the big data, the great amount of datum that the organization try to recollect to improve their decision making, therefore the results[7].

- Variety
  Different type and sources of data. Organizations have to analyze lots of different types of data gotten form different types of information. With the abundance of sensors and intelligent devices, the data generated is present in lots of different ways.

- Velocity
  Big data has not only to manage large volumes of different types of data, but also it has to manage the high speeded way this data is produced.

- Veracity
  The uncertainty of data is important at the level of reliability of the information obtained. It is very important to get data of high quality that is why it is very important to establish a good system of data cleaning and treatment.

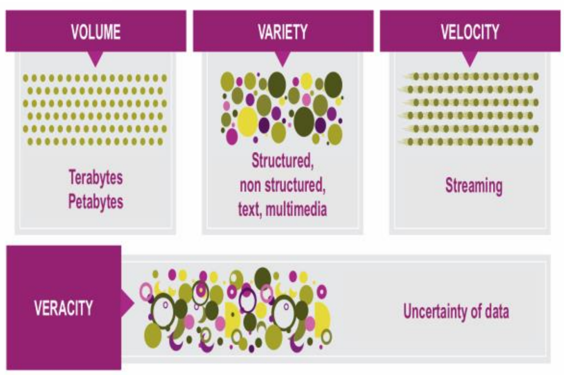


Figure 1.6. Dimensions of Big Data

Further than focusing in which data has to be analyzed, big data focuses in "what problem has to be solved" [7].

It is possible to found very different types of data. *Figure 1.7.*

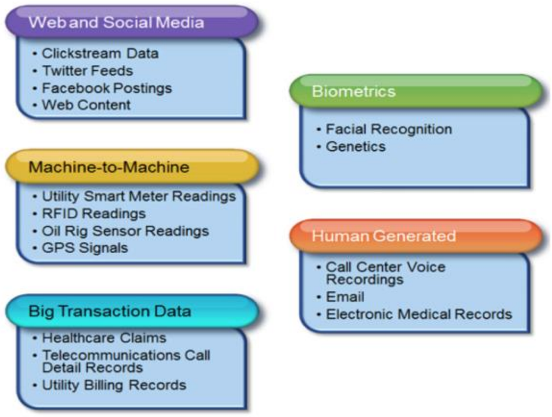


Figure 1.7. Types of Data

Big data is a topic that could well give to write a monograph only about this issue, but here it is focused in the idea of big data analysis for processes' optimization inside an organization.

Most of the efforts done by organizations with respect to big data are getting and analyzing internal information. Most of this internal information proceeds of data generated by machines and sensors that are used to register details of automated functions, due to the fact that this amount of information generated is really large, with traditional ways of analysis and storage it was really difficult to study it, therefore most of it weren't analyzed[8].

This type of information is called Machine-To-Machine (M2M) and it is referred to technologies that permit the connection with other devices. M2M uses devices such as sensors or measurement devices that get information of events as speed, temperature, pressure, etc., that are transmitted through wire or wireless network to other applications that translate these events in meaningful information.

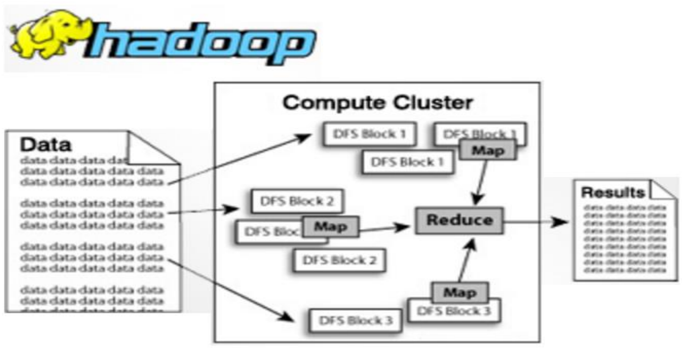


Figure 1.8. Hadoop®

As a small parenthesis it is relevant to present *HADOOP®, Figure 1.8.* the most popular platform used to analyze this large amounts of data.

Basically consists in two different processes, mapper and reducer, where data first is "mapped" which means is grouped in blocks, and then it is "reduced", which means that the amount of data that is redundant to already existing data is deleted. In this way it is possible to analyze large amounts of information in a faster way.

Collecting, storing and analyzing data from industrial sensors and devices connected to the so-called "*Internet of The Things*" [9] has become more feasible because of the emergence of big data technologies, and every time more organizations are joining to this techniques and it would increase even more in the future, looking to take advantage of the data and information available.

1.2.2. ADVANTAGES

The nature of the information is different nowadays, due to the abundance of sensors, images, records, etc., in organizations; the data produced by these documents has reached volumes of very difficult treatment and control.

The most important advantage of big data is that it has permitted organizations to discover patterns in processes that it would be impossible to find in other circumstances due to the large amount of work that should be needed.

To compete in an integrated global scale it is necessary for organizations of today to highly understand the market, clients, employees and internal processes. This understanding requires a use of information and data analysis available.
Big data is a way of optimizing any decision inside the organization, having as result a more efficient system.

Even though not every sector or organization is forced to use the big data analytics, it is recommendable due to the fact that it permits them to understand their operations, therefore it is possible to prevent events in the organization as it could be failures in machines: And as a result of all of this, it will differentiate the organization giving it a high competitive advantage.

1.3. E-MAINTENANCE

1.3.1. DEFINITION AND DEVELOPMENT

Nowadays with a global competitive situation, exists the need for manufacturing industries to reduce and delete non-programmed inactivity time due to its high costs.

With Internet and the rise of new technologies (*big data*), organizations need important changes in traditional practices that affect to failure maintenance from a corrective and prevention way to a prediction way.

Even though the concept of E-maintenance emerged in the year 2000, is not yet clearly defined, and it is possible to find different types of points of view from which it is defined the idea.

For example Professor Benoit Iung, of the University of Nancy, and Adolfo Crespo-Marquez of the University of Seville, in their book "*PRODUCTION, PLANNING AND CONTROL"* define this concept as [10] *Figure 1.9.*:

E-Maintenance =
Excellent Maintenance =

Efficient Maintenance
(do more with few people and fewer money) +

Effective Maintenance +

Enterprise Maintenance
(Contribute directly to enterprise, to the organization)

Figure 1.9. Definition of E-Maintenance

While authors like Kenneth Holmberg and Adam Adgar in their book "E-Maintenance" published by SPRINGER ®. In 2010, write:

"E-maintenance is a synthesis of two large trends in our society: on one hand the growing importance of maintenance as a key technology to keep machines running properly, efficiently and safely in industry and transportation and on the other hand, the very rapid development of information and communication technology (ICT). This has opened the way to completely new concepts and solutions with more detailed equipment for health information and more effective diagnostic and prognostic tools and users interfaces to ensure good reliability and availability of plants and vehicles remotely worldwide" [11].

E-Maintenance approaches the needs of predictive intelligence tools to supervise and control the changes more than detecting failures, so it optimizes resources.

For users, usually machines seem as if it fail impromptu, but really, machines usually go through a process of degradation that is measurable and normally invisible. But, most of the sensors and measurement devices used in organizations for parameter measurements are capable of proposing information about the situation and performance of the machines or system.

When intelligent systems are connected to the common network and are supervised and when its information and data is analyzed continuously with specific systems it is possible to go further than basic “preventive maintenance” to “maintenance by prognostics.” [12] These are systematic issues than can track the deterioration of the machine or system to predict a failure. This continuous knowledge of machine’s situation permits the progress to e-maintenance where maintenance actions are synchronized with operation of the system and also with the needed resources both of materials and labor. This situation should lead to a near zero wasted time of production.

Traditionally failure prevention is done based on data that describes the timing between failures without taking in account the health of the service, machine, or process at each moment.

The principal of degradation analysis is to investigate its physical characteristics, it evolution, the evolution of the performance and functioning that drives the system until its failure.

It is important to present a maintenance issue called “*Condition Based Maintenance (CBM)*, *Figure 1.10.* that it develops considering actual degradation and how it changes with time.

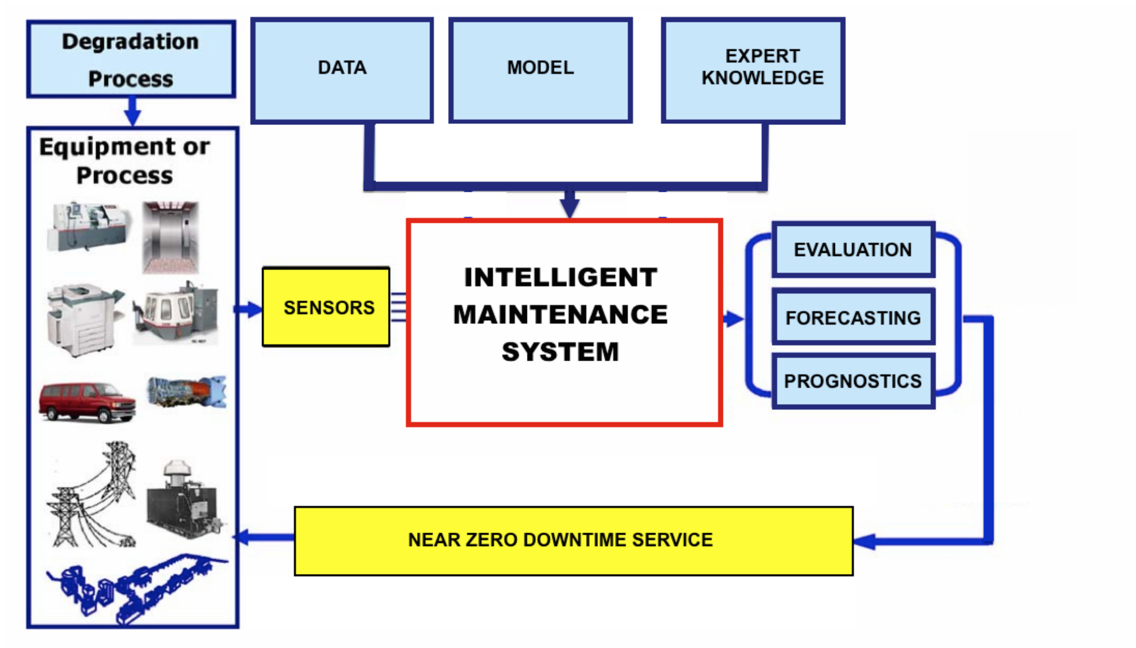


Figure 1.10. Intelligent Maintenance System

The principal idea of CBM is to use the information generated by measured parameters of the degradation of the process or machinery, to minimize time a process is inactive, minimizing failure risk and minimizing benefits. The decision making of CBM is based in prediction maintenance. For this many tools have been developed.

It exists a large variety of prognostic applications for example it can be the development of the so-called *Neural Network*, what will be discussed further on.

Actual prognostic issues can be classified in three basic groups, *prognostics based in model*, *prognostics based in data,* and *hybrid prognostics of models and data.* The difference between both is that prognostics based in models include data recollected of simulations made with models of normal and deterioration conditions. The models are designed over the base of different load conditions.

In the other hand, sometimes is very difficult to get a model or at east a feasible and precise model of the system, so another way to try to determine when failure will occur is to monitor de evolution of a possible failure that is developing and try to predict how many time it will pass until this failure evaluation reaches a determined level of deterioration and then some action is needed. This method is called prognostics based in data.

All the information of the prognostics obtained from the process or system it also can be used to improve process operations and maintenance with the idea of increasing feasibility of the process and increasing final's product quality.

A very useful smart algorithm for prognostics for intelligent maintenance is called **WATCHDOG AGENT ®.** *Figure 1.11.*

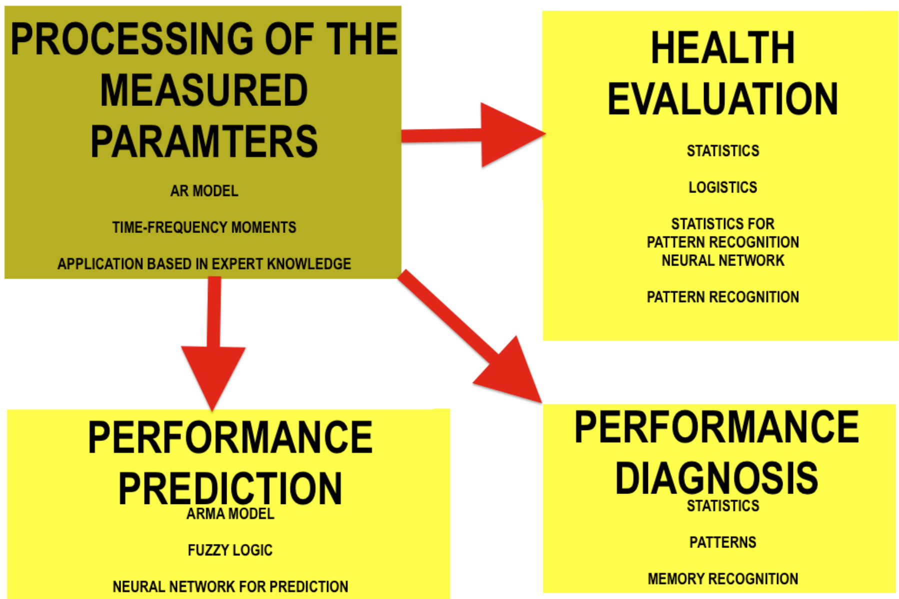


Figure 1.11 Watchdog Agent ®

WATCHDOG AGENT ® bases its evaluation of degradation in reading multiple sensors that measure the parameters of the process or machines that are wanted to be analyzed and studied. This application's prognostic function is realized through typical trends and/or statistical models of the performance parameters. With this it can be decided the future behavior of the patterns and with that it is possible to predict the behavior of the process or system that is analyzed.

WATCHDOG AGENT ® also has de ability to memorize repeated patterns in order to recognize what has been already observed in the past, and therefore also can recognize situations that have never occurred before[7]. So this system has the ability not only to answer the question of *WHEN* is going to occur the failure, but also to diagnostic *WHAT* is causing this failure.

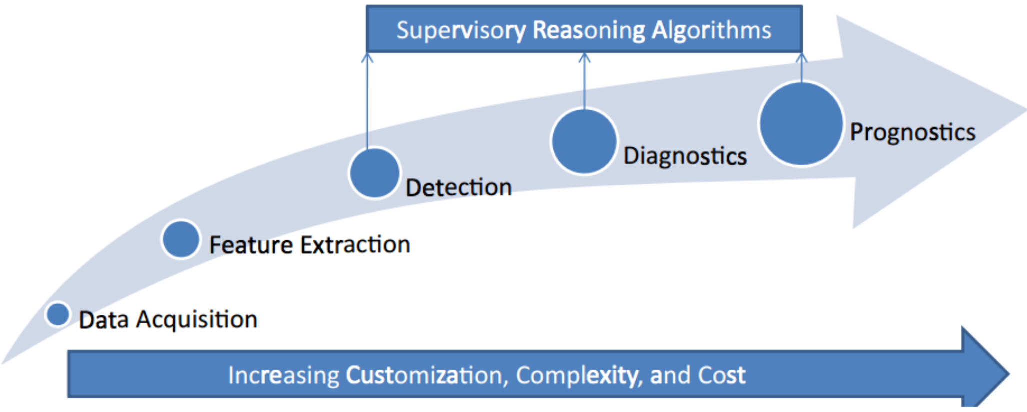


Figure 1.12. Maintenance Evolution

Turning back to CBM, *Figure 1.12.* it is possible to see the great difference between the programmed maintenance also called preventive maintenance, where maintenance is done every fixed interval of time while with CBM strategy maintenance

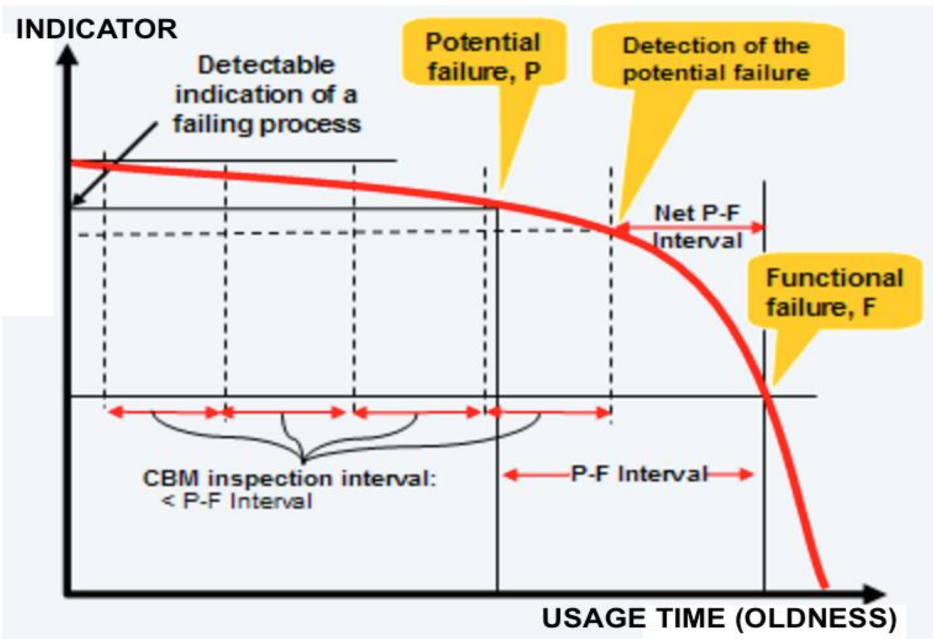


Figure 1.13. Condition Based Maintenance (CBM)

is only done when it has been seen deterioration or a negative evolution of the performance and it has reached a determined threshold established as a limit of correctness. *Figure 1.13.*

If it is compared with preventive maintenance it can be seen that less repairs are done and less failures occur due to the fact that maintenance is done when it is necessary.

A very typical example of CBM is the maintenance of the oil of a car. In this situation, usually manufacturer of the car established the intervals of time for which the oil has to be changed. These intervals are decided because of studies done before, estimations and averages (this will correspond to preventive maintenance). With CBM strategy the idea is to install sensors that get information in real time about the conditions of the oil, so the oil isn't needed to be changed until it is really detected that it is starting to be in not so good conditions as before. The benefit of this is that it doesn't occur the situation where the product is wasted because of a change before it is really needed.

CBM strategy needs fast data acquisition, treatment and storage due to this it is easier to sea implemented this strategy in critical systems more than in simple systems.

There exist some techniques applied to preventive maintenance, for example:

- Tree Failure Analysis
- FMECA Analysis

Tree Failure Analysis is a deductive technique that focuses on a particular event (accident) and provides a method to determine the causes that have produced such an accident [13]. It was born in the 60's for verification of design reliability of the

Minuteman rocket and has been widely used in the nuclear and chemical field. The fact of its great utilization is based on that it can provide both qualitative results by the search of critical paths, as well as quantitative, in terms of probability of component failures.

For the treatment of the problem a graphic model is used that shows the different combinations of faults of components and / or human errors whose simultaneous occurrence is sufficient to lead to an accidental event.

The technique consists of a deductive process based on the laws of boolean algebra, which allows determining the expression of complex events studied in function of the basic faults of the elements that intervene in it.

It consists of systematically decomposing a complex into intermediate events until it reaches basic events, usually linked to component faults, human errors, operational errors, etc. This process is done by linking these events by logical gates.

Each of these aspects is represented graphically during the elaboration of the tree by different symbols representing the types of events, the logical gates and the subsequent transfers or developments of the tree.

An example of a fault tree is as follows [14]: *Figure 1.14.*

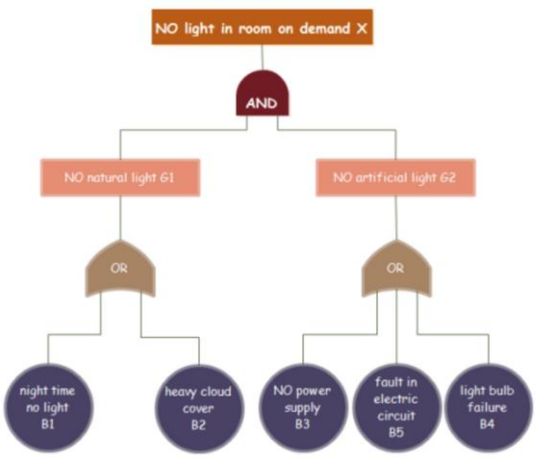


Figure 1.14. Fault Tree

Another useful technique is the analysis of failure modes and critical effects (FMECA).
The intent is to identify the areas or assemblies that are most likely to lead to failure [13]. The FMECA defines the function as the task that a component performs, for example, the function of a valve is to open and close, and failure modes are the ways in which the component can fail. The valve will fail at the opening if its spring is broken, but may also trip on its guide or be held in the open position by the cam due to a break in the camshaft strap.

The technique consists in evaluating three aspects of the system and its operation:

- Advance conditions of operation, and most likely failure.
- Effect of failure on performance.
- Severity of failure in the mechanism.

The probability of failures is usually evaluated on a scale of 1 to 10, or in an alphabetical order, with criticality increasing with the value of the number, or letter. This technique is useful for evaluating alternative solutions to a problem but is not easy to use with precision.

The FMECA *Figure 1.15.* helps in identifying the failure modes that are likely to cause product use problems. It also helps to eliminate excessive weaknesses or complications, and to identify components that may fail most likely.
It can also be used effectively to assess the causes of downtime on production machines.

COMPONENT	FAILURE MODE	FAILURE CAUSE	FAILURE EFFECTS	SEVERITY	PROBABILITY	REMARKS
POWER SUPPLY (ENGINE)	No electricity	Voltage drop	No motion can be produced in engine, no electric power	III	B	Periodical inspections
ENGINE	Excess current	Overload	Impossible to open/close the slipping doors	IV	B	Load cell controller
	Burnt	Excess time of usage		II	C	Preventive maintenance
	No start	Control error	No torque generation	IV	B	Periodical inspection
		De-conexion				
INFRARED	Deterioration	No signal transmission	No detection of entity between slipping doors	III	C	Periodical inspection
		Distorted signal transmission	Lateness in open/close motion	IV	A	High quality sensors

Figure 1.15. FMECA

(This FMECA is an example made by Manuel Perez Balaguer in the year 2016 for the possible failures of an automatic door, for the class of Security in Automation)

It is important to consider that the productivity of an industry will increase as machine or process failures decrease in a sustainable way over time. In order to achieve this, it is essential to have the most appropriate maintenance strategy and with trained personnel both in the use of fault analysis and diagnostic techniques implemented and also with sufficient knowledge about the design and operation characteristics of the machines.

In the last pages it has been exposed from strategies to techniques for the application of e-maintenance through the presentation of platforms and applications to be able to achieve an outstanding decrease of errors and failures in the processes of the company getting with it to optimize to the maximum the resources of the company since it is possible to predict and prognostic events and their dimensions reliably, highlighting the savings in non-productive downtime of machinery[15].

1.3.2. ADVANTAGES

Often corrective and preventive maintenance based on intervals of time does not succeed to increase life cycle of components or machines. Most of the time, this components or machines are replaced when it still have remaining useful hours, having as a consequence an unnecessary waste of both parts and machinery as well as time and labor resources.

CBM, as preventive maintenance offers an alternative way of dealing with failure in process, which as seen before it monitories some parameters of the process to control and analyze the situation of it in real time, so it can determine when is the real moment in which this process or component has gone over a marked threshold that determines that it has started to deteriorate.

E-maintenance should be considered by organizations as an important strategy for optimization and control of the systems and processes where advantages described below can be outstand:

- Failures are detected in their initial stages so there is enough time to plan and schedule corrective under controlled conditions that minimize downtime in the system and the negative effect on the production guaranteeing a better quality.

- Detection techniques are "ON-condition" techniques, which means that inspections can be performed with the machinery in operation, with sensors that measure and monitories as much parameters as possible.

- Is a proactive maintenance because it allows you to anticipate faults before they occur in operation and not afterwards, as corrective maintenance does.

Today, these strategies are only used for critical process due to the fact that even though it has as a consequence savings in costs of production, it is still quit expensive to implement.

CHAPTER 2. OBJECTIVES

With the present work, it is wanted to set a criterion for the establishment of the ideas and concepts exposed in the previous point.

The general objective is to help in the creation of a system that establishes as maintenance strategy the E-Maintenance concept starting from the point of having available a relatively large amount of semi-processed data obtained from the measurements made by the installed sensors and by the personnel in the workshop.

First of all to establish the specific objectives of this work it is necessary to answer some questions:

- It is necessary to know where does the data come from and what does it mean, its nature, to easily understand and know how to treat each type of datum. If this datum has to be processed with which criteria do we need to process it?

- Once having available all the processed data, it is necessary to know if some pattern can be detected, but also it is necessary to know what methods and techniques can we use to determine this pattern if it exists, and the tools that it will be used to do this.

- In the case some pattern has been identified, determine if it will be necessary or helpful to design any type of system that permits the control and understanding of the situation, of the conditions of the machine o process, in a feasible and precise way, the most approximated to real time situation.

Therefore, to answer these questions exposed, three specific objectives are established for the correct development of the study.

- Identify and processing of data.
- Study data behavior and determine possible patterns to prevent failures.
- Design of a platform and of an interface to monitor conditions in real time.

2.1. IDENTIFICATION AND PROCESSING OF DATA

When working with data generated in a system, usually it is imprecise and in most of the cases it is difficult to understand.
Therefore to obtain a good final result it would be very useful if data that has to be managed is understand, thus it will make more efficient the work done and surely more precise[16,17].

Turning back for a moment for some information exposed before, it was seen that usually in present organizations it is needed to work with very large volumes of data that are generated at high rhythms. This is the idea of working with big data, so first of all it will be described how data is categorized with big data systems.

Two of the categorizations most used in Big Data are usually those that relate the structure of the data and those that depend on the origin of the same[16,1718,19].

Types of data by structure of the datum, which usually are organized in two big categories:

- Structured:

  Created: data generated by systems in a predefined way (records).
  Triggers: data created indirectly from a previous action (ratings).
  Compiled: summaries of data in an organization, registers
  Experimental: data generated as part of tests or simulations.

- Unstructured:

  Captured: data created from behavior of the users (tracking position).
  Generated by users: data that a user specifies (in our case of industry could be the data generated by a production worker).

Types of data by its source [17]:

- Web Networks (data of the system).

- Communication between machines (sensors).

- Generated by people (operators).

From the point of view of the author of this monograph, more than classifying the data received by the machines in this ways exposed, it will be treated from know on with another criterion.

Data can be of different types, and origins, but most of the data that is generated is some type of numerical or alphanumerical value, which usually gives us values of measurable quantities. This type of data represents the evolution of conditions of some determined indicators.

The other type of data useful for this work is data that does not represent directly a value over the system, but it transmits information of how are the process characteristics, it can be seen as a code, one example of this type of data described could be the identification number of an operator in an organization, also the code of a process or a machine.

Once studied the data and after understanding what does each thing mean, it is important to clean this information, so only the useful data is used.
Data cleaning consists of detecting irrelevant data, or data that has errors, and delete it.

One problem that it can be found is the missing of data, to fill this value, it can be taken various strategies, and some of these can be to introduce the value of the average or make an interpolation with the surrounding results of data.

Other problem can occur, with wrongly manipulated data, this is difficult to detect, but if detected this data must be deleted because it is erroneous data, and working with this type of erroneous data can lead to erroneous results. An example of this data can be the detection that one machine gives values totally different to the rest of machines that are exactly the same, and are working in exactly the same conditions, this data could be treated as false, because it can be a problem of the sensors that are installed in that exact machine, another typical example could be the wrong data collection of an operator, in this case it is very important to determine which operator is making the wrong data collection and delete also the data corresponding to it.

Another problem could be the noise, this problem is also difficult to detect, but starting from the base that sensors are highly influenced by noise in the system, it would be advisable to filter the data looking to reduce the effect of the noise in the results.

A technique to filter the noise could be applying the called:

- *EXPONENTIALLY WEIGHTED MOVING AVERAGE (EWMA) Figure 2.1.*
  This method is a variation of the simplest method of MOVING AVERAGE witch consists in making and average over specific windows of the whole data provided and working with this value obtained for every instant.
  Exponentially weighted moving average instead of equally weighting all data, it puts more weight on the most recent data by a mean of an exponential weighting factor, this makes that if a change in a mean occurs it can be detected faster.
  It is important to remark that excessive filtering can hide the faulty conditions[20,21].

Figure 2.1. MA vs. EWMA

2.2. IDENTIFICATION OF PATTERNS IN DATA

The general idea of this section is to explain how it is possible to find patterns inside the data, which can lead to find the conditions that end up producing failure in processes and systems.

Two different methods or strategies are presented, being these Process Monitoring method, and the Neural Network method, emphasizing in this last method, due to the fact that even though is not a new idea, yes it has been started to be implemented for this situations in the last years due to its great complexity.

To find useful patterns in data it can be used independently both of the strategies by their own, or also it can be used both together complementing each other.

In process monitoring what it is tried to be found is some parameter that analyzing its evolution during the process it can be helpful to determine when the process will fail, while the Neural Network strategy it permits finding some patterns that will determine if the input data provided is in one situation or another. This means, Neural Networks doesn't give the clue that will take in a future to a determined situation, but it gives the exact clue to know in what situation it is found at a certain moment.

The ideas of these strategies will be described in the next sections and also analyzing the case of this work further on.

2.2.1. PROCESS MONITORING

Process monitoring consists in the idea of monitoring a parameter of the system, for example pressure, temperature, color, etc., that changes during the process. This parameter (or parameters) is wanted to be monitored to identify any change that always occurs or occurs most of the times before a failure (or any event) occurs.

This means that it can be studied and analyzed a parameter to describe its evolution from a correct working state until the failure mode, to detect the moment its behavior changes.

Is the basis of the before explained*, Condition-Based- Maintenance*, i.e., preventive maintenance. Which will permit maintenance to be scheduled to prevent the system from failing and therefore from all the consequences that this has.

For this strategy, it is also used the concepts exposed in the section 2.1 about techniques for data filtering.

For using this method, control charts, *Figure 2.2,* are used to show and detect unusual variability of the process or system. This control charts are monitoring tables against time (discrete or continuous). In this charts also are characterized some thresholds marking where the limits of a condition are, this helps to detect strange results that can be important for preventing events.

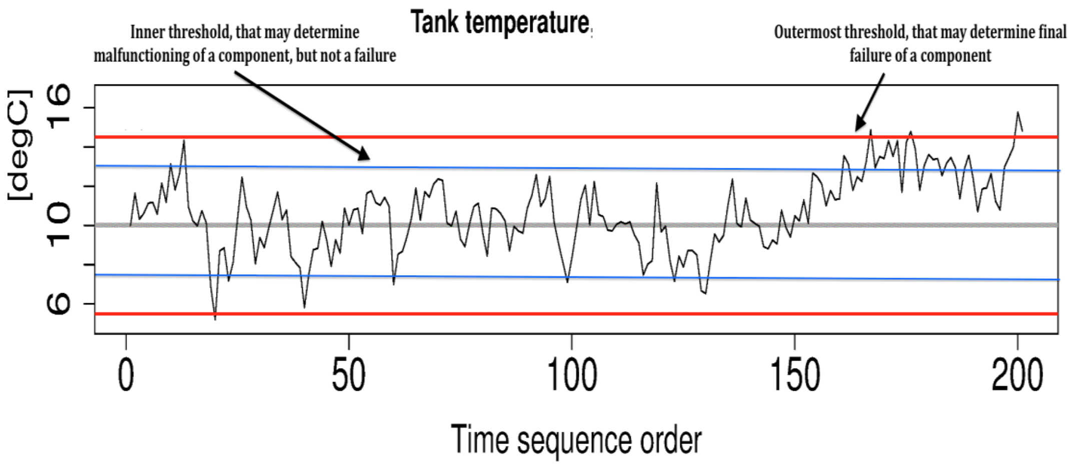


Figure 2.2. Process Monitoring

Control charts are developed in two phases:

First of all, the chart is constructed with historical data that has been recollected through past experiences, which is done offline in an iterative way. The idea is to discover parameters that are usually stable, and that get destabilized when the desired event of study happens [20].

The second phase to implement this method in E-maintenance will be to use the information obtained for real time processes, this means to monitor the process' parameters at the same (or similar) rate that are generated.

Any information or data can be monitored, and its basic purpose is to react as soon as possible to strange conditions that can lead to failure events. So, ideally, when the information is generated, it should be processed. Not only variables of the production system should be processed for E-maintenance, but also variable of resources needed for possible reparation in malfunctioning components.

2.2.2. NEURAL NETWORK

In the last decade Artificial Neural Networks have received a particular interest as a technology for data processing, since it offers the means to model effective and efficient large and complex problems[22]. Neural Network models are constructed from the data, that is, it is able to find relations (patterns) in an inductive way through learning algorithms based on existing data rather than requiring the help of a model to specify the functional form and its interactions[23].

Neural Network is a method of solving problems, individually or in combination with others methods, for those tasks of classification,[22] identification, diagnosis, optimization or prediction in which there may be a need for learning at runtime and for certain fault tolerance. In these cases the RNAs adapt dynamically by readjusting constantly the "weights" *Figure 2.3* of their interconnections[22].

Neural Networks are based on the analogy that exists in the way it behaviors the human brain. In particular the nervous system, which is composed of networks of biological neurons that have low processing capacities, yet all their cognitive ability is sustained in the connectivity of these between them.

The unit of an artificial neural network is an elemental processor called neuron that has the ability to calculate, in general, a weighted sum of its activation function to obtain a signal that will be transmitted to the next neuron.

These artificial neurons are grouped in layers or levels *Figure 2.4,* and have a high degree of connectivity between them, connectivity that is pondered by weights. Through a learning algorithm the Neural Networks adjust their architecture and parameters so they can minimize any error function that indicates the degree of fit to the data[24].

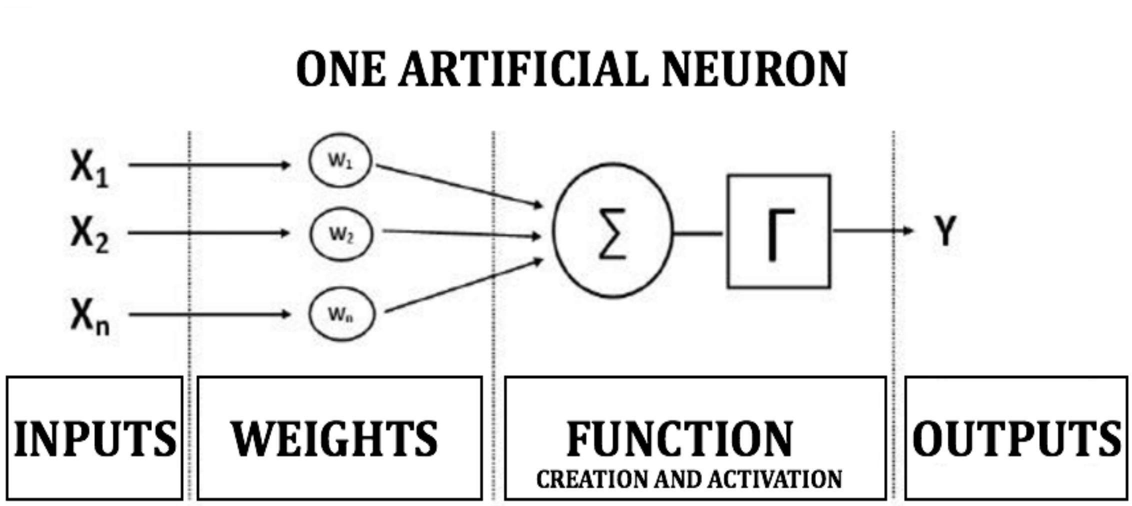


Figure 2.3. Neuron

An Artificial Neural Network, as discussed above, is a distributed computing scheme inspired by the structure of the nervous system of humans. The architecture of a neural network was formed connecting multiple elementary processors, this being an adaptive system *Figure 2.5,* that has an algorithm to adjust its weights (free parameters) to reach the performance requirements of the problem based on representative samples, it is important to note this idea already which is the main basis for such a system. Therefore we can point out that an Artificial Neural Network is a distributed computing system characterized by [25]:

- A set of elementary units, each of which has low processing capacity.
- A dense interconnected structure using weighted bonds.
- Free parameters that must be adjusted to meet the requirements of performance.
- A high degree of parallelism.

It is important to note that the most important property of artificial neural networks is their ability to learn from a set of training patterns, i.e. is able to find a "model" that fits the data[26].

The learning process also known as *network training* can be supervised or unsupervised[27].

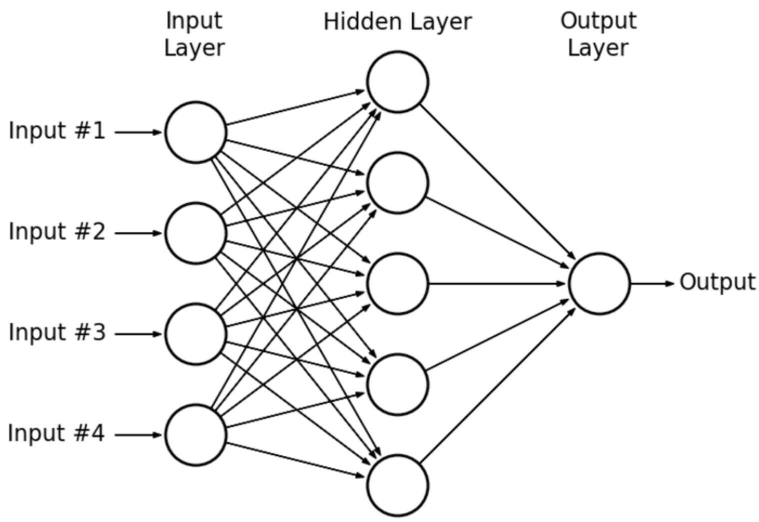


Figure 2.4. Neural Network

- Supervised learning consists of training the network from a data set or training patterns composed of input and output patterns. The objective of the learning algorithm is to adjust the weights of the network such that the output generated by the Artificial Neural Network is as close as possible to the true output given a certain input. That is, the neural network tries to find a "model" for the unknown processes that generated the output. This learning is called supervised because the output pattern is known which plays the role of supervisor of the network. This output pattern that is known will be called as *target values* further on[27].

- Unsupervised learning, only one set of patterns is presented to the Artificial Neural Network, and the aim of the learning algorithm is to adjust the weights of the network in such a way that the network finds some structure or configuration present in the data.

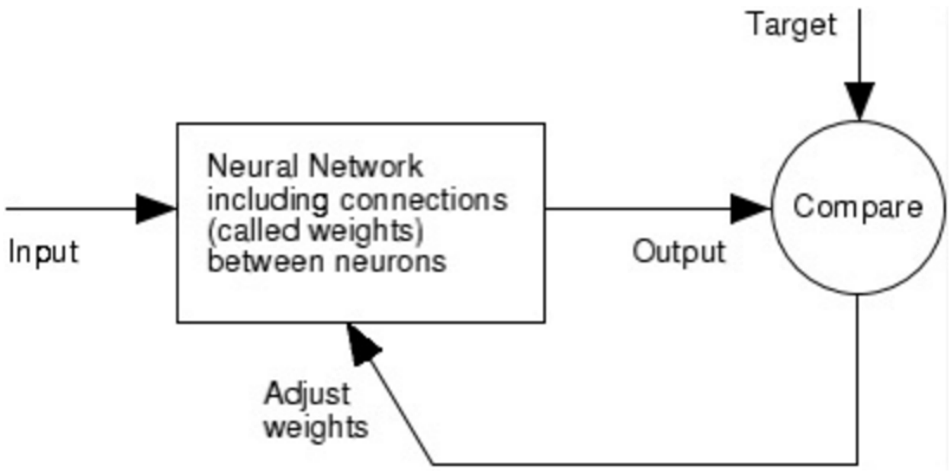


Figure 2.5. Neural Network Adaptive Control

These networks have been and are widely used in many fields like aerospace, for example for auto pilots, flight paths, fault detection, also are used in automotive, finance, for example fraud detection, telecommunications, like for example image compression, real-time translation. For this work it is important to understand the idea that this system can be used for possible failure detection in industry.

Matlab® offers the *Neural Network Toolbox®* that has a wide variety of architectures and training functions for modeling complex non-linear systems in a simple, man-made way. The applications available in Neural Network Toolbox allow the user to design, train, visualize and simulate interactively the network to later generate the equivalent MATLAB® code and therefore automate the process.

MATLAB®, with this tool, permits the user to use artificial neurons simulations, to work in several different ways, *Figure 2.6*:

- CURVE FITTING
- PATTERN RECOGNITION
- CLUSTERING
- DYNAMIC TIME SERIES

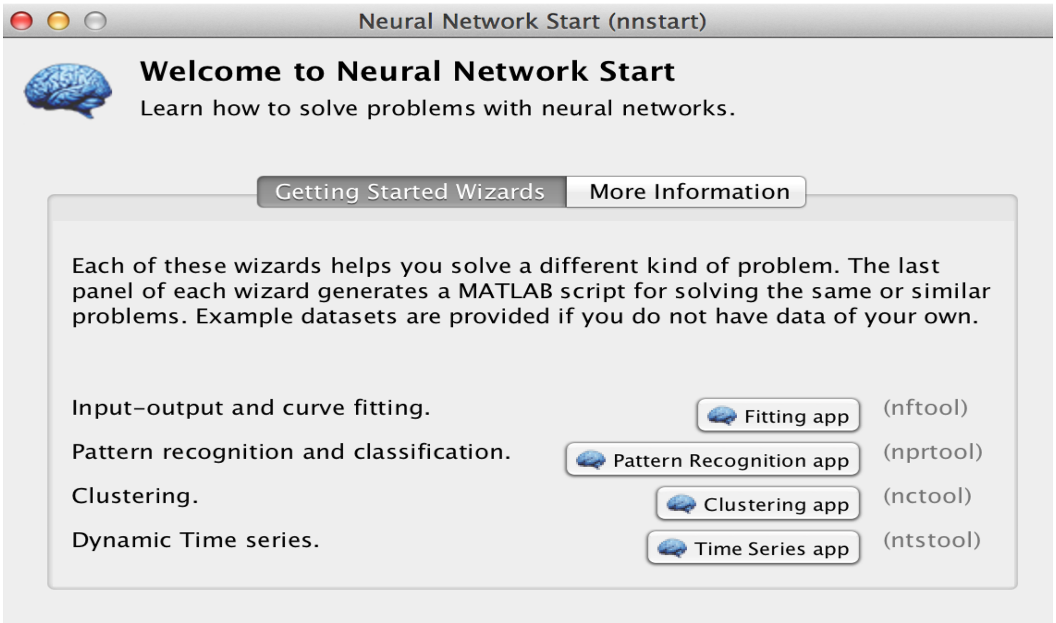


Figure 2.6. Neural Network Matlab® - Menu

For the present work it is going to be analyzed how to implement the Pattern Recognition Neural Network strategy. In a basic way, what is pretended with this strategy is to analyze all the data provided to find possible patterns in it, that are not possible, or very difficult to recognize for the user only looking at this data.

For getting this section activated *(Figure 2.6.)* it is only necessary to write in the command window of MATLAB® "*nnstart"* and select the pattern recognition option.

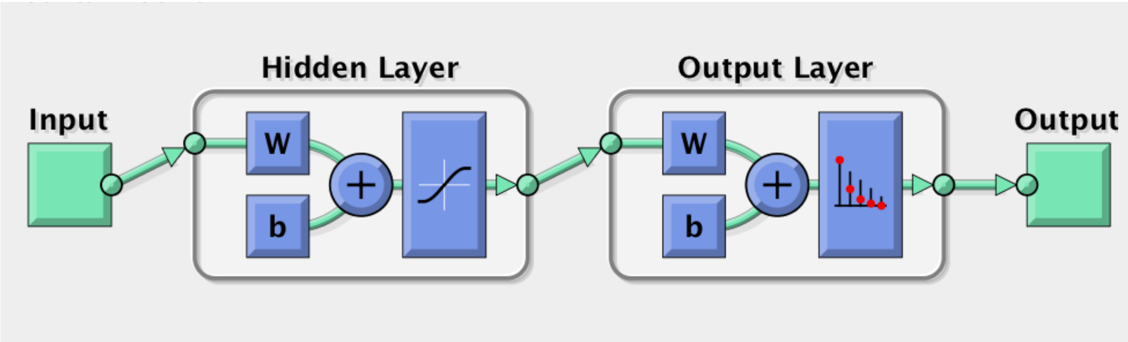


Figure 2.7. Neural Network Matlab ®

Once having selected the said option, this *(Figure 2.7)* is the neural network that will be applied; pattern recognition is the process of training a neural network to assign the correct target classes to a set of input patterns. Once trained the network can be used to classify patterns it has not seen before.

The variables have to be loaded to the system, these variables will be the INPUTS, which basically is the organized data that wants to be analyzed for looking for any type of pattern, and then it has to be loaded a variable that represents the target values, these means that for one sample of data, (i.e. is selected one sample of every variable containing the information of the inputs) it has to be provided a target value, this is the "ideal" or "expected" value that has to be outputted by the system making some combinations of functions with the input data.

After entering al the data, it has to be decided the number of neurons that are wanted to be trained, by default 10 neurons are selected, if an increase of the number of neurons is decided to be applied, it would more easily find some pattern, but it would be needed much more computation time, if volume of data is not very large it is a good idea to use higher number of neurons, but if high amount of neurons are used when treating large volumes of data this will be unviable considering time restrictions.

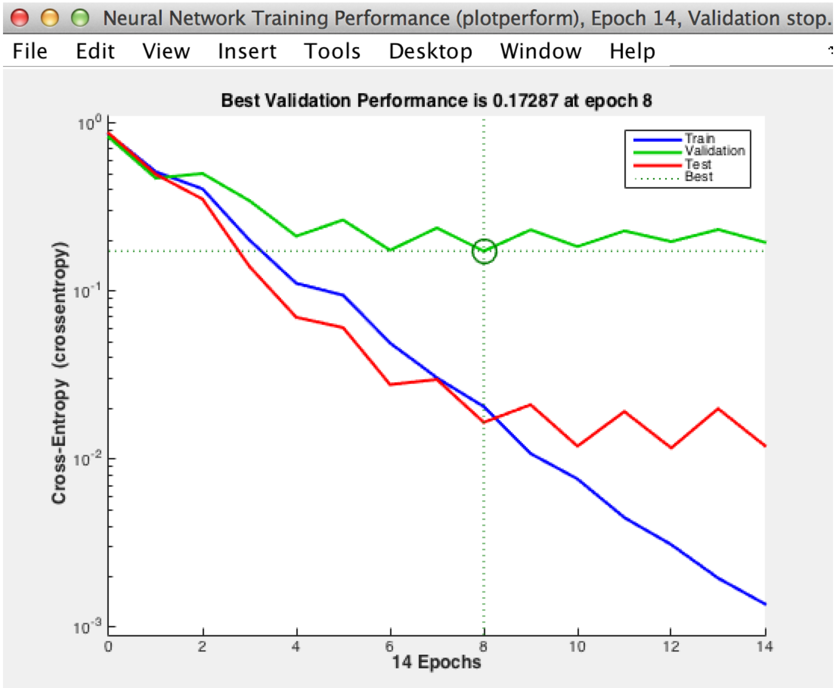


Figure 2.8. Training, Validation and Test Error's

The tool, looks for patterns in data in a "*try and failure*" way, i.e. when it is trained, it looks with small amounts of data for some pattern, and when it founds some type of pattern that it can be in some way modeled, it tries to validate it with higher amount of data. The program does this with limited amount of data, instead of doing it with all the data available, to reduce compilation time.

This process is done until the error made in the validation part with respect to the target values, is constant and does not decrease any more.

This idea can be seen in the next graphs *(Figure 2.8. and Figure 2.9)* where it can be seen that even though the training and the testing error, continues decreasing with more iterations done, the tool selects the point (with a circle) in which the error stabilizes.

Also for every iteration, the rest of data that is not used in the training nor in the validation part, it is used to test the model obtained.

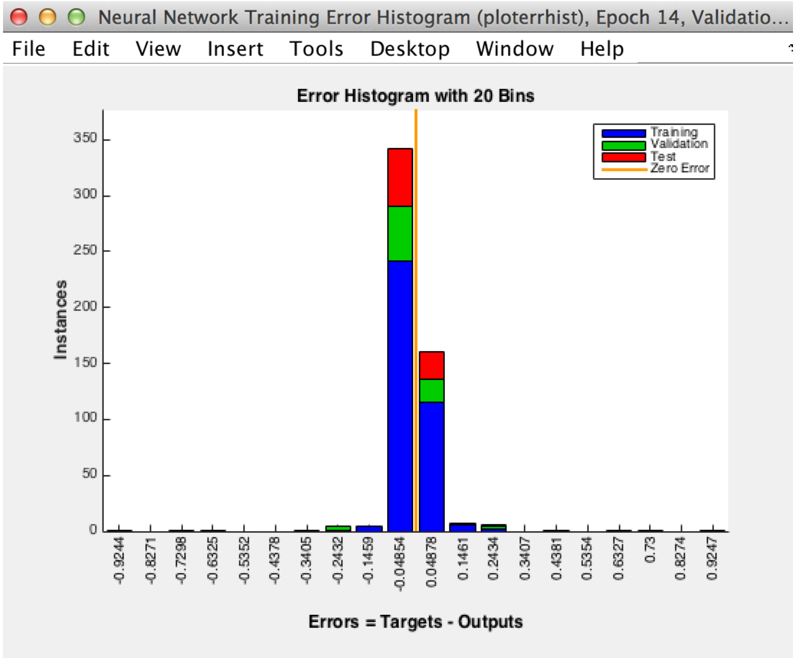


Figure 2.9. Error by Neural Network

Figure 2.10. Neural Networks' Results

Finally this last graph *(Figure 2.10.)* is the most important graph to understand the results obtained. In the example used to present the tool, it is seen that with the data introduced it can have three different outputs, represented in the boxes as 1, 2 and 3.

Each green box in the center represents the percentage and amount of the analyzed data that the tool calculated as fitting in a determined type of output, and that it corresponds with that type of output in the target information provided.
The red boxes around represent the data that was erroneously classified, i.e., the data that by the model was classified in a determined output, but comparing it with the target it didn't was coincident.

Each of the four box corresponds to a determined situation, it can be seen that it says *Training*, *Validation* and *Test,* each of this set of boxes represent the results obtained for each activity done, and the last set of boxes corresponds to the total data, the sum of the results of the described three situations. It can be seen that in the set of boxes that describes all of the data, if all the portions of data described are added it would give the total number of data that was introduced as an input to the tool.
It is important to remark that is very difficult to obtain a system in which it can be found a model that correctly classifies 100% of the data, this is due to the fact that it is very difficult to enter data totally correctly and even less when the volumes of data that are being processed are very large, so values higher or equal than 95% of success **in each type of output** will be considered as "perfect" results.

As described at the beginning of the section, the most important idea about Neural Networks is the ability to learn to perform tasks based on training or an initial experience and that it can create its own organization or representation of the information it receives through a learning stage.

Adaptive learning ability is one of the most attractive features of neural networks, due to the fact that it learns to perform certain tasks through training, because neural networks can learn to differentiate patterns through examples and training, it is not necessary to develop prior models or need to specify probability distribution functions.

Neural networks are self-adaptive dynamic systems. They are adaptable because of the self-tuning ability of the elements (neurons) that make up the system. They are dynamic because they are capable of constantly changing to adapt to the new conditions.

But it is possible to find disadvantages in neural network systems because the ability of neural networks lies in its ability to process the information and nowadays machines can take very high amounts of time if the volume if information is very high. Due to the fact that time is one of the most important resources, this makes quite often Neural Networks to be not viable solutions to a problem.

2.3. PLATFORM AND INTERFACE

the above objectives have been satisfactorily developed it will be necessary to create a platform and an interface that is simple and fast to be able to determine, through operators or some other system of the company, the conditions in which the system is located for each time instant.

This interface will be developed from the critical parameter (or parameters), the one that has been found to have a direct relationship with the real operative conditions of the system, that with which a quick analysis of the situation could determine the point where the process is, at an operative level.

To develop this interface you could use many programs, where can be highlighted JAVA®, Lab View® or MATLAB®.

For this work, as in other sections the decision has been taken to use MATLAB®, this program allows the use of a tool called GUI. MATLAB GUI® is a standalone MATLAB® program with one that can automate a task or a calculation.

CHAPTER 3. APPLICATION TO PRACTICAL CASE

In the particular case of this work, it will be worked with a machine that fills the coil of refrigerators with the cooling fluid, called FRIGUS, and it is wanted to discover a trend that can determine when the machine is going to fail.

In the system under observation, the department consists of eight of these machines, each of them operating on two boxes (called from now on IDbox, where for example IDbox 1 and 2 are machine 1, IDbox 3 and 4, are machine 2, and on like that until IDbox 16, machine 8).

The FRIGUS machine follows an operating cycle which is the same for each refrigerator model worked; what changes is only the time required to complete the cycle. *Figure 3.1*.

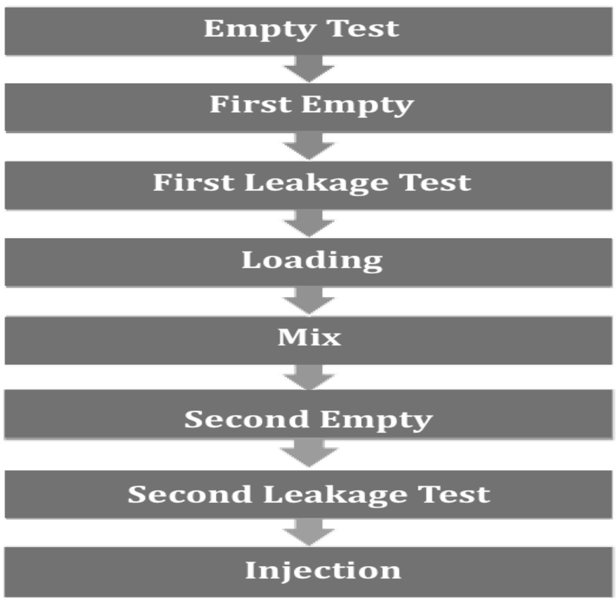


Figure 3.1. Frigus Cycles

A board machine, *Figure 3.2,* are mounted the sensors that provide the measurements recorded during the work cycle.

The working cycle is composed of several cycles visible in the previous diagram, *(Figure 3.1.)* each of which has a thresholds that, decided by the organization that if not respected, it is processed as a failed cycle.

The organization indicates that the cycle has failed and that the part has to be marked as defective. Defective parts are then sent to the repair department.

It is wanted to be identified a trend in one or more data values from the machine that can lead to consider a fault of the machine itself, with the objective of avoiding false positives, which are the pieces reported as defective, i.e., when failure is detected but everything works correctly, or at least doesn't cause the failure of the system.

To improve quality of the process, the organization wants to study how it can be prevented the failure of the machines, so it can be taken from now on an strategy of preventive maintenance to save resources, and also it wants to reduce the amount of false positives made by the sensors

.

Figure 3.2. Frigus

3.1. IDENTIFICATION AND PROCESSING OF DATA (PRACTICAL CASE)

First of all it must be observed (example in the clipping of excel, *Figure 3.3.*) how are distributed the data provided by the company and we must try to understand as much as possible what represents each one of them. This data is historical data from the years 2014 and 2015 of the company, *Figure 3.3,* and are going to be the base of all the study.

Serial Number	C12NC	IDMachine	IDBox	Data	Esito	Valore Vuoto Test Iniziale (Allarme > 100 Pa)	Valore Vuoto Evacuazione Pa (allarme >500 Pa)	Valore Vuoto Prova Perdite Pa (allarme >800 Pa)	Pressione Miscelazione kPA (allarme <3)	Valore Evacuazione Post Mini Pa (allarme >500 Pa)	Valore Vuoto Prova Perdite Post Mini Pa (allarme >500 Pa)	Refrigerante	Dose Program mata gr	Temperatura Refrigerante °C	Pressione Refrigerante kPa (allarme <1000 kPa)	Dose Iniettata gr	Esito/Codice Errore (0 = OK)	Delta dose
301422010613	854907986007	2	14	2014-06-03 07:11:56.723	1	1	42	266	76	58	314	r600	60	17,3	1500	60,7	0	0,7
301422010612	854907986007	2	13	2014-06-03 07:11:58.223	1	1	48	211	62	84	357	r600	60	17,4	1435	60,7	0	0,7
301422010611	854907986007	2	16	2014-06-03 07:12:58.953	1	1	45	263	66	58	301	r600	60	17	1445	60,4	0	0,4
301422010610	854907986007	2	15	2014-06-03 07:13:00.627	1	1	48	285	72	81	412	r600	60	16,9	1384	60,7	0	0,7
301423001935	854907796007	2	12	2014-06-03 07:14:03.877	1	3	51	321	60	107	587	r600	92	17,5	1826	92,8	0	0,8
301423001934	854907796007	2	11	2014-06-03 07:14:11.957	1	3	55	340	63	100	555	r600	92	17,6	1768	93	0	1
301423007226	854903496006	2	7	2014-06-03 07:15:30.760	1	1	55	314	56	94	490	r600	86	17,7	1661	86,7	0	0,7
301423001933	854907796007	2	8	2014-06-03 07:15:31.927	1	1	51	305	53	94	513	r600	92	17,5	1741	92,9	0	0,9
301423007225	854903496006	2	4	2014-06-03 07:16:39.720	1	1	51	308	55	94	448	r600	86	18	1722	87,3	0	1,3
301423007227	854903496006	2	3	2014-06-03 07:16:41.223	1	1	55	321	53	146	763	r600	86	18,1	1653	87	0	1
301423007229	854903496006	2	2	2014-06-03 07:17:31.630	2	1	51	305	61	97	474	r600	86	0	1526	0	13	-86
301423007228	854903496006	2	1	2014-06-03 07:17:33.723	1	1	58	438	67	120	821	r600	86	18,4	1521	86,6	0	0,6
301423007232	854903496006	2	6	2014-06-03 07:18:36.143	1	3	51	214	69	90	334	r600	86	18	2149	86,9	0	0,9
301423007231	854903496006	2	5	2014-06-03 07:18:37.327	1	3	55	230	64	123	444	r600	86	18,2	2088	87,2	0	1,2
301423007230	854903496006	2	10	2014-06-03 07:19:43.520	1	1	45	282	58	94	500	r600	86	17,7	1636	86,9	0	0,9
301423007224	854903496006	2	9	2014-06-03 07:19:45.017	1	1	48	314	63	107	646	r600	86	17,9	1543	86,9	0	0,9
301423007223	854903496006	2	14	2014-06-03 07:20:40.737	1	12	45	305	73	71	389	r600	86	17,3	1509	87	0	1
301423001925	854907796007	2	13	2014-06-03 07:20:42.337	1	12	55	256	53	87	376	r600	92	17,5	1471	92,9	0	0,9
301423001924	854907796007	2	16	2014-06-03 07:21:39.817	1	9	51	321	58	71	370	r600	92	17,1	1426	92,6	0	0,6
301423001927	854907796007	2	15	2014-06-03 07:21:41.017	1	9	55	327	63	84	438	r600	92	17,1	1410	92,7	0	0,7
301423001926	854907796007	2	12	2014-06-03 07:22:41.320	1	12	55	344	60	103	555	r600	92	17,7	1808	92,8	0	0,8
301423001929	854907796007	2	11	2014-06-03 07:22:42.817	1	12	55	344	65	97	529	r600	92	17,7	1773	93	0	1
301423001932	854907796007	2	7	2014-06-03 07:24:00.127	1	12	55	337	52	103	526	r600	92	17,8	1675	93	0	1
301423001928	854907796007	2	8	2014-06-03 07:24:02.817	1	12	55	337	53	87	464	r600	92	17,6	1723	92,8	0	0,8
301423001931	854907796007	2	4	2014-06-03 07:25:19.317	1	19	58	347	48	129	649	r600	92	18,1	1707	92,9	0	0,9
301423001930	854907796007	2	3	2014-06-03 07:25:20.520	1	19	58	340	50	110	558	r600	92	18,4	1668	92,9	0	0,9
301423001923	854907796007	2	2	2014-06-03 07:26:02.207	2	16	55	327	57	97	493	r600	92	0	1530	0	13	-92
301423002710	854909796012	2	1	2014-06-03 07:26:08.910	1	16	71	539	62	107	711	r600	104	18,9	1520	104,7	0	0,7
301423002709	854909796012	2	6	2014-06-03 07:26:45.213	1	16	68	282	63	90	347	r600	104	18,2	2107	105,1	0	1,1
301423002708	854909796012	2	5	2014-06-03 07:26:46.707	1	16	68	288	65	120	451	r600	104	18,4	2067	105,2	0	1,2
301423004208	850722601002	2	7	2014-06-03 07:31:06.910	1	16	64	386	54	103	535	r600	102	18	1679	102,9	0	0,9
301423004209	850722601002	2	8	2014-06-03 07:31:09.620	1	16	64	379	51	107	532	r600	102	17,8	1722	103	0	1

Figure 3.3. Data for Practical Case

Each column represents:

1. **SERIAL NUMBER**: Reference code for each piece being worked out.

2. **C12NC:** Reference code for every operator that took the data.

3. **IDMachine:** Type of machine (is non-variant).

4. **IDbox:** Each one of the boxes (two for each machine)

5. **Date:** Is the date where the register has been done.

6. **Success:** It has a value of 2 if the process has failed, and one if everything has worked correctly.

7. **Empty Test Initial Value:** Maximum Value of pressure during the test.

8. **Empty Evacuation Value:** Maximum Value of pressure during the test [Pa].

9. **Empty Leakage Test Value:** Maximum Value of pressure during the test [Pa].

10. **Mix Pressure:** Maximum Value of pressure during the test [Pa].

11. **Empty Evacuation Test Value Post Mini:** Maximum Value of pressure during the test [Pa].

12. **Empty Leakage Test Value Post Mini:** Maximum Value of pressure during the test [Pa].

13. **Refrigerant:** Type of commercial refrigerant used.

14. **Programmed Doses:** Programmed dose of fluid to inject. [g].

15. **Refrigerant Temperature** [degrees Celsius]

16. **Refrigerant Pressure** [Pa].

17. **Injected Dose** [g].

18. **Error Code:** Code that determines the type of failure that has occurred.

This last parameter is important to take it into account since it is not a measurement, but represents the type of failure for which the machine has failed, which will be one of the most important parameters to consider from this moment on, during the development of this monograph. For the development of this study is not important to know what exactly means each code, but it is important to understand that it exists different type of failures, each represented by a number.

Together with this data, the company has provided data about the reparations done all over this period of time, and relative information to the maintenance interventions over the machines.

Once it has been analyzed the nature of each type of data, and with a more or less clear idea of what each data represents, it is passed to be read with the program MATLAB ®.

A common problem when working with data is the need to change the format in which the data has been printed by the company, to be able to read it by the usual computer programs.

In order to be able to process these data, it was often necessary to copy them to *.txt* files, in addition to excel, so MATLAB ® could read the data character by character to process it.

Once this data-reading problem was solved, and with all the data inside the computer system, the next step was the detection of strange values, those that were considered impossible to have, and that it was necessary to clean for the subsequent analysis. An example of this is the elimination of data provided with values of 0 for Serial Number, IDMachine, or for IDbox, which is impossible, since these parameters never adopted values of 0 as the company had defined it.

With this data eliminated, the next step necessary for the study and analysis is its grouping into blocks of data measured on identical systems, i.e., to divide all the data taken in blocks according to the IDbox, as a result independent blocks of about 10000 measures for every variable and IDbox where obtained.

Once the data were grouped according to the value of IDbox, analyzing how the measures were taken, it was observed that when a measure within the general cycle was a failure, the rest of the measurements that had to be done later were not done and were given by default value 0, due to the fact that these values do not correspond with reality, it was necessary to change them to other fictitious values.

It was decided to use the interpolation method to create these fictitious values. These values were interpolated with respect to the nearest values of the non-zero data for the same type of measurement, thus obtaining a mean value corresponding to the actual evolution of the measurements. These measurements could not be eliminated because are the defective ones and these measurements are the ones that store the faults occurred, and therefore, the data with important information.

In the next images *(Figure 3.4. and Figure 3.5.)* it can be seen the effect this repair had over the hole data for example this corresponds to IDbox 1 and variable ***Empty Leakage Test Value Post Mini.*** Where all the data corresponding to non measured values was repaired and extremely strange data was deleted.

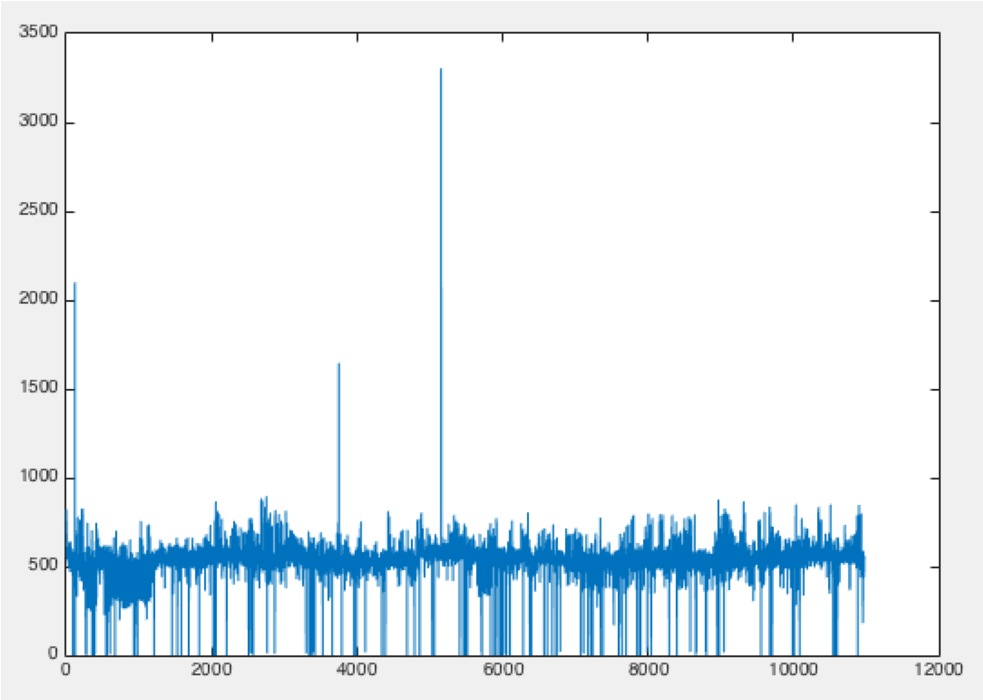


Figure 3.4. Data Before Treatment

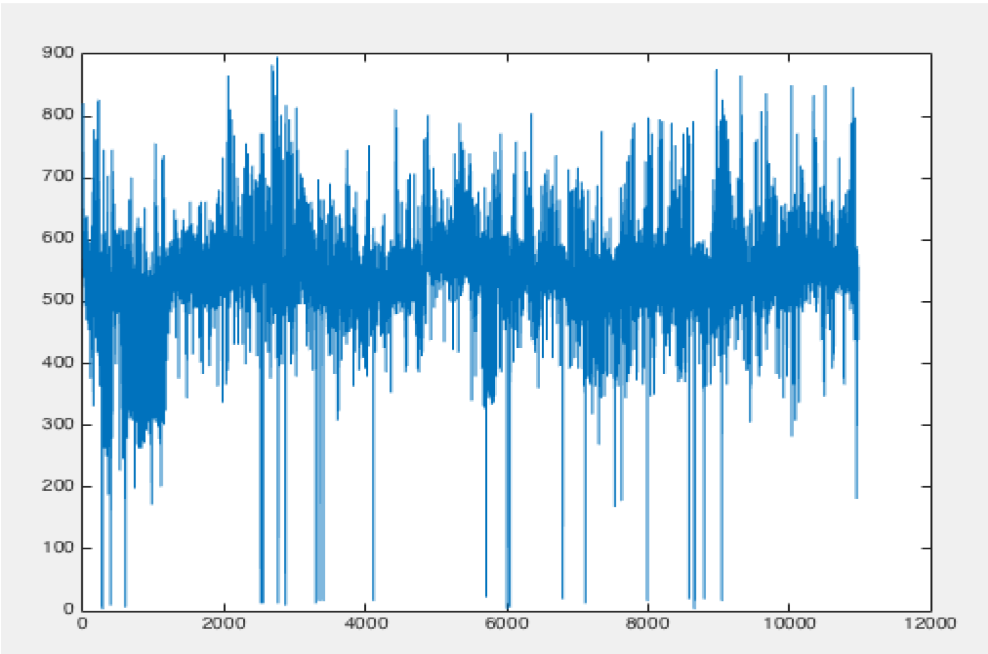


Figure 3.5. Data After Treatment

Finally, it can be observed that for each IDbox are had data sets measured between 192 and 196 different C12NCs, and taking as true the hypothesis that the values of the variables C12NC correspond to an identification code of the different operators that are in charge either of taking measures or of transcribing them, it was arranged to analyze statistically the percentage of failures measured by each different operator for each IDbox.

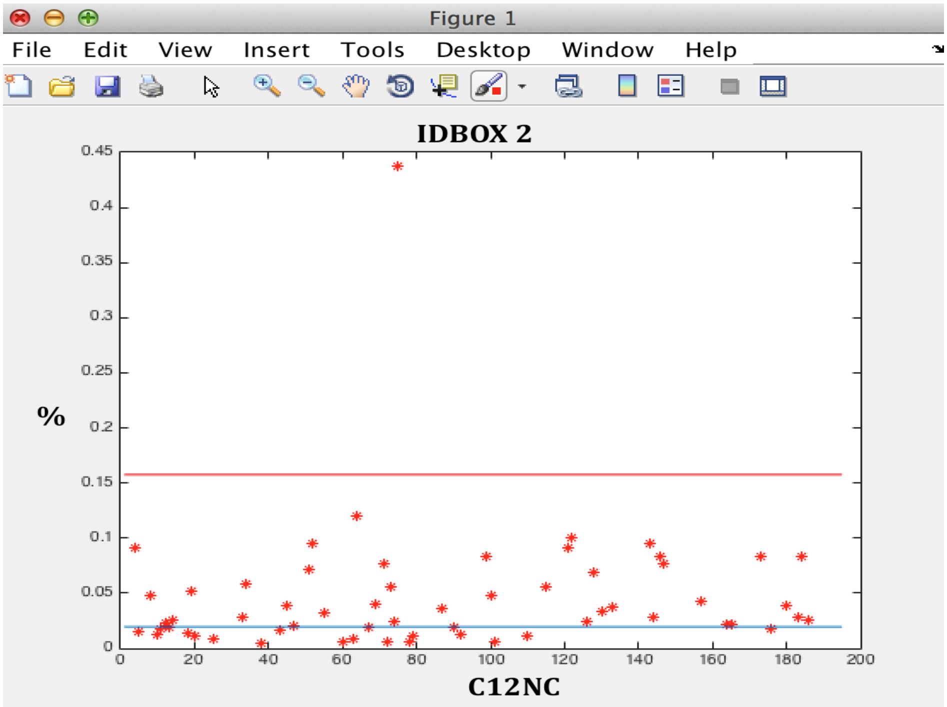


Figure 3.6. C12NC

These last graphs *Figure 3.6 and Figure 3.7* correspond to the IDbox 2 and 4 respectively, (in ANEX III all the graphs corresponding to each IDbox can be found), where each point corresponds to the percentage level of failures measured by each operator (each value of x-axis being an operator). It is quickly observed that most of the data correspond to percentage levels is rarely higher than 5%, but there are some anomalous data that even occasionally reach values near 45% of failures.

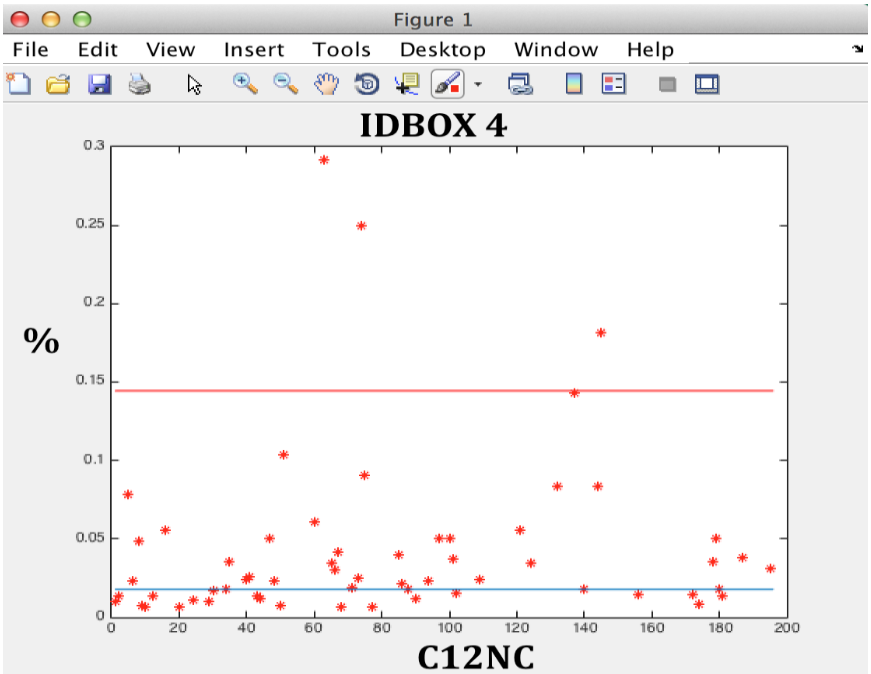
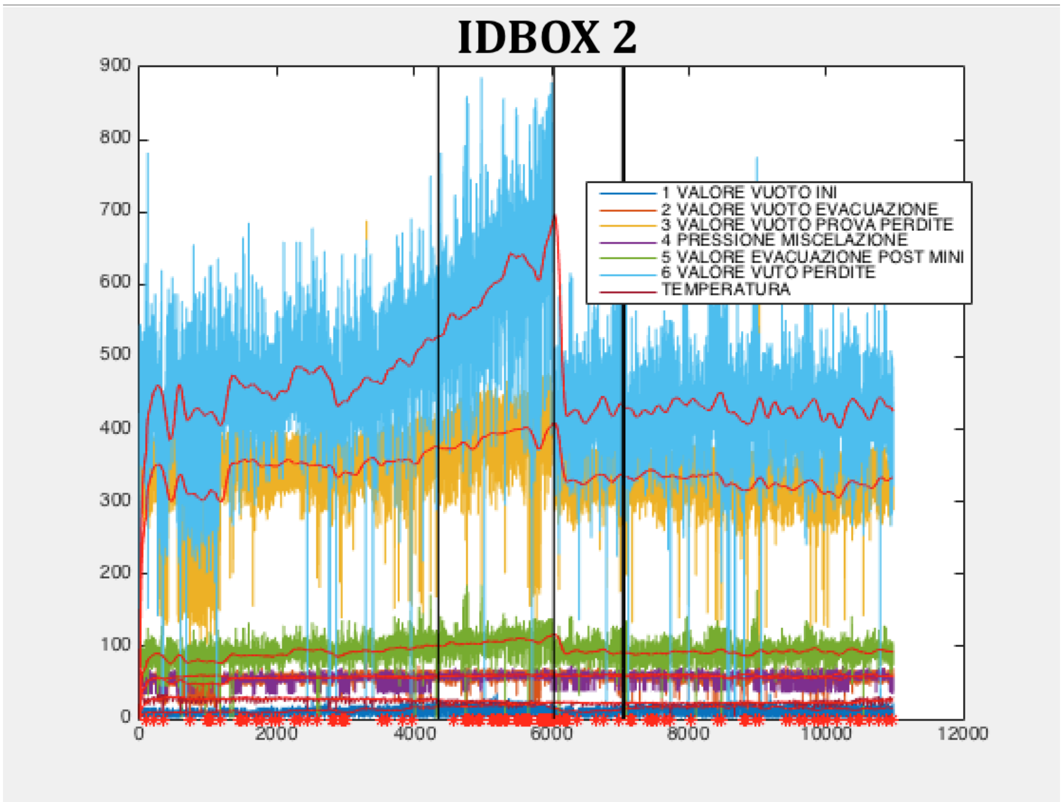


Figure 3.7. C12NC (2)

Taking into account that the total number of failures detected in the system in relation to the total number of instants in which measurements have been taken is 1: 100, therefore it is very probable that these high values of failure percentages correspond to erroneous or falsified data, therefore to proceed to the correct analysis of the data, it had been deleted all the values of measurements made by the C12NCs (operators) with these anomalous level of failures, because of the great probability that are not true and thus avoid polluting the future results.

After cleaning and processing all the obtained data, as results we can see the following graphs, which represent the IDbox 2 and 10. *Figure 3.8.*

(SEE ANEX IV FOR ALL THE GRAPHS OF THIS TYPE).



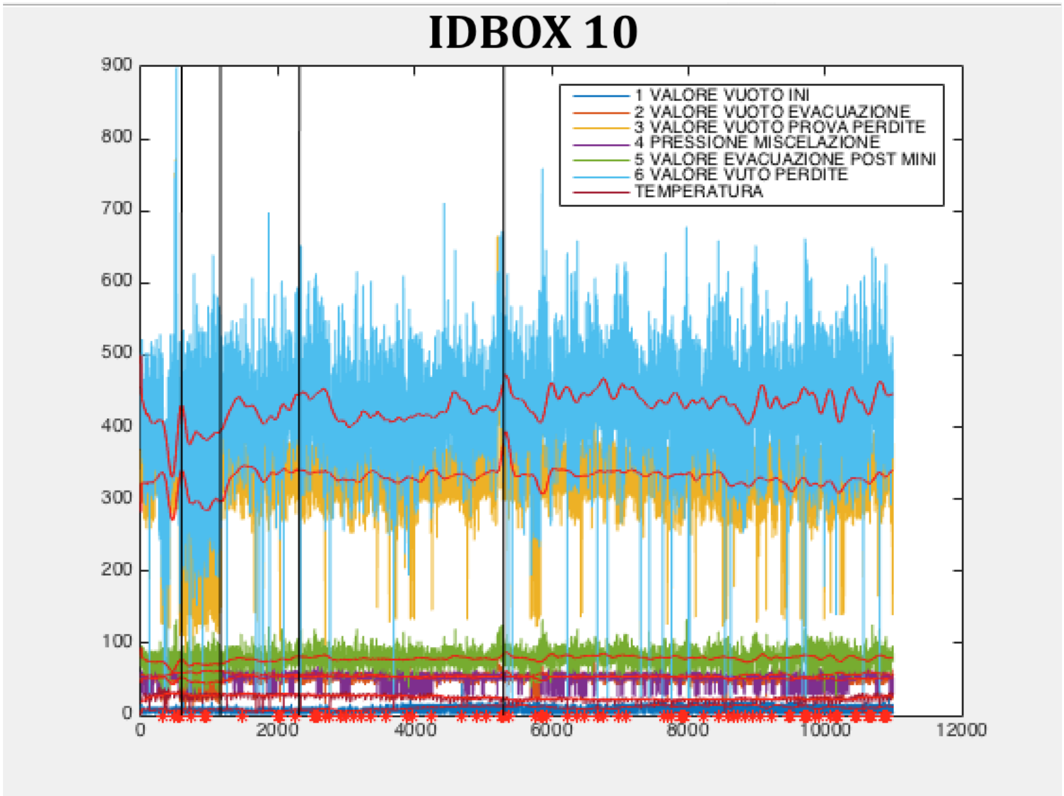


Figure 3.8. Data Processed

In these graphs all the variables of the cycles for each IDbox have been represented, in addition, for completing them with more information, it has been introduced as red dots on the x-axis, the instants in which failures have been detected. In addition also have been added to facilitate visually their subsequent analysis some vertical black lines representing any maintenance that has been done on the machinery, either a repair or a review.

And finally a red line that runs through all the variables, this red line represents the average done by the technique of moving average with a window of the last 100 instants values of data.

Once these results are obtained, the data is considered clean and processed, being able to pass to its analysis and study to detect or not, possible patterns.

3.2. IDENTIFICATION OF PATTERNS IN DATA (PRACTICAL CASE)

Once all the data had been cleaned and processed, it passed to the analysis of it, to try to find some pattern inside the data that can help to predict when a failure in the machine can occur.

The first problem when analyzing is that the system can have almost 50 different types of failure, and it is unviable to analyze all the different 50 types, because what occurs is that there are some types that occur only once in some machines, until other failure modes that occur hundreds of times in other machines.

To analyze the statistics of failure modes, it is necessary to know how many failure modes of each type occur for each IDbox.

	FAILURE MODE = 2	TOTAL FAILURES	% OF TOTAL FAILURES
IDBox 1	111	140	79,29
IDBox 2	142	176	80,68
IDBox 3	126	154	81,82
IDBox 4	152	183	83,06
IDBox 5	109	202	53,96
IDBox 6	86	195	44,10
IDBox 7	115	147	78,23
IDBox 8	151	181	83,43
IDBox 9	124	163	76,07
IDBox 10	88	130	67,69
IDBox 11	117	178	65,73
IDBox 12	88	139	63,31
IDBox 13	102	175	58,29
IDBox 14	111	181	61,33
IDBox 15	85	134	63,43
IDBox 16	108	155	69,68
		AVERAGE %	65,03

Figure 3.9. Failure Mode 2

	FAILURE MODE = 33	% OF TOTAL FAILURES	FAILURE MODE = 33	% OF TOTAL FAILURES
IDBox 1	2	1,43	6	4,29
IDBox 2	2	1,14	7	3,98
IDBox 3	2	1,30	8	5,19
IDBox 4	2	1,09	10	5,46
IDBox 5	43	21,29	16	7,92
IDBox 6	43	22,05	13	6,67
IDBox 7	4	2,72	9	6,12
IDBox 8	4	2,21	8	4,42
IDBox 9	9	5,52	3	1,84
IDBox 10	9	6,92	3	2,31
IDBox 11	2	1,12	17	9,55
IDBox 12	2	1,44	15	10,79
IDBox 13	8	4,57	17	9,71
IDBox 14	8	4,42	14	7,73
IDBox 15	9	6,72	15	11,19
IDBox 16	9	5,81	16	10,32
	AVERAGE %	5,61		6,72

Figure 3.10. Failure Modes 33 and 37

In this tables are present the three failure modes that most commonly occur, an it is easily seen how *failure mode 2* represents 65%, *Figure 3.9*, of the total failure in the system, but in some IDboxs, it represents nearly the 84% of the failure modes occurring on it. While the second and the third failures that most occur are failure mode 33 and failure mode 37, *Figure 3.10*, where both of them represent around 6%, of the total of the failure modes occurring in the system. The other failure modes not present in tables because they represent less 23% of the failures distributed in more than 40 failure modes, which for this analysis will not be important.

Looking at his results analyzing the statistics of failure mode, it is important to think about these results as a Pareto Distribution, or like an ABC distribution, where few failure modes concentrate most of the percentage of failures, this means in few types of data, lots of information is stored.

Looking at the results as an ABC distribution, where:

- A represents the most important part of the data, that has to be strictly controlled.
- B represents the medium importance data, where it is necessary some type of control, but not as exhaustive as A data.
- C represents the data that in some way must be more than controlled; it has to be at least taken in account that it exists.

It is clearly identified, that the A-type data, corresponds to failure mode number 2, while B will be failure modes 33 and 37, and all the rest of failure modes will be treated as C-type data.

To optimize resources, and time, it is important to analyze with much more detail the failure mode 2, so from now on, when the author talks about failure mode, faults, or fails of the machine, it will be referring exclusively to ***FAILURE MODE 2***.
If it wants to be talked about another type of failure mode it will be expressly defined of what failure mode is it talking.

For the first analysis strategy all failures detected in the system are taken to analyze the viability of some pattern that may follow the data until a failure, (if existed).

Later it will be seen that another mode of analysis will be necessary to do, but to start from the beginning it will be shown the analysis made taking as valid and correct all the failures detected by the system *Figure 3.11*.

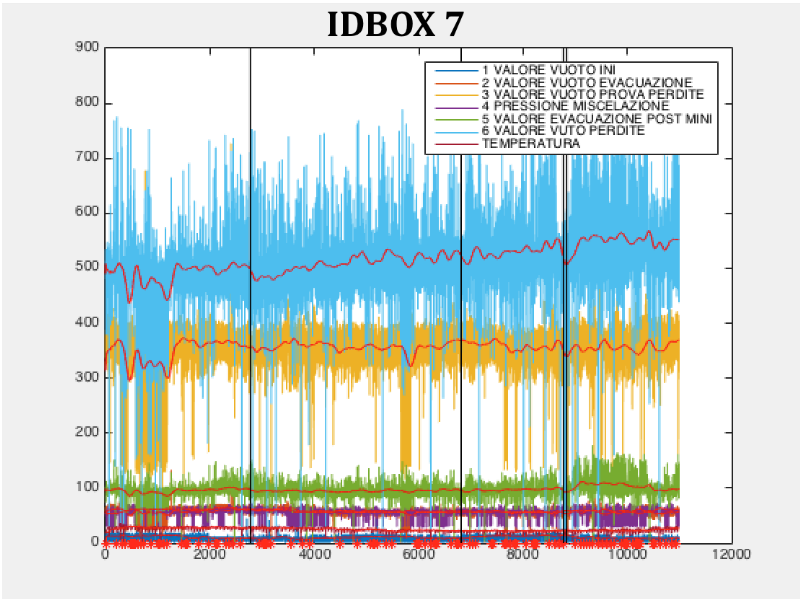


Figure 3.11. Data and Failures Detected

Visually it is impossible to detect any type of pattern of why faults occur (marked in red on the x-axis of the graph, to see all the graphs go to ANNEX IV).

Therefore, following a predictive logic, the sections of data between two consecutive failures are analyzed to try to detect some pattern.

As can be seen, we cannot see anything clearly, since it seems that failures appear randomly. By analyzing the frequency with which failures are detected in the system, a time statistic cannot be made, since these failures occur on intervals of time between very small to even very large intervals of time between two failures.

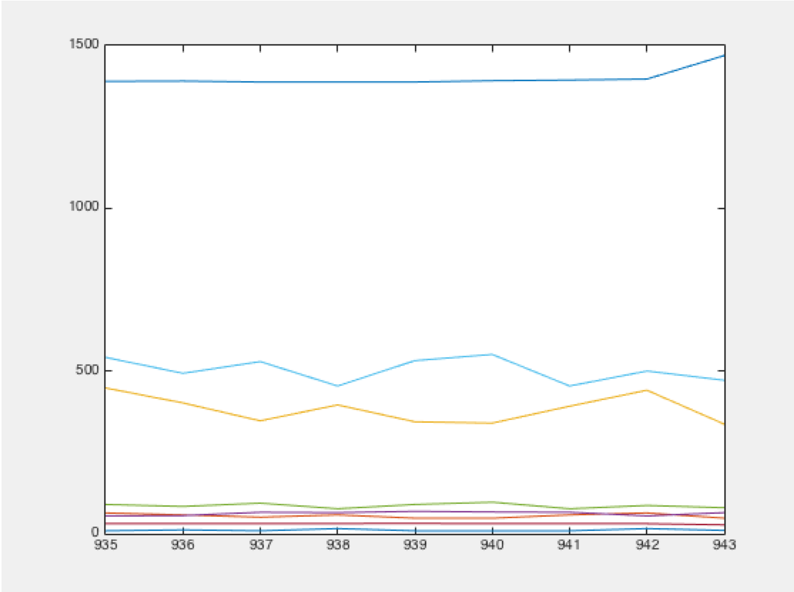


Figure 3.12. Data Between Two Consecutive Faults Detected

In addition, by looking at the evolution of the parameters measured from the system, *Figure 3.12 and Figure 3.13* no pattern can be visibly distinguished following the evolution of these data.

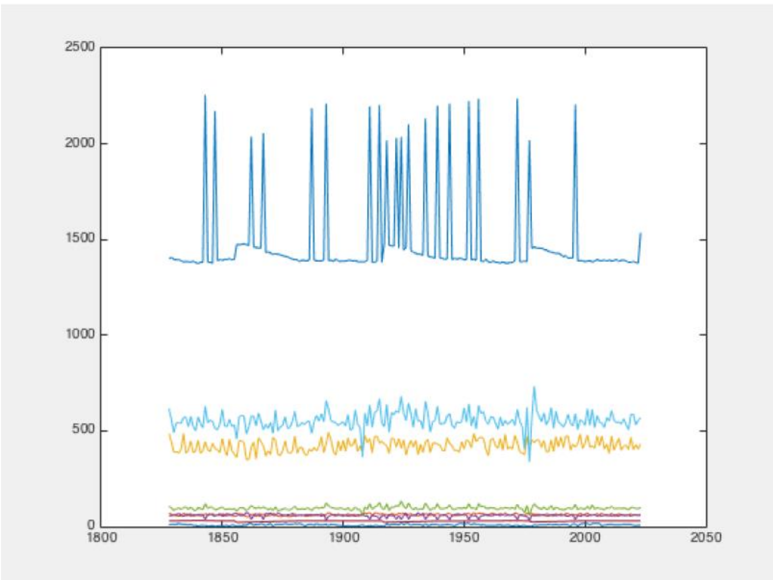


Figure 3.13. Data Between Two Consecutive Faults Detected – 2

Therefore for being possible to detect if there is any common pattern between detected failures, it will be preceded to analyze the data using the method or strategy called, Artificial Neural Network.

The method of analysis through Neural Networks was presented theoretically in section 2 of this work. What it does is to search through iterative analysis and through the use of artificial neurons, that have the ability to learn from past experiences, the existence of some pattern between the historical data and the actual results obtained for each time instant, in order to be able determine a model for future measurements.

In this case, the historical data will be the measurements of the parameters taken for all the different time intervals, and for the different IDboxs, while the results obtained will be the fact of a failure or non-failure for each time instant. These target results will be presented as a vector of 0 and 1, where this vector will have 1 in the boxes corresponding to an instant of time theoretically where a failure has been detected.

Figure 3.14. Neural Network Strategy - 1

This graph, *(Figure 3.14),* corresponds to the adequacy of the results, i.e. if the tool through Neural Networks has managed to find a direct relationship between measured data and detected failures, introduced as target values.

This particular case corresponds to the analysis of IDbox 1, in which more than 10000 measured data of each variable have been introduced having 8 variables that are measured, therefore more than 80000 datum, therefore with these volumes it is easy to realize that finding some pattern hidden in the data is a very complicated task without using some helping tool.

The box marked with an "A" corresponds to the level of accuracy obtained by analyzing with the "model" obtained from the Neural Network, the data provided with respect to the expected values. Initially this value seems very good since it can be read that it has done in 99.6% of the cases.

At this point it is very important to keep in mind what it is exactly being looked for, and the modus operandi of the tool.

The tool does not know to what does the user as a researcher give importance, in the case of this section the importance is given to the failures, not to the non-failures, therefore it must be known how much of the data entered gave a failure result in real time, and this corresponds to only 1% (due that for 111 time instances was detected a failure in the system for IDbox 1, while the whole amount of data entered is nearly 11k different instants).

At this point to help the understanding of the issue it is going to be done a parenthesis to explain a similar situation.

For example, imagine that we have a system of 10 balls, *Figure 3.15,* of which nine are blue and one is red, and we also have a list of variables that describe characteristics of the balls and we want to analyze the relationship between the characteristics of the balls And the color of these.

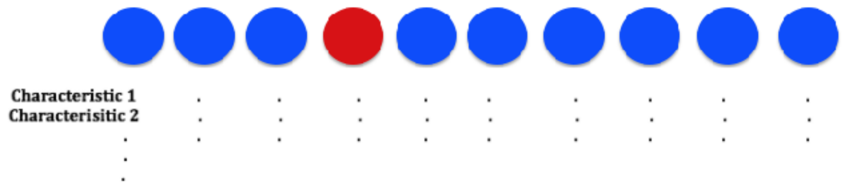


Figure 3.15. Colored Balls Example

It is assumed that in order to study a possible characteristic-color relation it is used a system of Neural Networks, and this finally gives as a general result that in 90% of cases it has been successful with the result value when doing the test.

This can mean a lot of things, but the only thing that exactly transmits is that 9/10 of the expected values coincided with the result obtained, i.e. it has a 10% of error. Therefore it is necessary to analyze more deeply the result, since to this fictitious study of colored balls what really interests to the investigator is to know if the red balls have different characteristics than blue balls.

The next step will be to analyze the proportion of times it has been said that a ball was red and then comparing with reality, it was not red, or, on the contrary, it was not said that a ball is red (i.e. that it has been said that is blue) and then turned out to be red.

Another assumption is made for this example and it is assumed that the 1/10 times of error regarding reality corresponds to that one has taken a ball like blue when in fact it was red. If this is the case, the model obtained by neural networks is not valid, since the main idea was to detect the red ball, and even having a 90% success, in no case that 10% corresponding to the red ball is detected.

Therefore if we consider the problem as well presented and the Neural Network system as reliable, it could be concluded that there is no real relation between the color of the ball and the characteristics of the ball (this result would be given for 10 balls exactly identical where the only difference between them is that one is red).

Returning to the results obtained by the analysis with Neural Networks for the case of this work, it must be observed section "B" of the previous table, since this section corresponds to the percentage of failures, in red it describes the faults that existed (according to the real system, target, introduced) and that the Neural Network tool has not been able to detect, and in green the failures that have been detected as such by both the real system and the Neural Network tool.

It can be read that 0.6% of the detected faults in the real system were detected by the tool analysis, while 0.4% of the faults detected in the real case were not detected by the tool. Taking into account that the failures only describe 1% of the total data, this means that 40% of the failures that were detected in the real system could not be related in any way to the measured data entered.

Therefore it is not a good result, because the data entered do not follow a clear pattern, since in the best case the detected pattern would only detect 60% of the failures detected by the real system.

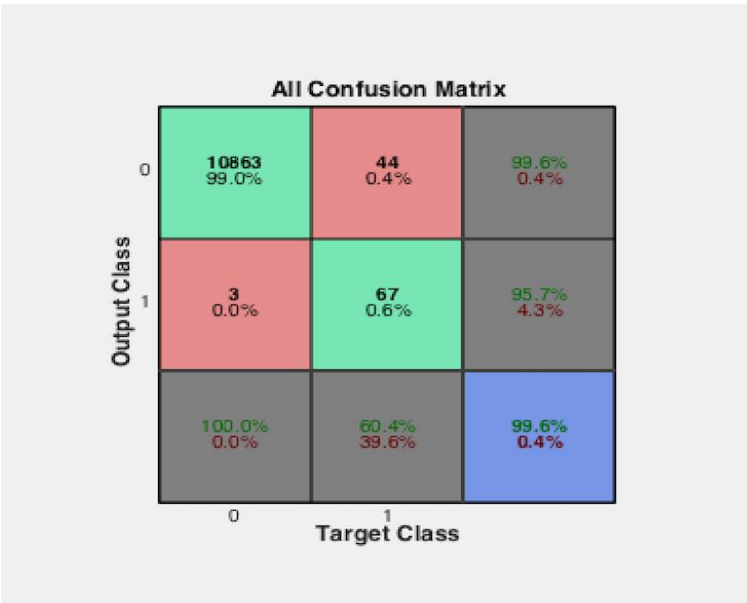


Figure 3.16. Confusion Matrix

Another analysis similar to the previous one has been done, but instead of analyzing each set of data introduced at each instant, it has been introduced the average of the last 100 instants for each instant, made using a Moving Average system with a window of 100 instants. The idea of using this method is to try to detect a trend theoretically, initiated in the previous instants before each failure was detected in the system. *Figure 3.17.*

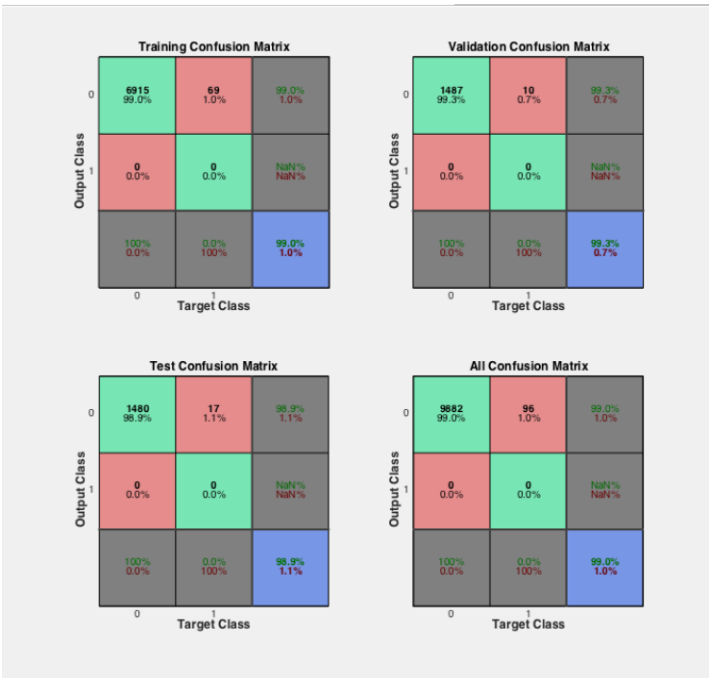


Figure 3.17. Neural Network Strategy -2

It can be seen that the results obtained of this case are even worst that the ones obtained for the previous one, due to the fact that no relationship is obtained with between the data introduced (the averages) and the target values introduced for each instants (the same as in the previous example).

At this point, the results obtained from the previous analysis have not confirmed the initial hypothesis that expected that was a clear relationship between the failures detected by the system and the data measured on each IDbox throughout the study period.

Discarded the previous analysis done, it is decided to change the strategy used for research and analysis of data and its behavior, being the author always convinced that there must be a causal relationship between the system failures and the data provided.

Apparently the system detects faults in a random way, but observing the data in detail, it can be seen that some areas concentrate more failures than others *Figure 3.18 and Figure 3.19*, and in most cases these zones of greater concentration occur prior to maintenance actions carried out by the company, and after these maintenance actions, the concentration of failures detected in the machines decreases.

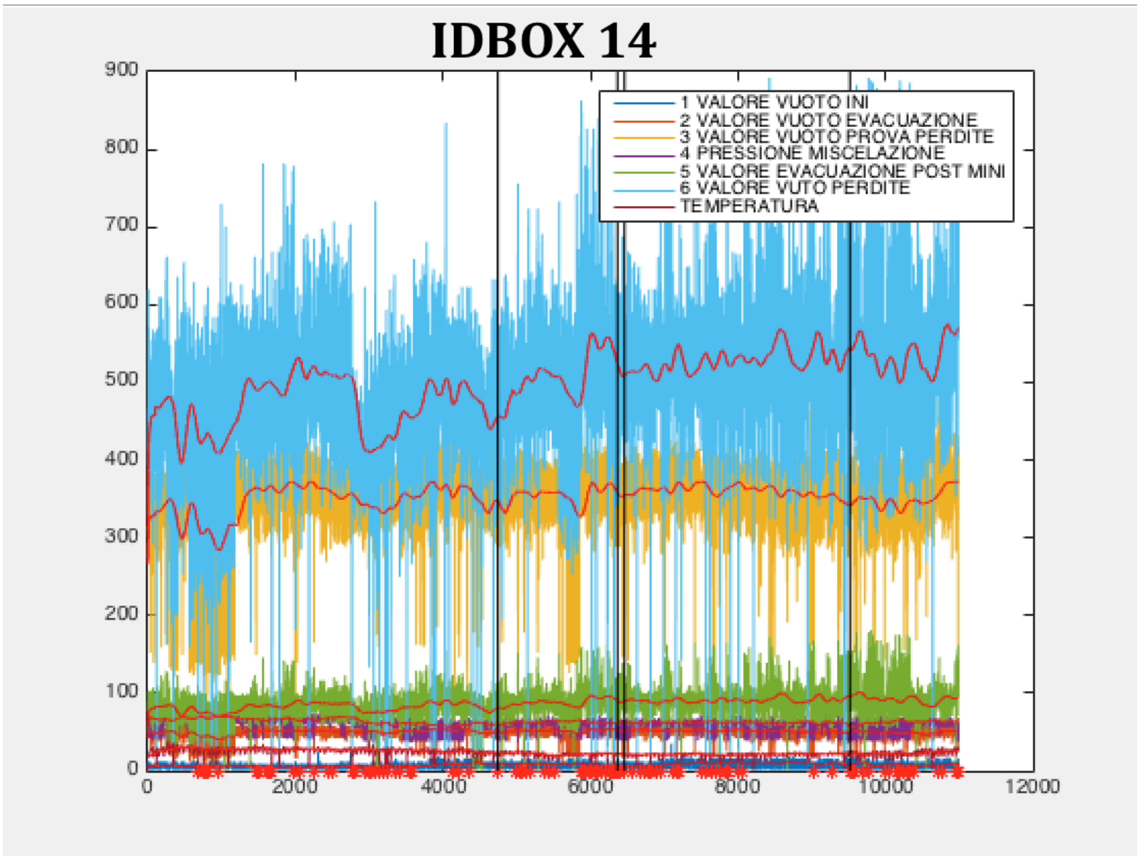


Figure 3.18. Data IDbox 14

For example observing variable and failure detection evolution for IDbox 14, it can be observed between the instant 6000 and 6500 a higher concentration of failure detections just until some maintenance actions have been done.

In order to continue from this point it is important to change the way failure detections are understand.

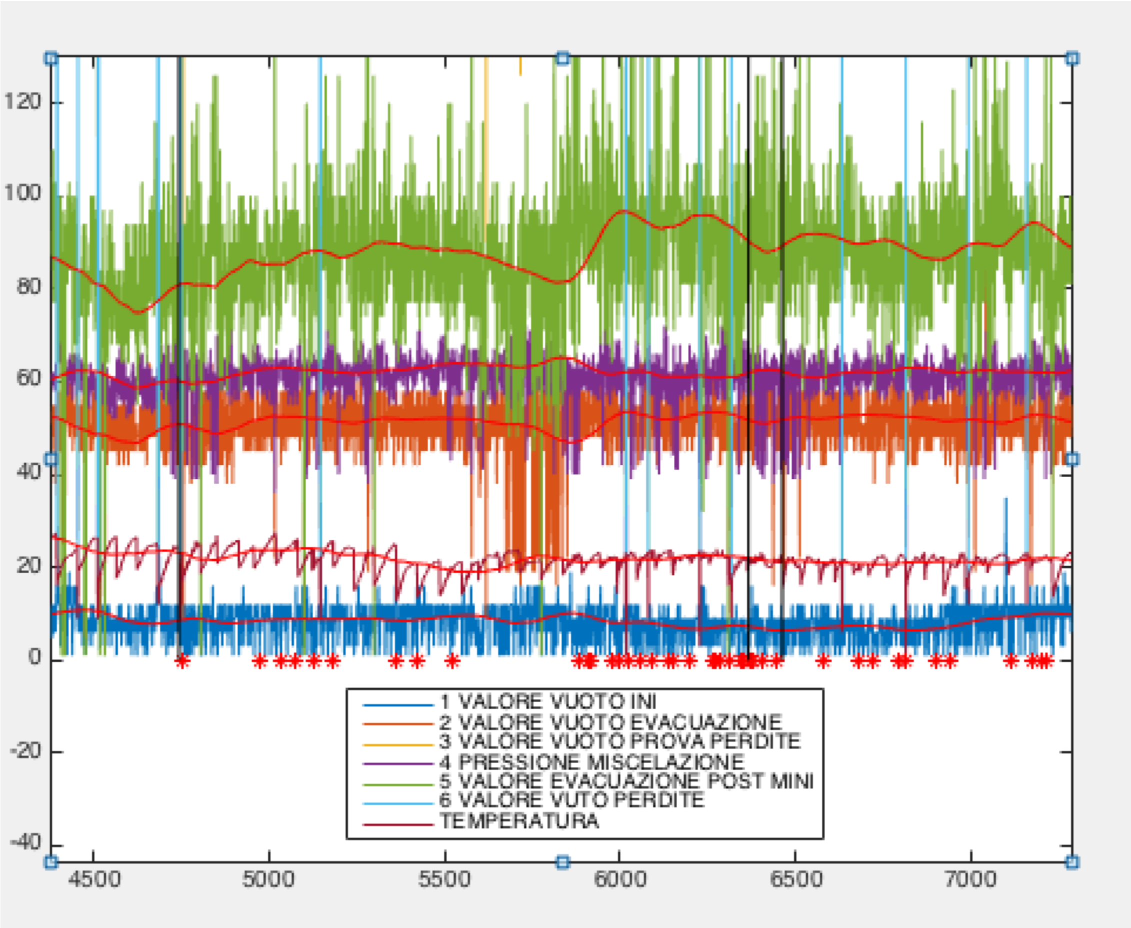


Figure 3.19 Frequency of Appearance of Detection Failures

It is clear that the fault detections do not cause the system to be blocked, as this would require a maintenance action just after each time a fault has been detected. This demonstrates that these failure detections do not mean that there is in principle a critical failure of the process.

Due to this, from this moment, these failures will not be treated as a failure in itself, as it has been treated so far, i.e. it loses the role of CONSEQUENCE produced by the data, and will adopt the role of CAUSE, that means to say these values will be treated as if it were some variables of binary nature of the system.

Explaining this idea with an example:

Imagine a light bulb that starts flickering from time to time, the fact that the light flickers (failure detection) does not mean that the light is damaged (blocking), but rather it can be an indicator that something is starting to system is starting to fail, and as the light begins to flicker with a higher frequency could mean a relationship with its current deterioration, reaching a point where the frequency of flicker is excessive including coming to consider the light completely damaged.

Analogously to the above example, it could be used the fact that failures occur in the system rather than a concrete failure, use it as an alarm system to measure the current deterioration of each IDbox.

Of all these ideas expressed to understand the new conception about detected faults, a new concept is presented that will be called *DENSITY OF FAILURE*, *Figure 3.20,* it consists in giving to each instant a value corresponding to the sum of the detected faults in the last 1000 instants of time.

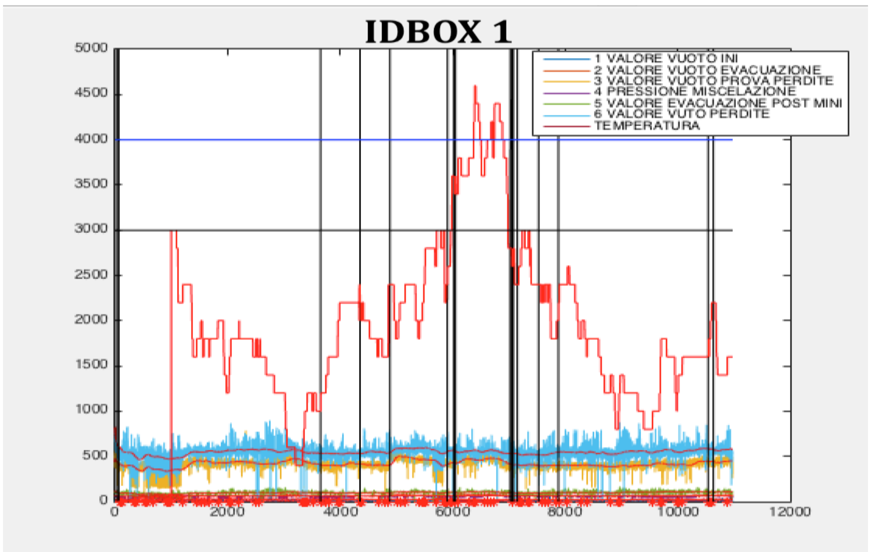


Figure 3.20. Density of Failure

This graphs show the results obtained when applying the *DENSITY OF FAILURE*. To make it more visual in the graphs the parameter has been calculated as said before with the sum of the detected failures in the last 1000 instants of time but multiplied this value by 200. This value of 200 is only a way of increasing the amplitude of the function so the scale is similar to the scale used in the other variables and can be visually recognized its trends in an easiest way. *Figure 3.21 and Figure 3.22.*

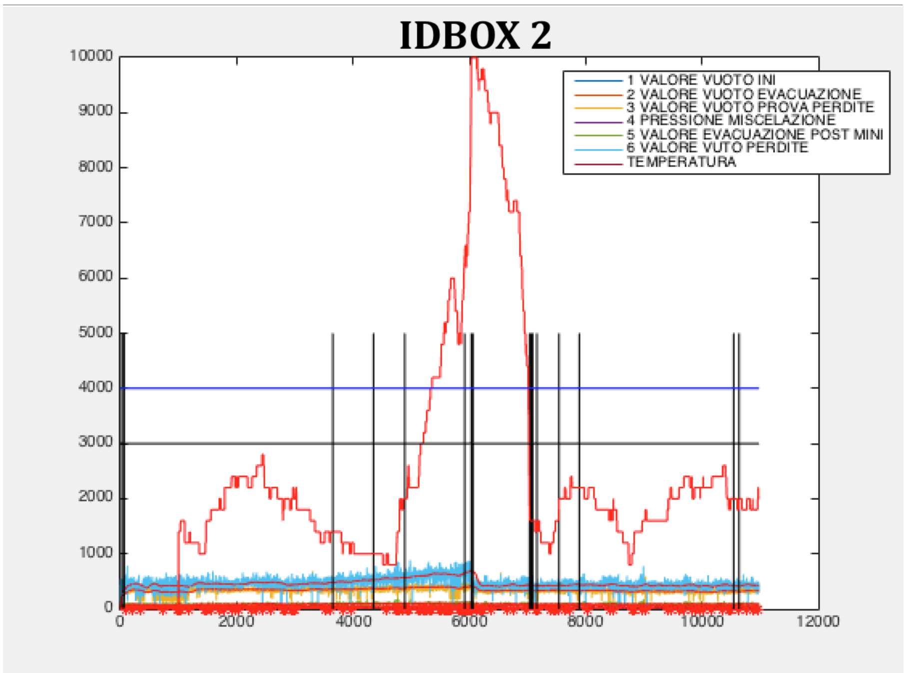


Figure 3.21. Density of Failure - 2

Therefore, it should be clarified that, for example, when the density level is 3000 units, it really means that 15 faults have been detected in the last 1000 instants of time, and if the value density was equal to 2000, 10 failures have been detected in the system in the last 1000 instants of study.

```
Density_of_fault(i) = Detected_faults(i-1000:i)*200
```

Observing the graphs on which the DENSISTY OF FAILURE function has been applied (can be seen for all IDboxs in ANNEX V), it can be distinguished two trends in the evolution of the density function over the period of study of the machines.

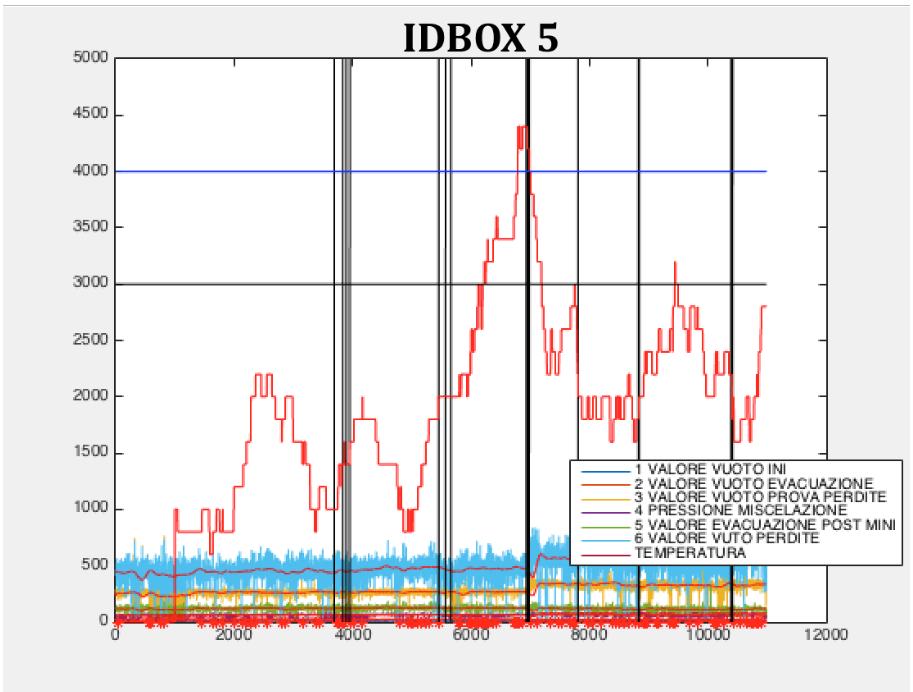


Figure 3.22. Density of Failure – 3

It can be distinguished as for all different IDboxs the density level is usually oscillating and as a rule less than 3000 (15 failures in last 1000 instants), except for some peaks in the function that stand out because their level increases to values very higher than the rest. It can be observed as for all cases when the level of density suits values strangely superior to the rest of the period (see case IDbox 2).

Then, seeing this mode of evolution of the parameter, the author decides to create of two different thresholds in the system, i.e., two level lines to differentiate three different states within the operation.

If the value of the density is less than 3000 (15 faults detected last 1000 instants) it will be taken that the process is working correctly.
If, on the other hand, the value of the density acquires values greater than 4000, the system will be taken as malfunctioning.

The latter is due to the fact that whenever this level of density is exceeded in all cases (arriving as it has been seen to levels sometimes around 10000), the tendency that the function acquires is of continuous growth, that is to say an increase in the number of detected failures, and to reverse this tendency it is observed that maintenance actions must be carried out (vertical black lines in the graphs) and then as a rule the density level returns to levels ranging from values lower than 3000 units of value.

Therefore as said before, if the density exceeds 4000 units of value, the system will be taken as malfunctioning, not desired situation.

The functioning of the IDboxs is divided into three different levels determines by the established thresholds, and a color is assigned to each one of the levels.

- Correct behavior (GREEN).
- Improvable behavior (YELLOW).
- Malfunctioning that will lead to failure of the system (RED).

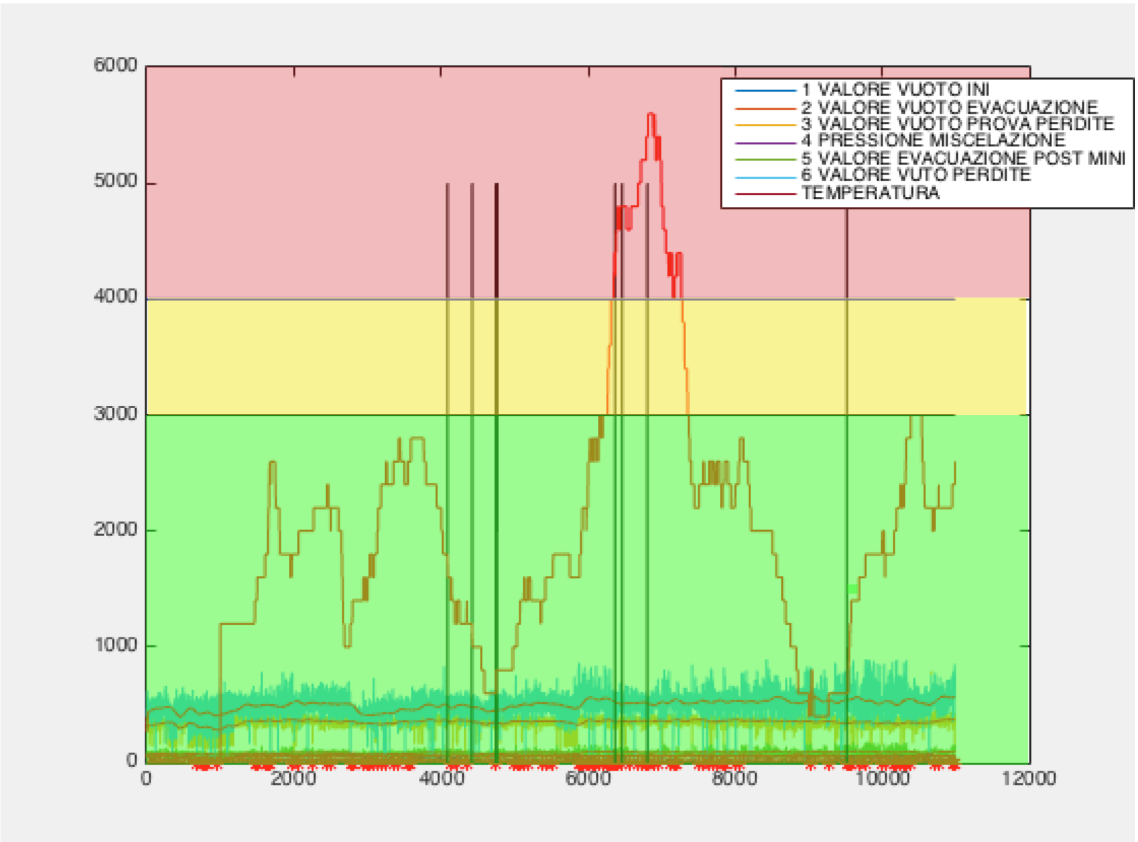


Figure 3.23. Alarm Levels

Therefore from this moment the system is divided into green, yellow and red levels, *Figure 3.23.* The green and red levels are simple to understand, as they represent whether the machine is OK (green) or NON-OK (red), while the yellow level is an alarm level in the system with respect to the fault, not necessarily if it is reached the yellow level means that there is going to be a fault, but in most cases.
The yellow level could be considered as an extension of the green level, since the machine is still at a good level of operation, but where an alarm system is activated to indicate that a fault is likely to occur sooner.
Later on, maintenance strategies will be proposed for this system in which the yellow level will play an important role, (SEE SECTION 4).

Once this possible failure prevention method has been analyzed, it will be analyzed by means of the Neural Network tool to determine the feasibility of this method, i.e., if there is a correlation between the level of function in which the IDbox is in each instant of time, and the data measured form the system, or on the contrary there is no clear relationship.

The data entered, as input in the Neural Network tool does not include system failure detections since it is clear that the level of the function *Density of Failure* depends on the number of failures that have been detected previously, therefore entering this parameter as input could contaminate and distort the result.

First of all due to the methodology to use the Neural Network tool, which determines that it should be introduced, apart from the system's input that in this case are the measured data, also must introduce some target values, i.e., some expected values that have been obtained by analyzing the historical data.

By definition it is very important the way in which these data are introduced, which can only be introduced as already seen in previous sections of this work, by means of a binary matrix.

This matrix must be represented with all the necessary columns where each one of them will represent a time instant in this case, and each row will represent a possible result, therefore by column only there can be a unit value since the result is unique for every moment.

Using an example to explain the ideas expressed above, imagine that there is an interval of instants in which the system passes from a correct operation functioning to a total failure level, passing through the intermediate level. *Figure 3.24.*

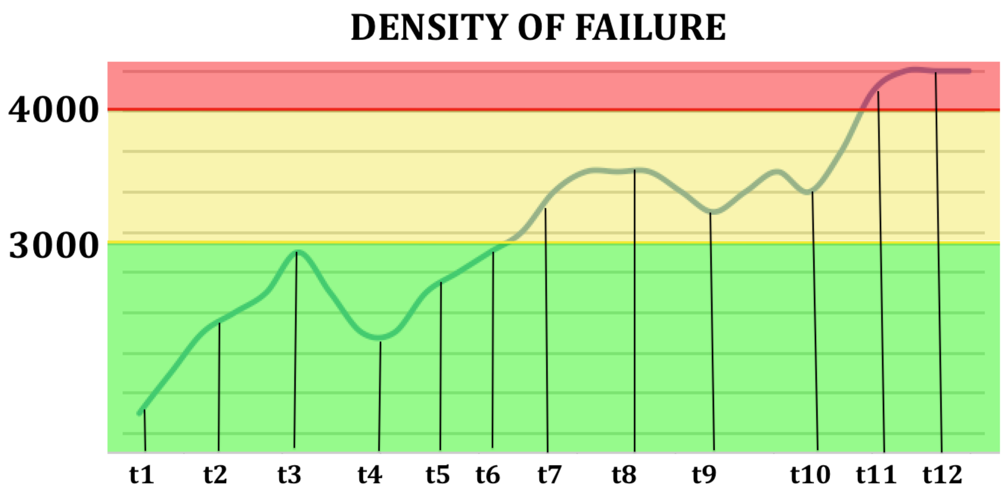


Figure 3.24. Example Target Value - 1

Where if data about the condition if it was expressed as a vector will be, *Figure 3.25*:

Figure 3.25. Example Target Value - 2

And transforming this information to a binary code [3,T], three rows and the same number of columns as time samples. *Figure 3.26*.

```
taget_data =

     1     1     1     1     1     1     0     0     0     0     0     0
     0     0     0     0     0     0     1     1     1     1     0     0
     0     0     0     0     0     0     0     0     0     0     1     1
```

Figure 3.26. Example Target Value – 3

When the results obtained from the analysis of the failure density have been changed in format to that of the target matrix presented, the analysis can be carried out using the Neural Network tool. *Figure 3.27.*

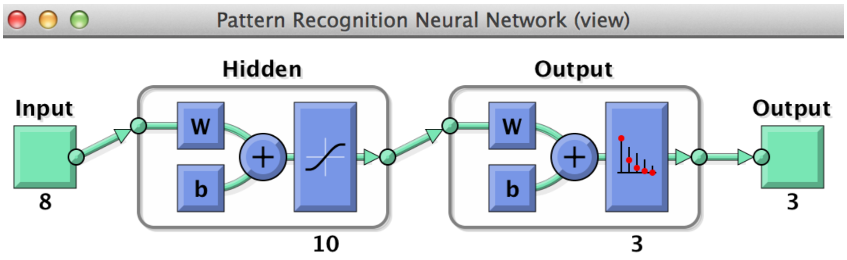


Figure 3.27. Neural Network Tool

Figure 3.28. Neural Network Results for "DENSITY OF FAILURE"

To carry out he analysis of pattern recognition, by default are used ten neurons in the network, and eight different variables are introduced as input, expecting to recognize a pattern between this data introduced and the three different types of output that are defined (Green, Yellow and Red Levels).

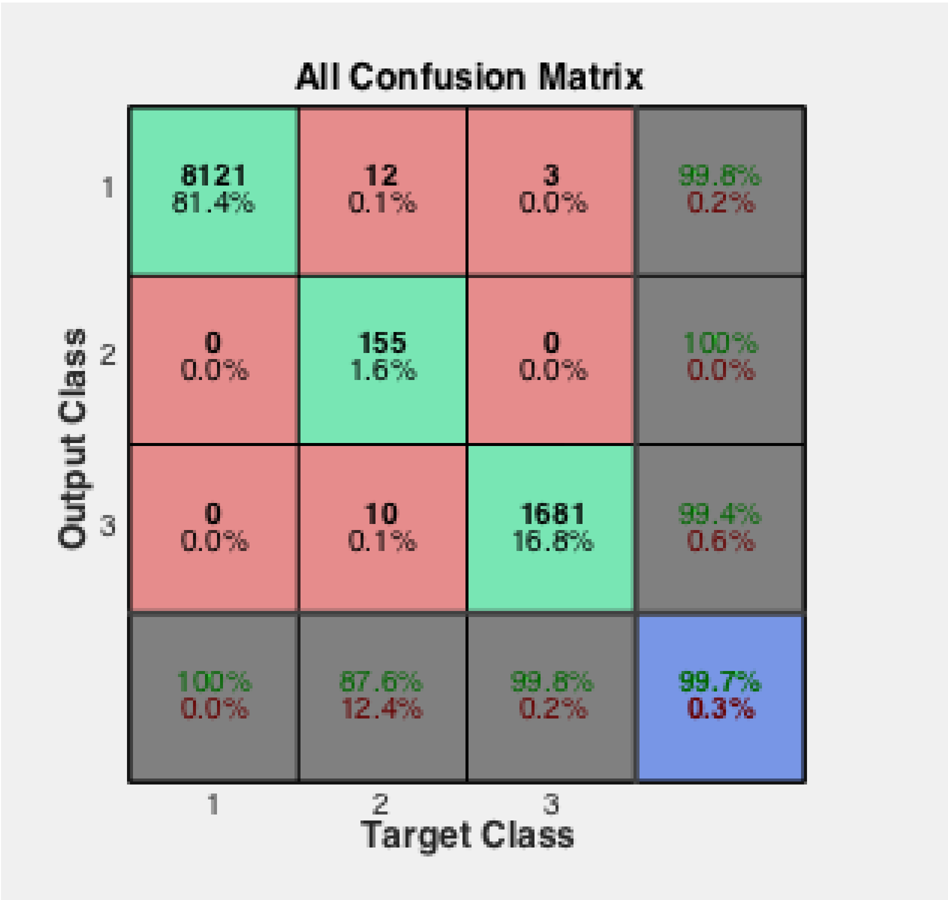


Figure 3.29. Neural Network Confusion Matrix for “DENSITY OF FAILURE”

As previously mentioned, this is the results table of the Neural Network analysis, where the level of feasibility of the pattern found by the tool is expressed in percentages. *Figure 3.28 and Figure 3.29.*

It is observed that 99.7% of the time the pattern determined by the Neural Network gave a result equal to the expected result, and only 0.3% of the time was wrong. This seems in principle a very good result, but as has been seen previously this "perfection" of the result does not always mean that the result is so "perfect".

Therefore the result should be analyzed in a deeper way, analyzing the rest of the boxes that we find in the confusion matrix to be able to determine if the result is feasible or not.

The first column represents in this case the time constants whose result has been taken as being in a state of correct behavior, that is, at the green level. It can be seen that all green-level detections have actually been green comparing them to the result it would be expected.

The second column represents the yellow level, where it is observed that 87% of the times the pattern established the result correctly, however, around 6.5% of the time had been detected as yellow level situations some situations that have turned out to be green level, and another 6.5% of the time had been detected as yellow level situations some, that when compared with the expected values, have turned out to be red level.

Finally, the third column corresponds to the red level. Comparing with the expected value, only 0.2% of the time the pattern has failed, that is, the pattern has practically placed correctly all the instants that were expected to be at the red level.

Therefore seeing the obtained results it is seen that critical parameter when considering as feasible or not the pattern developed by the tool, is the one of instants detection in yellow level. In which 87% made it correctly, and 13% failed. It is important to note that the pattern has not failed to detect any instant yellow level, but rather the errors are due to the fact that it has given yellow level value to instants that theoretically were not.

In addition, the yellow level, since it is not the critical process in terms of functionality of the actual process (this would be red, which means total or almost total malfunctioning), is assumed as acceptable the level of success of the pattern found by the Neural Network tool.

In view of the analysis carried out, and of the results obtained, it is possible to conclude that there is a CAUSE-EFFECT relationship between the data measured in the system and the results obtained by levels of deterioration depending on the number of failures detected in previous attempts.

The objective of this section was to find a pattern in some way that would help to predict or prognostic the failures in the given system for the study and with it to facilitate the implementation of an E-maintenance system, of predictive maintenance. During this section, several hypotheses have been considered, and after the meticulous analysis of the data, it has been possible to determine the existence of a pattern and its feasibility has been demonstrated through the use of the MATLAB® tool, called Neural Networks.

A simple way of implementing an interface is further on described, in order to be able to effectively control the proper functioning of the system in real time.

In a further section (SEE SECTION 4) will be considered how to approach the results obtained here from the point of view of business, that is, strategies and methods will be proposed to make a good use of these results. These topics are left for later as the present point does not have this as a goal.

3.3. PLATFORM AND INTERFACE (PRACTICAL CASE)

Matlab allows an easy developing set of screens (panels) with buttons, menus, windows, etc., that allow to use in a very simple way realized programs within this environment. This set of tools is called the graphical interface of user (GUI). The possibilities offered by MATLAB ® are not very extensive, compared to other Windows applications such as Visual Basic, Visual C. etc.

Developing GUIs can be carried out in two ways, the first is to write a program that generates the GUI (script), and the second option is to use the GUI design tool, included in the MATLAB ®, called GUIDE. This section will address both forms for the creation of the interface, but without entering in specific description, step by step GUI development, since the idea of this work is not to present how an interface is developed but rather to show the uses and benefits that can lead to its use.

Introducing GUIDE tool briefly, it has as stated in the previous paragraph two working options, the design part and the part where it works directly by writing a script in matlab.

The design part is where it is entered all the objects that will serve for the communication between the user and the internal program. These are objects through which the user can introduce inputs of any type in the system, and through which it can receive outputs.
It consists of a series of buttons, text boxes, figures, etc., that are dragged from a predetermined library by matlab to the working panel for later being edited by the designer.

The script part is where all the actions that are wanted to be done by the platform are defined by means of MATLAB® code, such as opening or closing the interface, or entering any type of input through the designed objects.

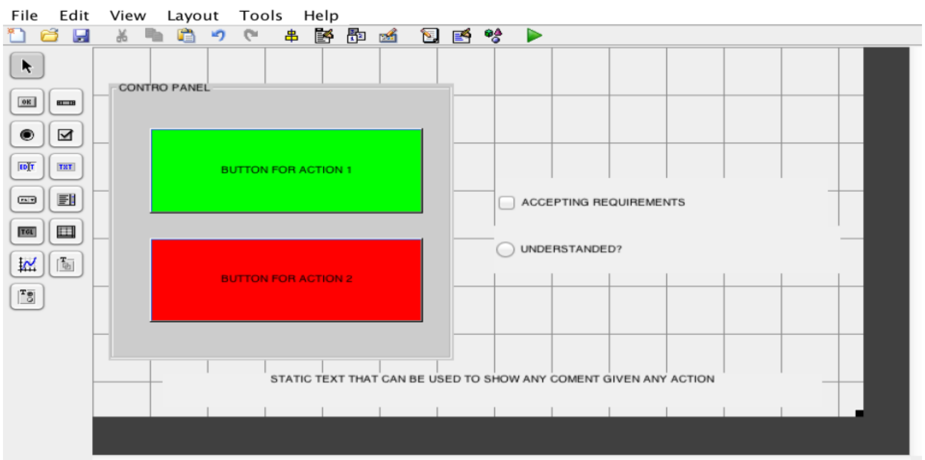


Figure 3.30. GUI Design Window

GUI tool, *Figure 3.30,* allows creating a fast system of communication between the user and the system or process that is wanted to be monitored. For this work, it is a question of developing the tool so that the user through the introduction of a deterministic parameter receives, in real time, information on the situation in which the system is to be controlled.

As it was seen in section 3.2 of this paper, it was analyzed that one of the simplest parameters to measure were the failures detected in the system, and at the same time it was demonstrated that this parameter was the most easily usable to be able to provide with clarity and reliability a conclusion about the situation in which the system is, with regard to the working conditions of each IDbox.

Therefore, this interface it is made as simple and intuitive as possible, taking as input values only if a failure is detected for each time instant, besides specifying exactly form which IDbox data is being took at each moment.

The platform will consist of aligning the data introduced from the previously exposed idea of *Density of Failure,* where for each different IDbox will be analyzed every time it is entered data, in which situation is, and it will be automatically transmitted to the user of the interface. *Figure 3.31.*

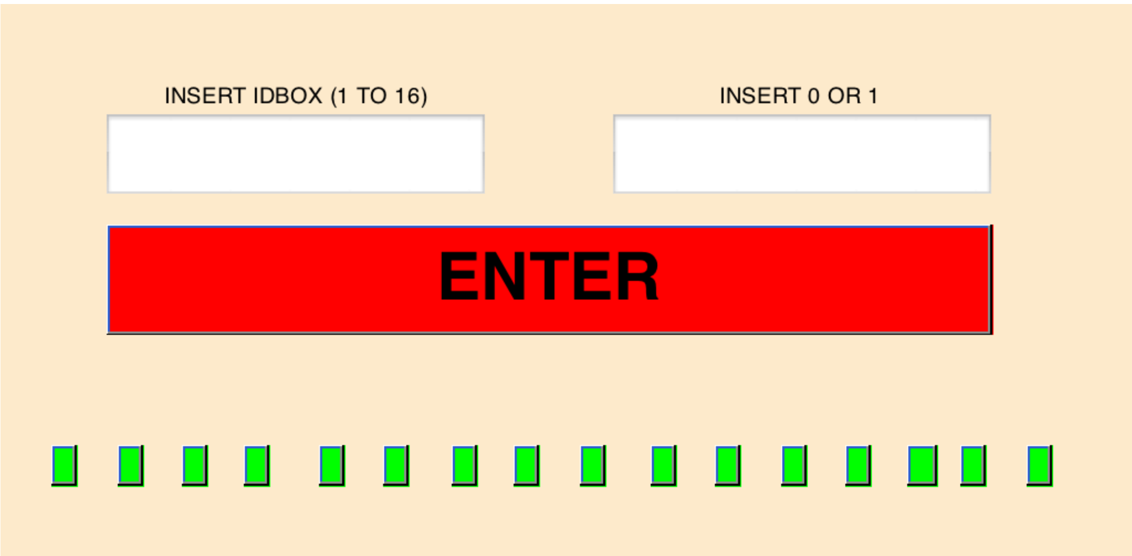


Figure 3.31. Interface

To understand the operation of this platform it is very important to take into account how the different levels of operation in each IDbox are determined. There are three possible levels (section 3.2), which are green when operating under normal conditions, yellow level when in a situation where the system could lead to failure, and finally red level where the system is considered malfunctioning.

The platform works with a database where the obtained values for failure detection are stored, that is to say 0s or 1s, for each of the 16 IDbox (it could be seen as a binary matrix where each row corresponds to a different IDbox, and each column at a different instant of time or vice versa).

In the database the only data stored is the one since the last moment of maintenance, since this system is not intended to store all the measured historical data but what is intended is simply to provide in a fast and reliable way the working conditions of each IDbox, and for this it is only needed the latest fault detection data from a repair (see section 3.2). Therefore every time an IDbox is repaired (ideally it should be repaired before failure) the part of the database corresponding to such IDbox will be initialized, being deleted the fault data detected until that moment.

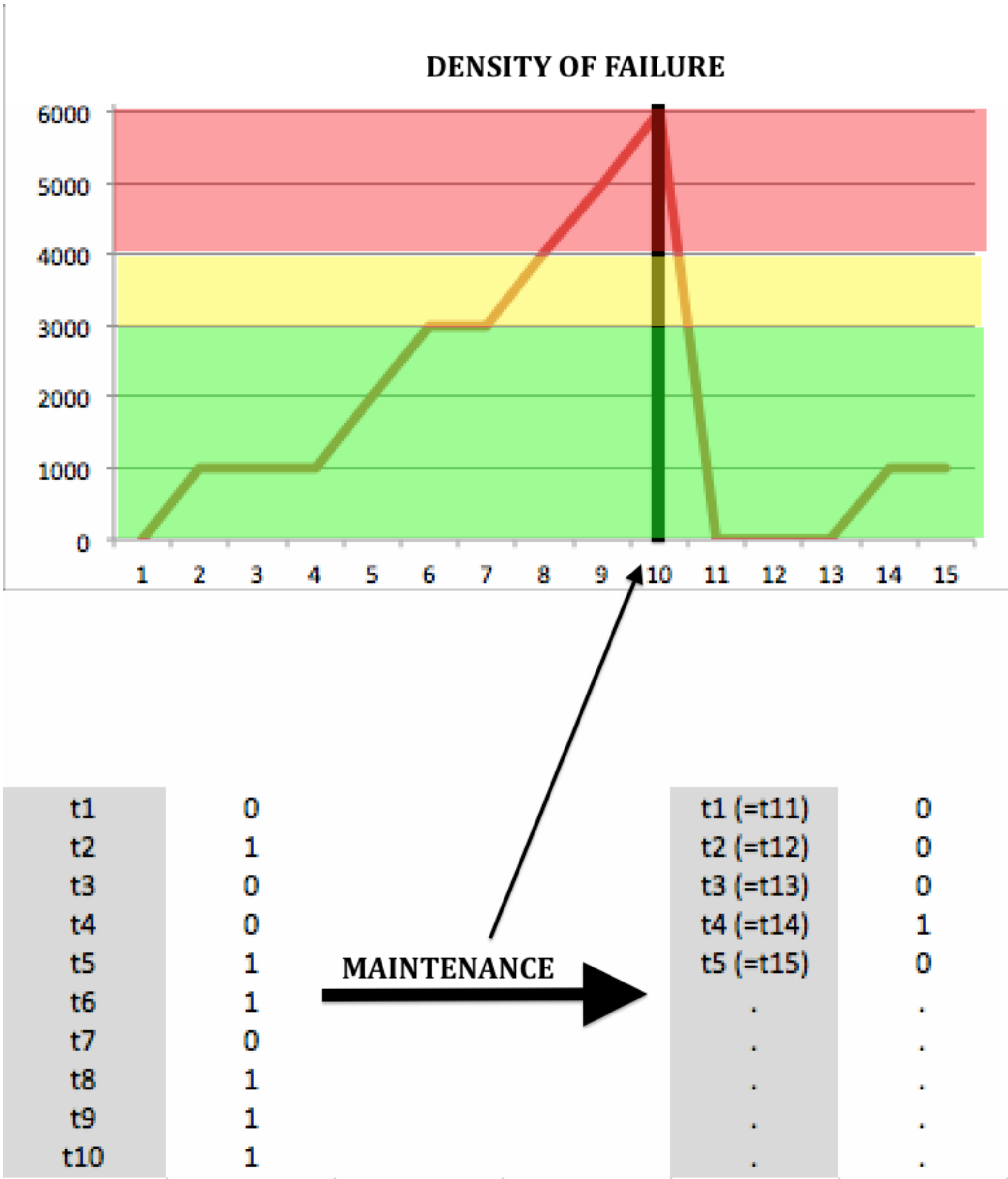


Figure 3.32. Database Working Mode

The user through the interface will enter for each instant the data corresponding to whether or not a system failure has been detected for a particular IDbox, and these data will be stored in the database of the platform, which will read the data stored for each IDbox and will always show in real time the level of deterioration of each IDbox, using the color system represented in the interface. *Figure 3.32.*

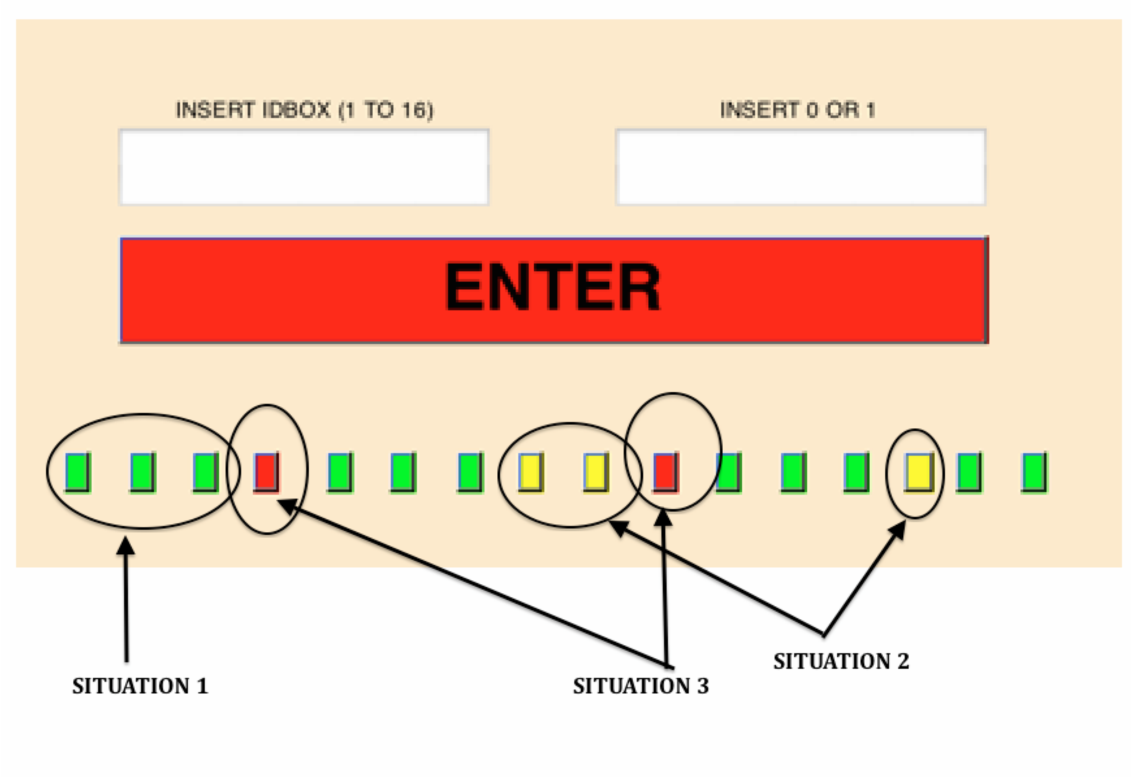


Figure 3.33. Alarm Levels in Interface

- **SITUATION 1:** IDBOX 1, IDBOX 2, IDBOX 3… are in correct working conditions, no maintenance have to be done.
- **SITUATION 2:** IDBOX 8, IDBOX 9, IDBOX 14, are in yellow level, not failed but special attention has to be made because possibly soon it will fail.
- **SITUATION 3:** IDBOX 4 and IDBOX 10 are failed, must be made a maintenance action as soon as possible to minimize time IDBOX is not working, or very badly working.

In the case that the user entered an IDbox code lower than 1 or higher than 16, the interface would display an error message, *(see Figure 3.34.),* and the values entered, as expected, would not be introduced into the database. In the same way it would happen in the case of entering a value other than 0 or 1 in the text box established to enter whether or not a failure has been detected.

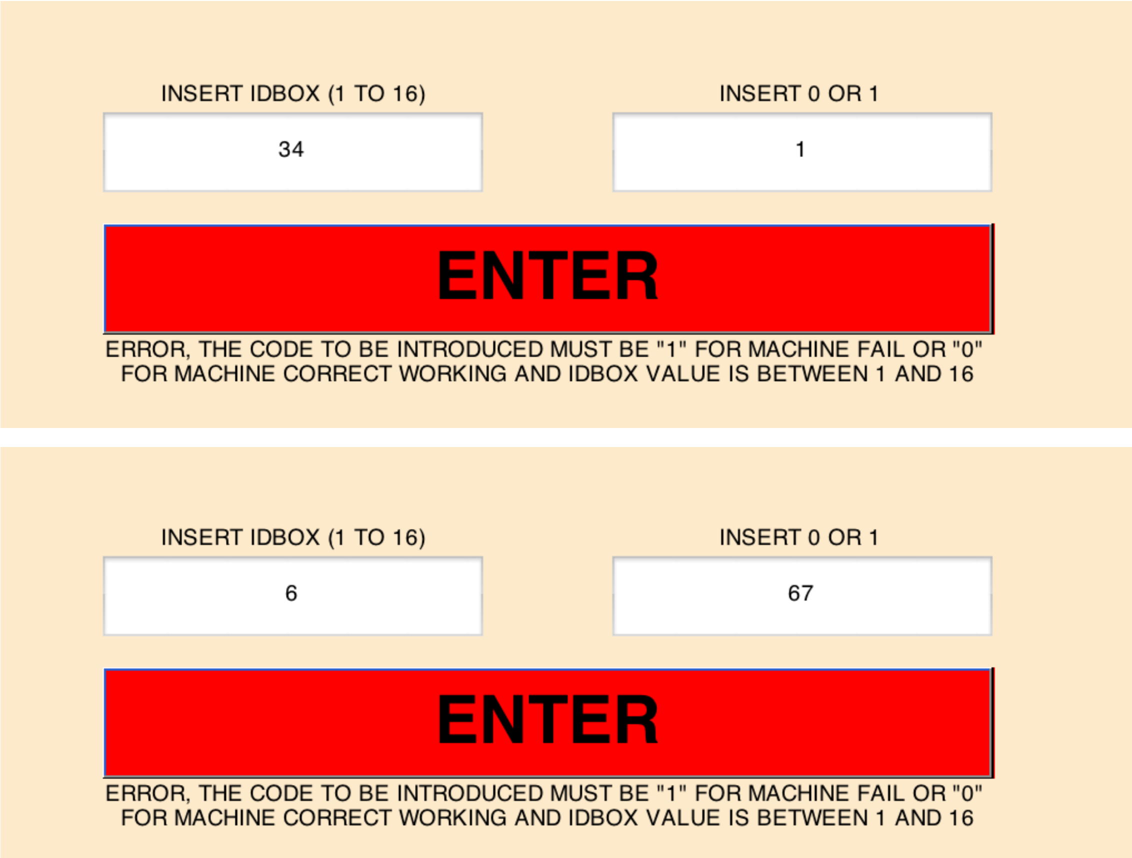


Figure 3.34. Error Message

Matlab allows creating a simple interface like the one shown in this section, which allows the user to keep a real-time control of the IDboxs that must be monitored.

In addition are shown the working conditions of each independent system in a very intuitive way, which allows to predict future unwanted situations and therefore to address the problem through maintenance actions.

CHAPTER 4. MAINTENANCE SCHEDULING

Once the data provided by the company has been analyzed and a pattern has been determined that helps prevent the system failure by monitoring the deterioration of the machines in real time, it is desired to present a different theme than the ones exposed until this moment but from the point of view of maintenance equally important to be able to develop a complete and useful E-maintenance system.

It be will briefly analyzed different possible strategies for scheduling the maintenance by the company to benefit from the results obtained from the analysis developed in section three of this monograph.

As it has been said, it will be analyzed briefly since for their complete and correct development it would be necessary more data about the process. Beyond the measurements of different parameters on the machines and the detection of failures in the system, it would be necessary another type of parameters like personnel costs, costs of production losses by time-outs, repair times for each IDbox, etc.

It is considered important to present these maintenance scheduling methods since in order to obtain economic and productive benefits from the results obtained from the analysis of the data in order to prevent system failures, it is necessary to carry out correct strategies of prognostic maintenance.

First of all the Maintenance Scheduling will be defined as:

"Procedures for the temporary assignment of the maintenance orders generated by the planning with the aim of optimizing the use of resources"

Maintenance scheduling allows the company to prepare all material or labor resources to be able to attack the problem due to the progressive deterioration of the machines minimizing the useful times in which the machine is not in operation.

In order to analyze this particular work case (FRIGUS case), a deeper analysis of the failure detection statistic must first be done.

In the data provided by the company have between 10000 and 11000 instants of measurements for each different IDbox spread over a period of one year (season 2014/2015).

Therefore, to simplify, making a uniform distribution corresponds to about 40 instants of measures taken a day (taking into account that in a natural year are 250 working days).

Noting also that the percentage of instants in which a fault has been detected (represented by red dots in the graphs) corresponds to approximately 1% of the instants, if the whole system had a uniform distribution throughout the period, it will statistically correspond to the occurrence of failure detection once every two and a half days of work.

Another consideration that should be taken into account in carrying out this analysis is the definition of green, yellow and red deterioration level. Green level means that the system is working correctly, whereas red level means that although in principle it works, the system is drifting in a total failure.

Meanwhile, the yellow level is an intermediate level of alarm, which although it corresponds to a good functioning works as an an alarm in the system since in most cases will end up reaching the red level, therefore the yellow level would not determine in principle the need to carry out actions and maintenance however if the system is put on alert for possible soon malfunction of the machine.

The yellow alarm level is equivalent to having detected 15 faults in the last 1000 instants, and the red level to have detected 20 faults in the last 1000 instants, therefore to go from yellow level to red level would require 5 more faults detections, and this statistically corresponds to 12.5 working days (2.5 * 5).

This means that statistically 12.5 days after the yellow level has been detected it will go to the red level. Properly the maintenance should be carried out at this moment before passing to red level. For security, the policy that should be taken is 10 days instead of 12.5 to perform maintenance actions on an IDbox from the time a yellow level has been detected.

The yellow level does not always mean that it is going to drift to a red level, but in most of its cases, therefore the 10 working day period is left in case the system returns to correct operating conditions with an index less than 15 faults detected in the last 1000 instants, which would correspond to a green level.

By the time the yellow level is reached, the machine should not be placed in a sort of waiting list for maintenance, but rather in a state of exhaustive control where the resources are prepared to carry out the maintenance in case it derives at red level, or in case it exceeds 10 working days with the yellow level alarm activated.

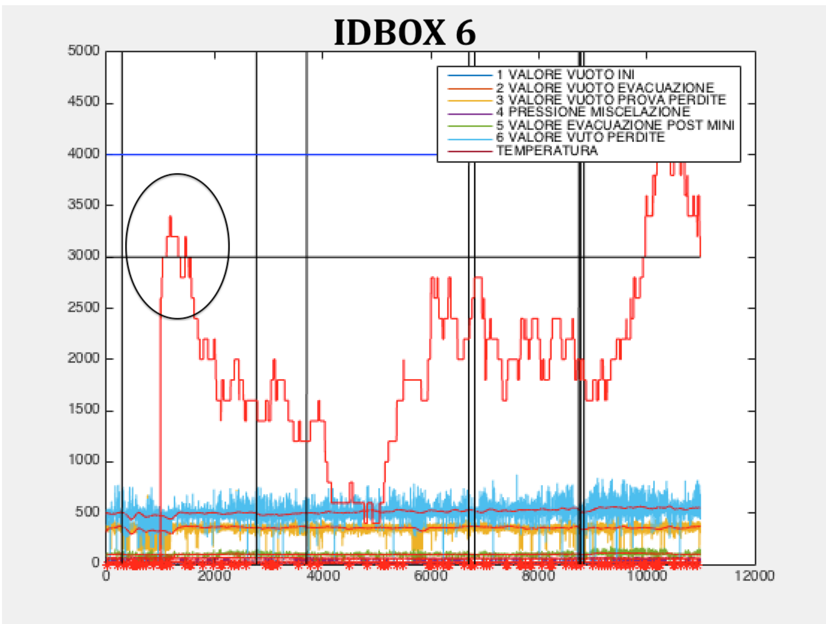


Figure 4.1. IDbox 6

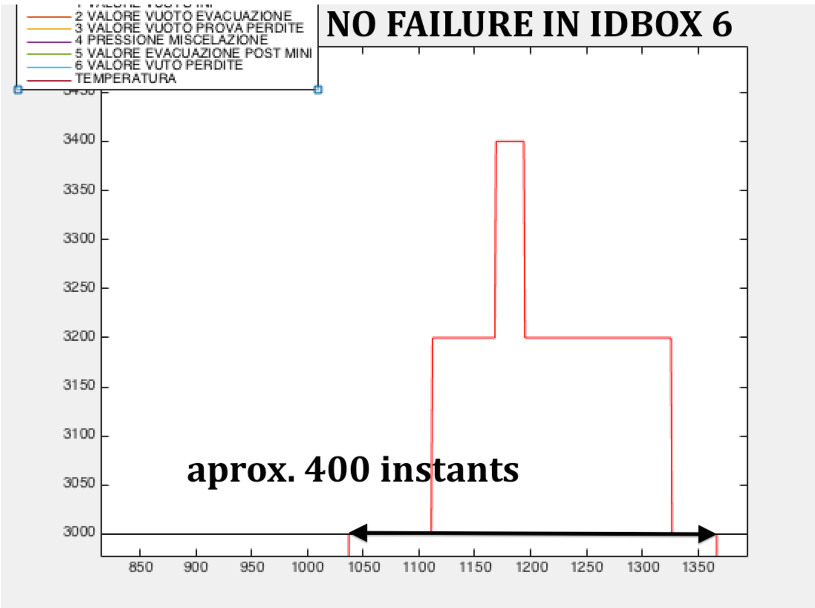


Figure 4.2. Not Failure Yellow Level

In the last graph *(Figure 4.2.),* it can be seen how sometimes the system is in yellow level, but it does not end up in red level conditions, in the other hand it turns back to the ideal green level situation. In the other hand, when yellow level it has been active for a big amount of time means that the system is not working as it should work ideally and it has a high probability of deriving into a red level situation, so maintenance actions have to be carried on.

Viewing the previous analysis it will be defined a *maintenance list* where IDboxs that should receive maintenance actions paying attention to those IDboxs that are in yellow level, but have not exceeded the 10 working days with constant yellow level Activated, because even so, at any time could drift at the red level and would need to be maintained.

To do this, two methods are proposed for maintenance scheduling, *Figure 4.3*:

- ***FIFO Rule***

- ***SOT Rule***

Where the application of one or other of these two rules would depend on the parameters available to be able to perform the analysis and optimize resources through the application of these rules.

FIFO stands for ***First In First Out***, is a criterion according to the idea in which the IDbox that first enters the waiting list for maintenance (that is to say those that in yellow level goes over the 10 days, or those in red level), will be the first IDbox to receive maintenance actions. This system is useful for situations in which are not known the repairing times for each IDbox, or in the case where the repair time for each IDbox is the same. This method allows the company to carry out a simple but effective control when setting the priority for a correct production schedule.

SOT stands for ***Shortest Operation Time***, is a useful maintenance criterion for those situations in which it must be carried out maintenance to more than one IDbox. It consists of giving priority to that IDbox whose time of repair is lower; therefore it is a rule of maintenance scheduling for which it is necessary to have specified the repair time of each IDbox to be able to apply it.

It is based on the idea that if maintenance work is carried out quickly, maintenance actions may be carried out earlier in other IDboxs that need it. This method optimizes the average waiting time of an IDbox to be treated with maintenance actions, thus obtaining a lower average time of IDboxs without working, and therefore a better resource optimization as the time having reduced the average time-out.

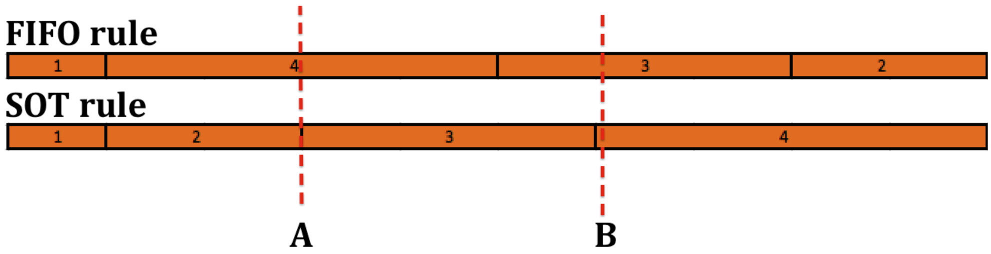


Figure 4.3. FIFO vs. SOT

In the last figure it can be seen a comparison between both rules and it can be clearly seen why SOT optimizes resources better. At point A using the FIFO rule it can be seen that only one IDBOX has been repaired, meaning that will be in correct operation conditions again, being three IDboxs until in the waiting list for maintenance, however with the same conditions, using the SOT rule at point A it will be had two IDboxs repaired, and two in the waiting list, therefore the productivity of the system is better, the same can be observed in point B, where when using the FIFO rule at this instant two IDboxs will be repaired and two yet waiting in the list for maintenance while if applying the SOT rule for this instant B three IDboxs will be already repaired and only one in the waiting list.

Therefore, through the graphical representation, it is easy to see how the application of this second rule will reward greater benefits to the company. Even though the total time needed for maintenance is the same (this means that the time operators that do maintenance are working is the same in both cases), the productivity will be higher with SOT's rule due to the fact that the time a IDbox remains waiting in the list fir maintenance is lower.

When time for maintenance can be determined for each of the IDboxs, for an efficient maintenance scheduling and therefore for a correct E-maintenance, the SOT's rule must be applied in the system.

CHAPTER 5. CONCLUSIONS AND RESULTS

The productivity of an industry will increase as the amount of faults decrease in a subsistent manner over time. To achieve this, it is important to apply appropriate maintenance strategies and have trained personnel in the use of fault forecasting techniques.

It has been demonstrated in this work how by analyzing the data produced in a system it is possible to find patterns to determine the behavior of a machine, in addition it has been presented different techniques and tools to carry out the research. It has been shown how through this knowledge are applicable strategies of predictive maintenance through which it achieves a higher productivity of the company or greater reliability and feasibility of the processes and systems of the organization.

The aim of this work was to show the importance and benefits of implementing advanced maintenance systems through the use of the available technology. Trying to focus the control of maintenance beyond the repair of failures, as a system towards it can be innovated and towards which companies should be directed.

5.1 RESULTS

The objective of the implementation of an E-maintenance based system is, through maintenance strategies such as CBM (Condition-Based-Maintenance), to provide a continuous study and analysis of all types of data produced in manufacturing processes to know more accurately the working conditions of a system, process or machine. In this way, it is possible to predict when a fault can occur, thus achieving to maximize the useful life of the components, optimizing in this way the available resources.

At the outset, several important concepts were presented (see section 1) for the understanding of this monographic work. It was presented the importance that must have for a company the idea of continuous quality improvement in all aspects of the organization, focusing on improving critical processes. In addition the term of Big Data was introduced, showing how to deal with a correct data treatment in order to implement a maintenance system based on the monitoring of conditions, such as E-maintenance.

Subsequently the main objectives of this work were defined (see section 2), starting from the need to work with correctly defined data, that is to say, with the smallest possible number of erroneous data, to be able to proceed to its correct analysis. Emphasizing the importance of performing a good data processing to obtain reliable results. Subsequently, techniques and methods were presented for the analysis of data, highlighting the presentation of the Artificial Neural Networks (see section 2.2), in order to provide the necessary knowledge to perform a fruitful study for finding patterns of behavior in large volumes of data.

Successively in section 3 a practical case was analyzed for a specific company where all the knowledge and concepts presented in the theory of previous sections was applied. A thorough processing of data was carried out where once, after its processing and filtering, it was passed to its analysis through pattern recognition techniques such as the so-called Neural Network (using a tool in MATLAB®), resulting in the determination of a useful pattern for the prevention of the failure which was called "DENSITY OF FAILURE" (see section 3.2). Subsequently a simple platform was developed by which the monitoring of the conditions of the system studied through the determined pattern was made visible.

Finally, it was presented the concept of maintenance scheduling where the feasibility of implementing a plan of scheduling maintenance actions as a way to complete an E-maintenance system and to be able to obtain the maximum optimization of the resources.

5.2 FUTURE DEVELOPMENTS

The most feasible direction to develop this work in the future would be to expand the work area, beyond focusing only on a certain type of machines and working with the data measured on these machines, i.e. working with larger volumes of data produced in all types of systems and processes within an organization, to try to obtain results for the improvement not only of a particular process but of everything possible, to achieve a high degree of quality improvement within the company.

In addition to correctly implementing an E-maintenance system, it would be useful to have datum relevant to the costs, costs of machine downtimes, labor costs, maintenance costs, as well as other parameters such as times required for maintenance, etc., necessary to be able to develop systems of intelligent maintenance in a company much more precise and complete, always having as main objective the improvement of quality through the reduction of the uncertainty, with special attention to the uncertainty caused by the failures in the processes.

To complete the implementation of an intelligent maintenance system, in addition to determining plans for maintenance scheduling, and to discover patterns through which the working conditions of machines can be determined in order to predict a possible failure, study and analysis of the real viability, that is to say of the costs that would suppose, and therefore if it would be feasible or not, the development of a platform that allows to connect all the machines of the company with all the workers and all the departments, the so-called, "Internet of things", to achieve a continuous and precise monitoring of the situation in which not only each processes are found, but also the capacity of work available in each one.

"Quality is never an accident, it is always a result of an effort of intelligence."

John Ruskin.

CHAPTER 6. BIBLIOGRAPHY

1. Asplund, M., Famurewa, S. and Rantatalo, M. (2014). Condition monitoring and e-maintenance solution of railway wheels. *Journal of Quality in Maintenance Engineering*, 20(3), pp.216-232.

2. Ministerio de Fomento,. (2005). Mejora Continua en la gestión del transporte - *La Gestión por Procesos* (http://www.fomento.es/NR/rdonlyres/9541ACDE-55BF-4F01-B8FA-03269D1ED94D/19421/CaptuloIVPrincipiosdelagestindelaCalidad.pdf). Madrid: Ministerio de Fomento.

3. Gonzalez, H. (2012). *LA MEJORA CONTINUA – DIAGRAMA DE PARETO*. *Calidad y Gestion*. Retrieved 11 January 2017, from https://calidadgestion.wordpress.com/2012/09/11/mejora_continua-diagrama_de_pareto/

4. Gonzalez, H. (2012). *HERRAMIENTAS PARA LA MEJORA CONTINUA*. *Calidad y Gestion*. Retrieved 11 January 2017, from https://calidadgestion.wordpress.com/2012/07/11/herramientas-para-la-mejora-continua/

5. Herminia D, Tavera L. "Lesson 5 - Management by Processes". Dirección de Empresa. Unviversidad Politécnica de Valencia. 21th November 2016.

6. Herminia D, Tavera L. "Lesson 6 - Six Sigma Methodology". Dirección de Empresa. Unviversidad Politécnica de Valencia. 6th December 2016.

7. Schroeck, M., & Shockley, R. (2012). *Analytics: el uso de big data en el mundo real Cómo las empresas más innovadoras extraen valor de datos inciertos* (1st ed., pp. 3 - 22). IBM Institute for Business Value. Retrieved 14 February 2017 from http://www-05.ibm.com/services/es/gbs/consulting/pdf/El_uso_de_Big_Data_en_el_mundo_real.pdf

8. Rüßmann, M., Lorenz, M., Gerbert, P., Waldner, M., Justus, J., Engel, P., & Harnisch, M. (2017). *Industry 4.0: The Future of Productivity and Growth in Manufacturing Industries*. *www.bcgperspectives.com*. Retrieved 9 February 2017, from https://www.bcgperspectives.com/content/articles/engineered_products_project_business_industry_40_future_productivity_growth_manufacturing_industries/

9. Lee, J., Ni, J., Djurdjanovic, D., Qiu, H., & Liao, H. (2006). Intelligent prognostics tools and e-maintenance. In B. Elsevier, *Computers in Industry* (1st ed., pp. 476–489). H. Wortmann, H.S. Jagdev. Retrieved 9 February 2017, from https://erc.engin.umich.edu/wp-content/uploads/sites/50/2013/08/Intelligentprognosticstoolsande-maintenance.pdf

10. Levrat, E., Iung, B., & Crespo Marquez, A. (2008). E-maintenance: review and conceptual framework. *Production Planning & Control*, *19*(4), 408-429. http://dx.doi.org/10.1080/09537280802062571

11. Holmberg, K. (2010). *E-maintenance* (1st ed., pp. 5 - 37). London: Springer.

12. Borissova, D., & Mustakerov, I. (2014). *A concept of intelligent e-maintenance decision making system* (1st ed.). Sofia. Retrieved 9 February 2017, from

https://www.researchgate.net/publication/259800598_A_concept_of_intelligent_e-maintenance_decision_making_system

13. Álvarez Tejedor, T., Singh, R., & Pilidis, P. (2013). Advanced gas turbine asset and performance management. In P. Jansohn, *Modern Gas Turbine Systems* (1st ed., pp. 515 - 564). P. Jansohn.

14. Recht JL. "*Introduction to Fault Tree Analysis"*. Safety in Automation. Università Politecnica di Milano. 2016

15. Schlosky A. "*Failure modes and effects analysis"*. Safety in Automation. Università Politecnica di Milano. 2016.

16. Ortiz Bravo, V., Ancizar Castañeda, J., & Nieto Arias, M. (2013). ANÁLISIS Y APLICACIÓN DEL FILTRO DE KALMAN A UNA SEÑAL CON RUIDO ALEATORIO. *Scientia Et Technica*, *18*(1), 267 - 275. Retrieved 13 February 2017, fromhttp://revistas.utp.edu.co/index.php/revistaciencia/article/viewFile/8241/5339

17. Cárdenas-Montes, M. (2015). *PREPROCESADO DE DATOS PARA MINERIA DE DATOS* (1st ed., pp. 1 - 5). Retrieved 9 February 2017, from http://wwwae.ciemat.es/~cardenas/docs/lessons/PreprocesadoDatos.pdf

18. Barranco Fragoso, R. (2017). *¿Qué es Big Data?*. *Ibm.com*. Retrieved 6 February 2017, from https://www.ibm.com/developerworks/ssa/local/im/que-es-big-data/

19. Cortázar Entrialgo, L. (2016). *Diferencias entre Big Data y Business Intelligence - Big Data News*. *Big Data News*. Retrieved 9 February 2017, from http://comunidad.iebschool.com/bigdatanews/2016/11/23/diferencias-que-debemos-saber-entre-big-data-y-business-intelligence/

20. Cinar A, Perk S. "*Process Monitoring"*. Safety in Automation. Università Politecnica di Milano. 2016

21. Duc M Le. "*Multivariate Analysis"* . Safety in Automation. Università Politecnica di Milano. 2016

22. Herrera-Rivera, J., Valdez, F., Romero, D., & Melin, P. (2012). *Optimization of Neural Networks for the Identification of Persons using Images of the Human Ear as a Biometric Measure* (1st ed., pp. 1-7). Retrieved from http://www.engineeringletters.com/issues_v20/issue_1/EL_20_1_12.pdf

23. Moraga, C., Allende, H., Salas, R., & Torres, R. (2004). Modular Neural Network Applied to Non-Stationary Time Series. In Reusch, *Computational Intelligence, Theory and Applications* (1st ed., pp. 585-598). Dortmund: Professor Dr. Bernd Reusch.

24. Katic, D. (2003). Neural Network Approach in Robotics. In D. Katic & M. Vukobratovic, *Intelligent Control of Robotic Systems* (1st ed., pp. 21-70).

25. Ordaz-Rivas, E. (2015). Control by Imitation Using a Maximum Sensibility Neural Network. In G. Sidorov & S. Galicia-Haro, *Advances in Artificial Intelligence and Soft Computing* (1st ed., pp. 306-315).

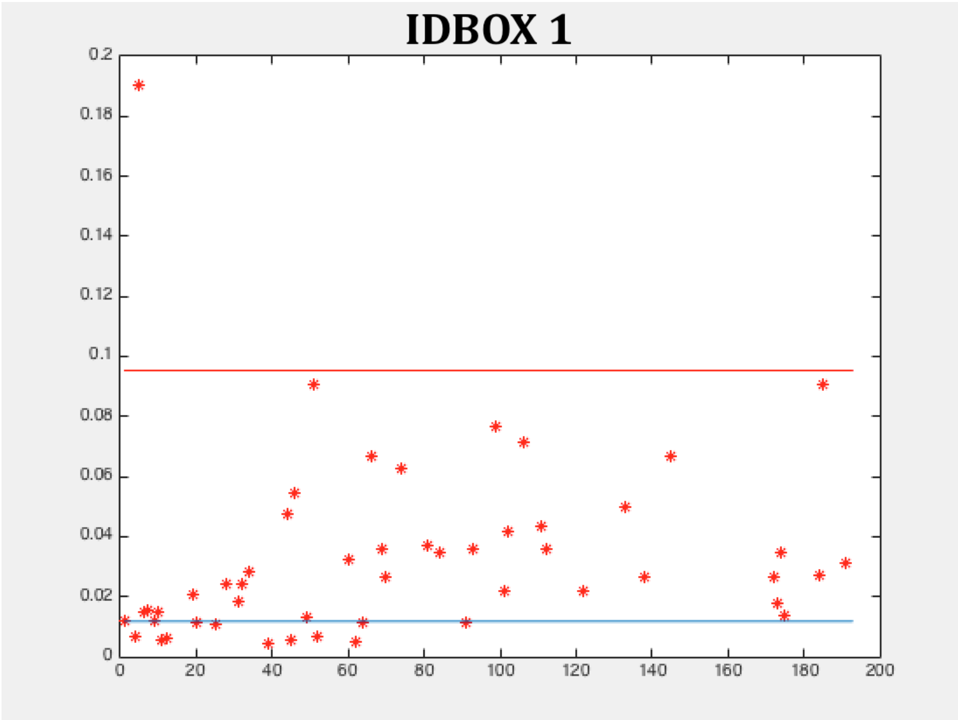
26. Gaxiola, F., Melin, P., & López, M. (2017). Modular Neural Networks for Person Recognition Using the Contour Segmentation of the Human Iris Biometric Measurement. In *Soft Computing for Recognition Based on Biometrics* (1st ed., pp. 137-153). Prof. Janusz Kacprzyk.

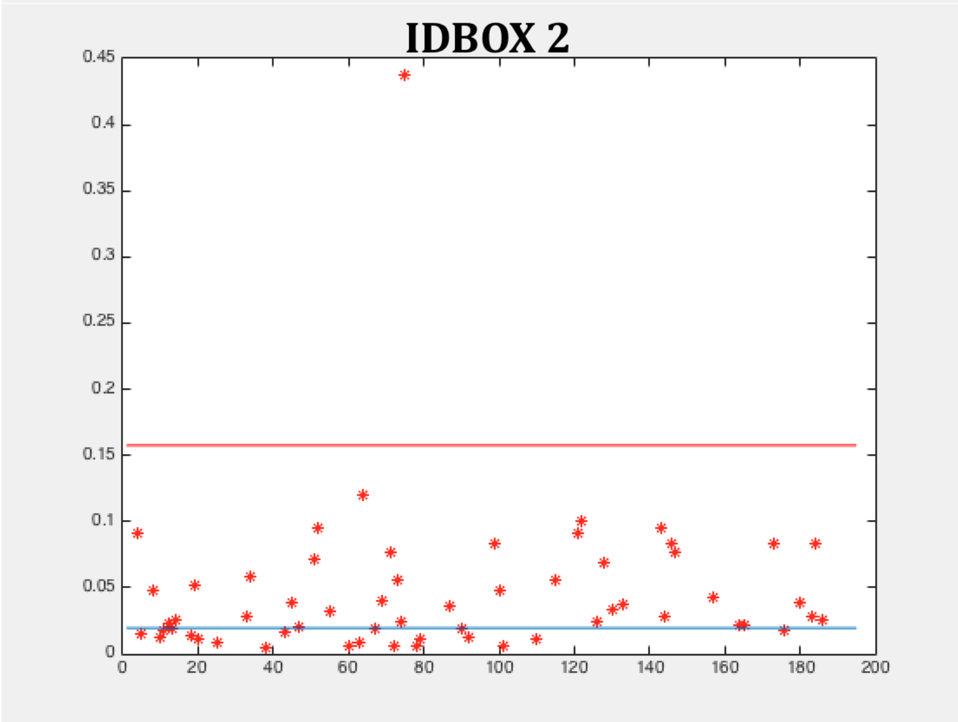
27. Bishop, C. (1995). *Natural networks for pattern recognition* (1st ed.). Oxford , Oxford Univ. (p. 191).

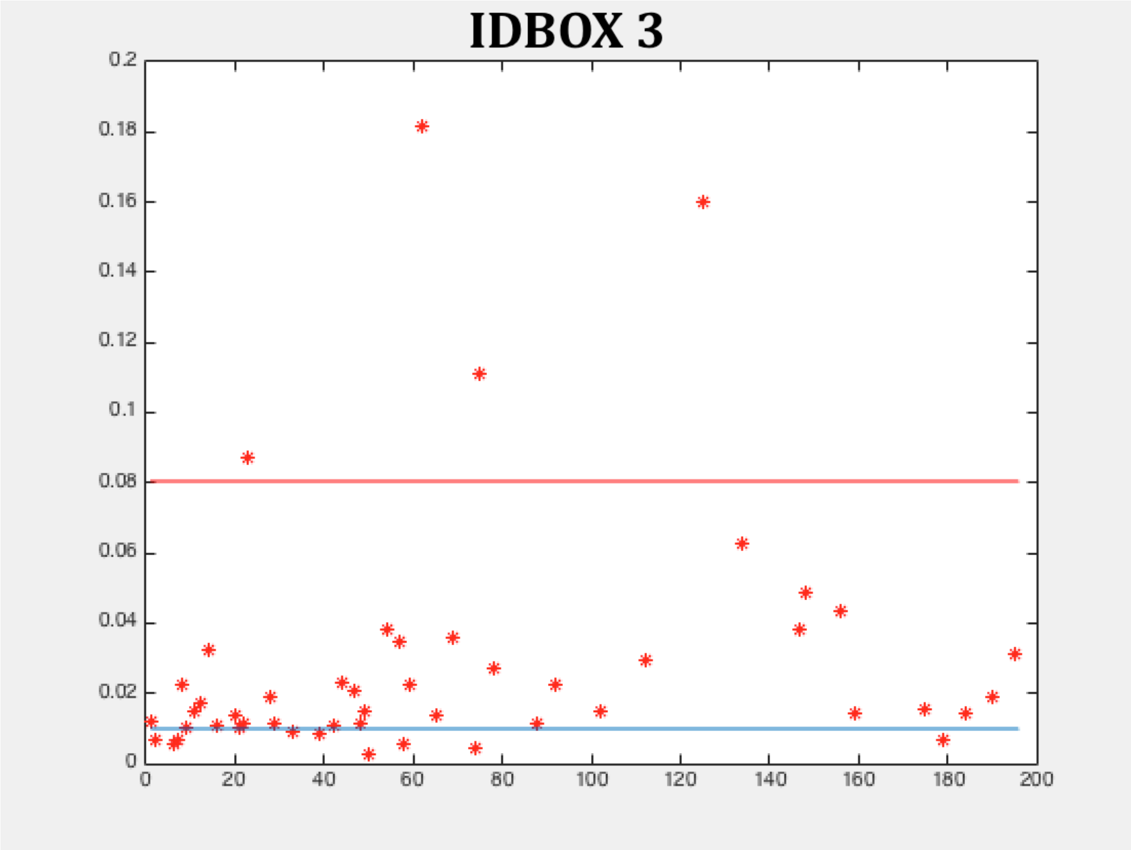
ANNEX

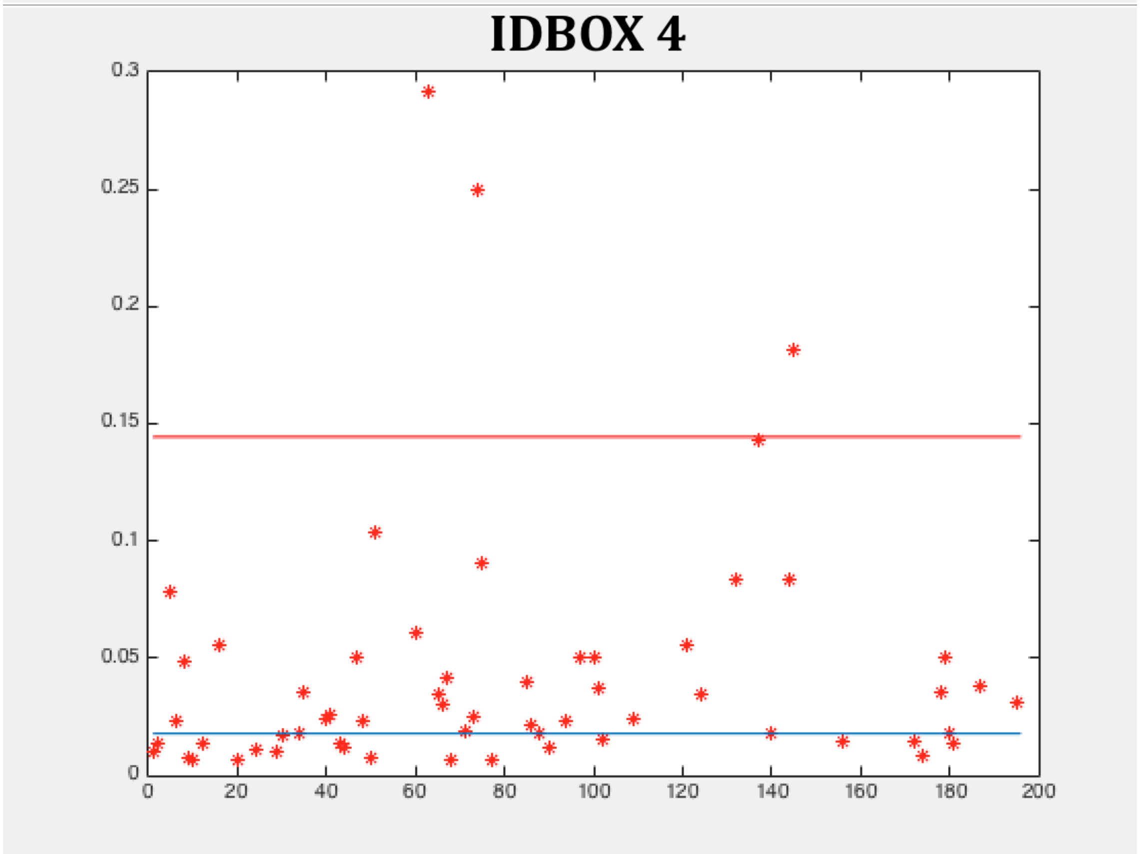
ANALYSIS OF DATA AND PATTERN RECOGNITION FOR THE IMPLEMENTATION OF AN E-MAINTENANCE SYSTEM

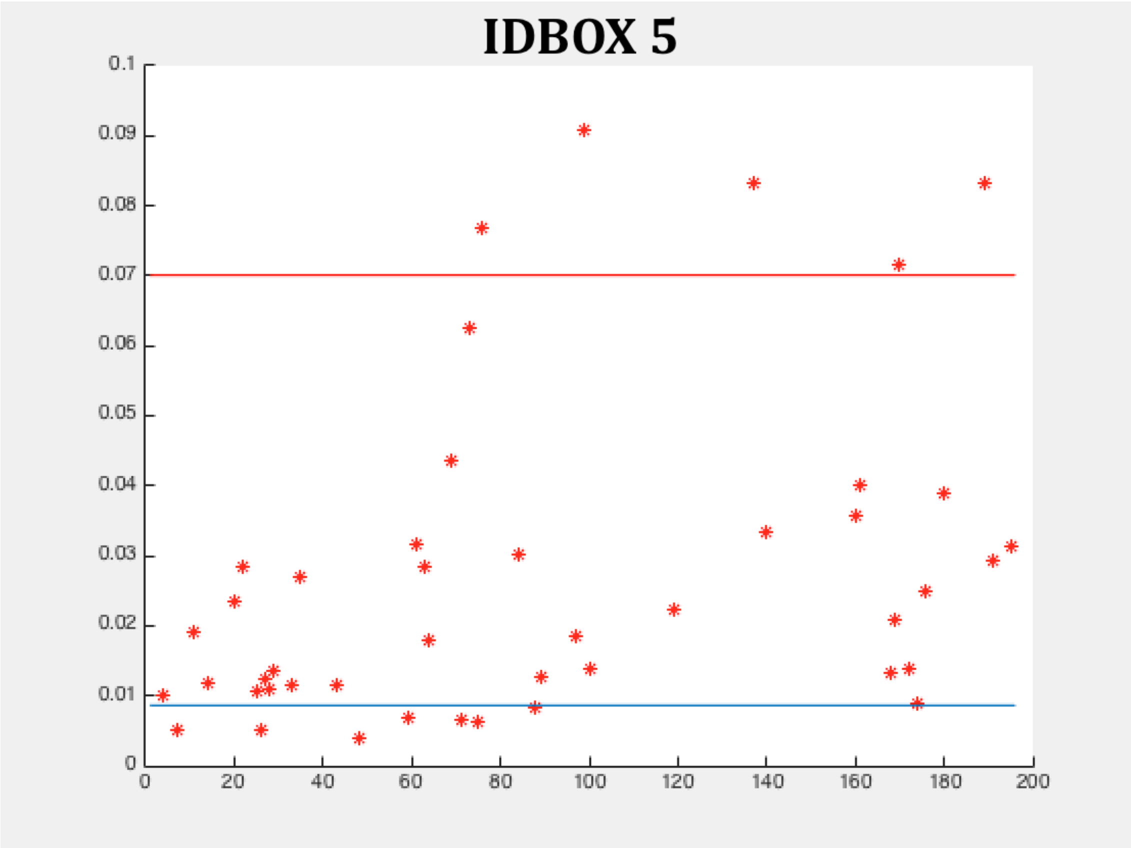
ANNEX I

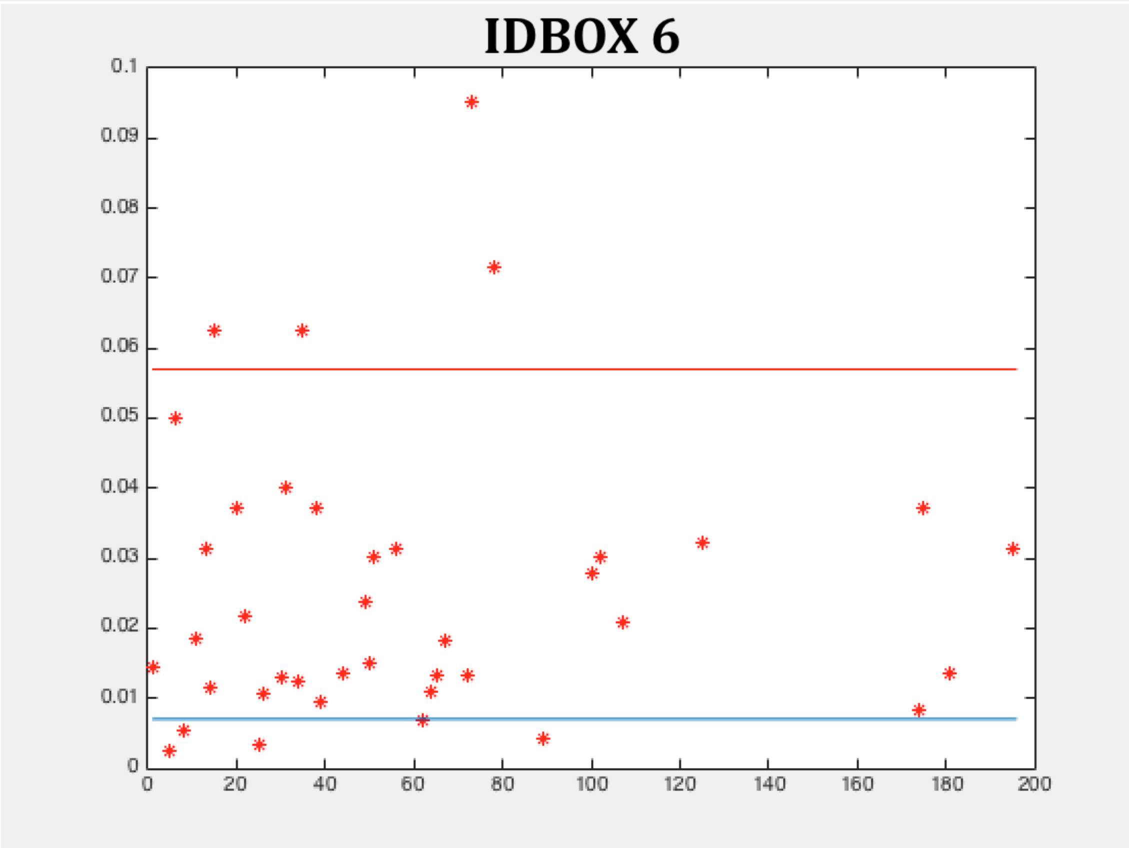

IDBOX 1
0.2
0.18
0.16
0.14
0.12
0.1
0.08
0.06
0.04
0.02
0
0
20
40
60
80
100
120
140
160
180
200


IDBOX 2
0.45
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
0
20
40
60
80
100
120
140
160
180
200

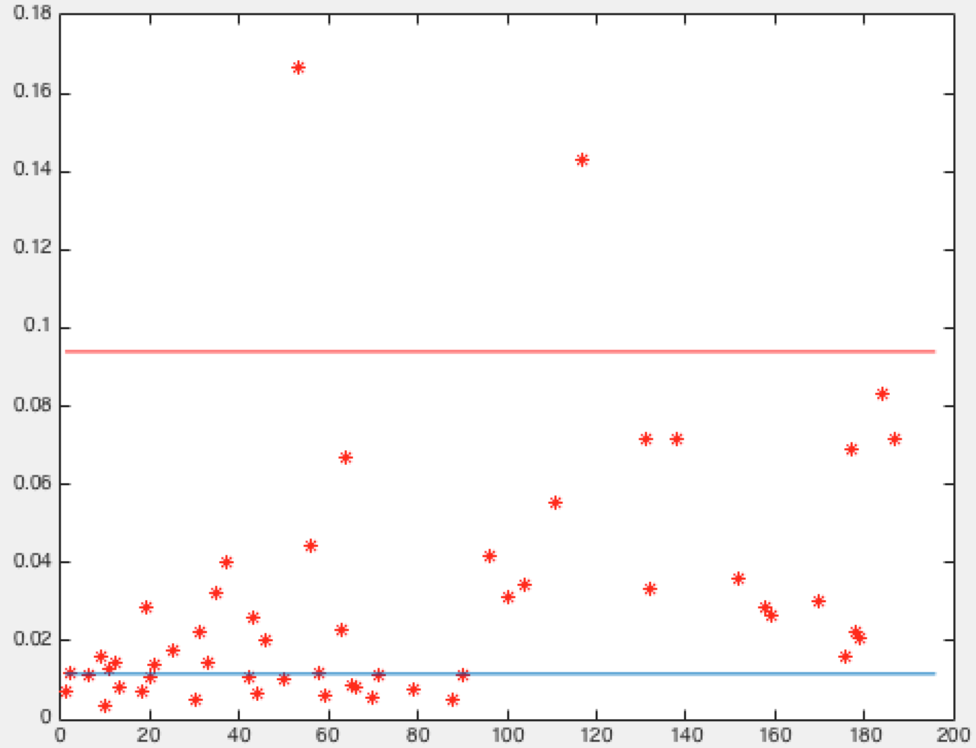

IDBOX 3
0.2
0.18
0.16
0.14
0.12
0.1
0.08
0.06
0.04
0.02
0
0
20
40
60
80
100
120
140
160
180
200


IDBOX 4
0.3
0.25
0.2
0.15
0.1
0.05
0
0
20
40
60
80
100
120
140
160
180
200

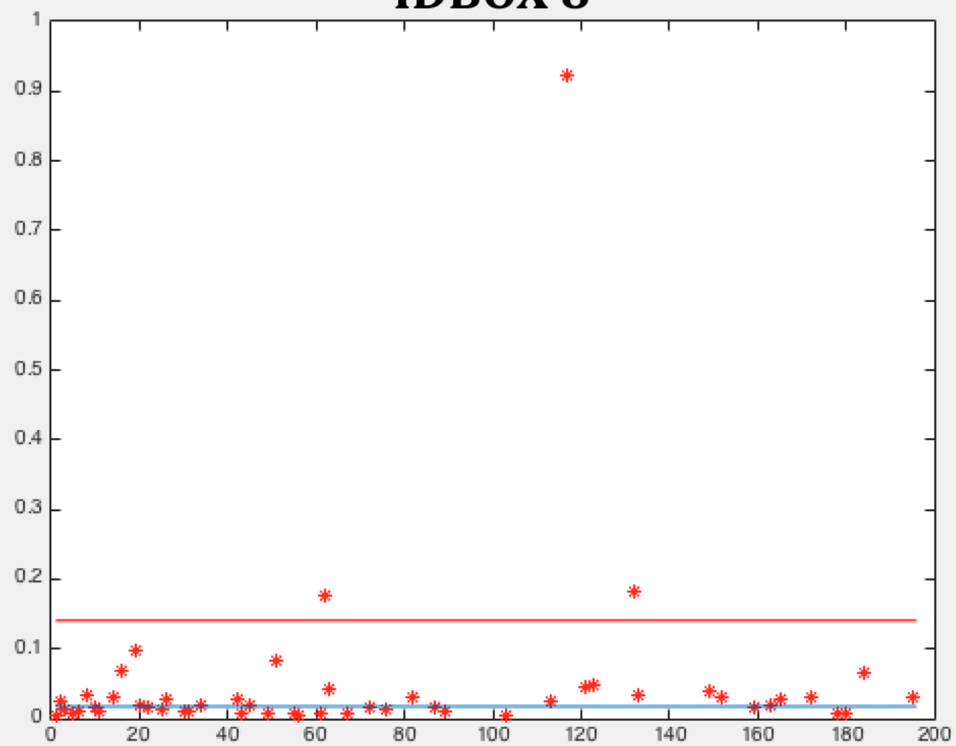

IDBOX 5
0.1
0.09
0.08
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0
0
20
40
60
80
100
120
140
160
180
200

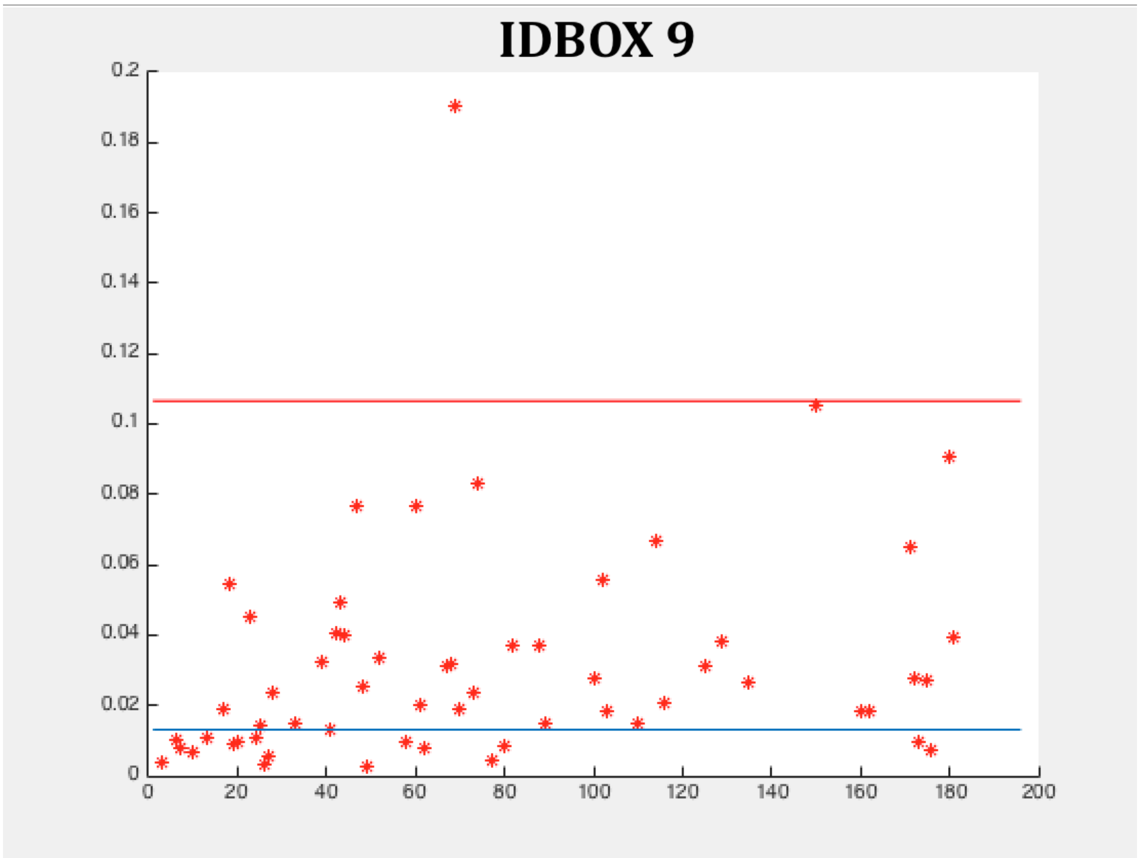

IDBOX 6
0.1
0.09
0.08
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0
0
20
40
60
80
100
120
140
160
180
200

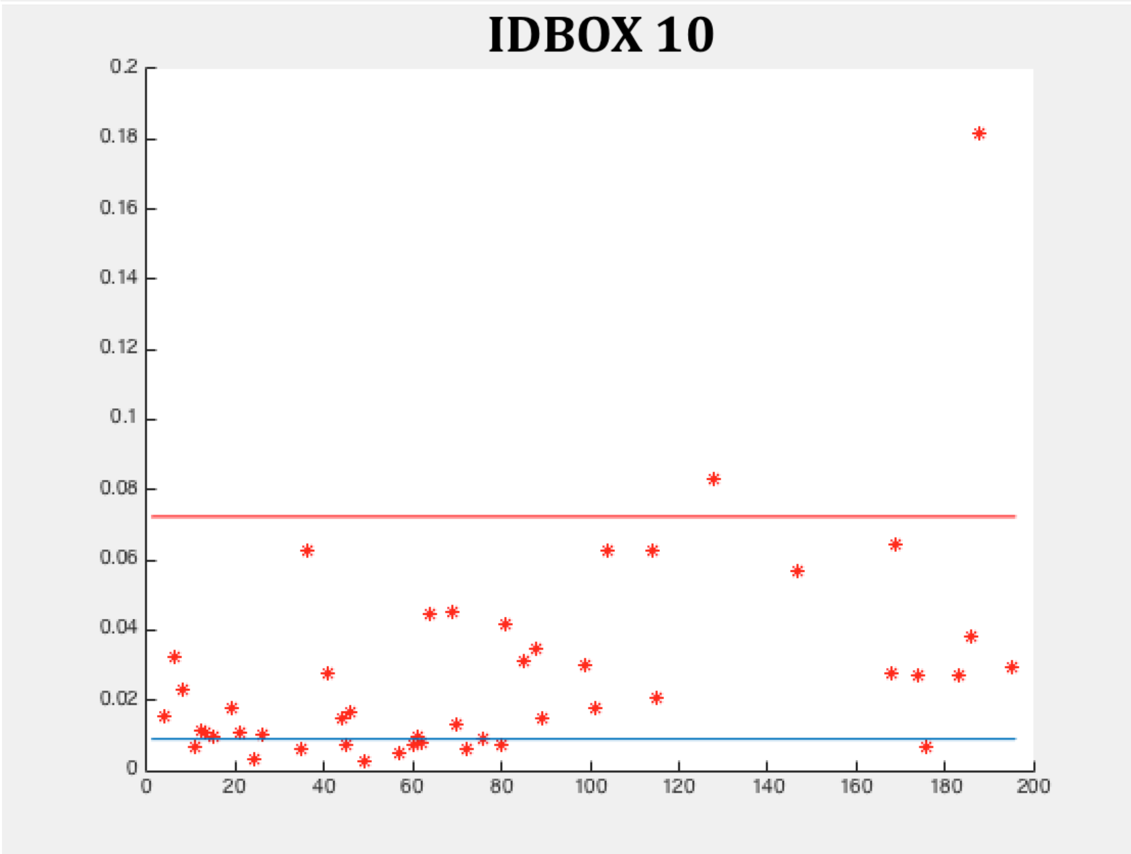
IDBOX 7

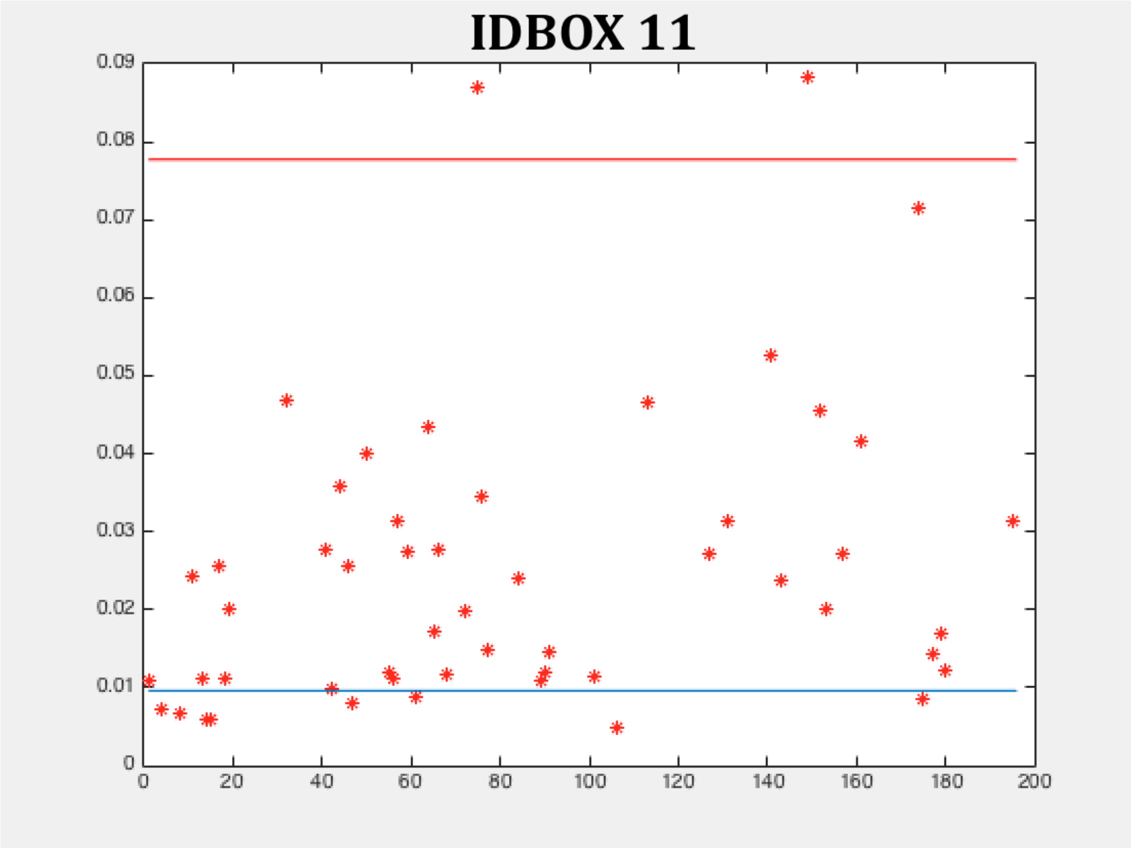


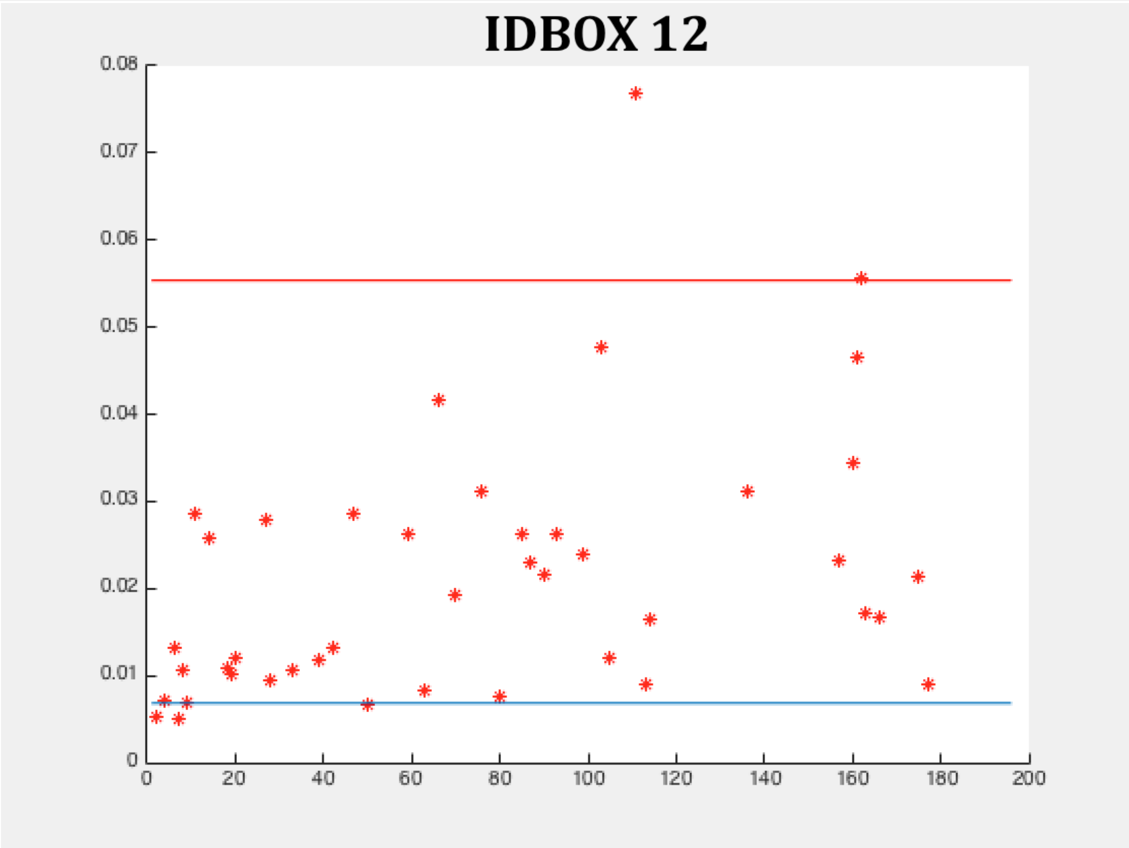
IDBOX 8

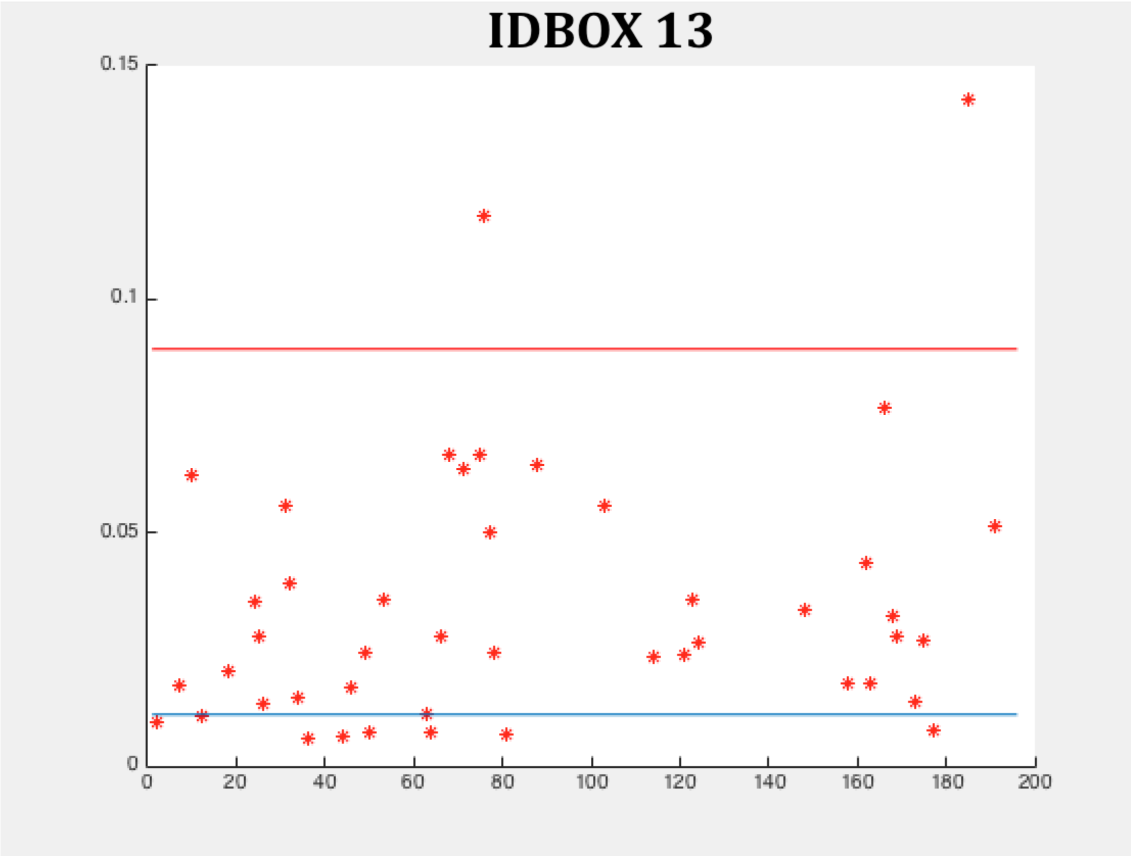


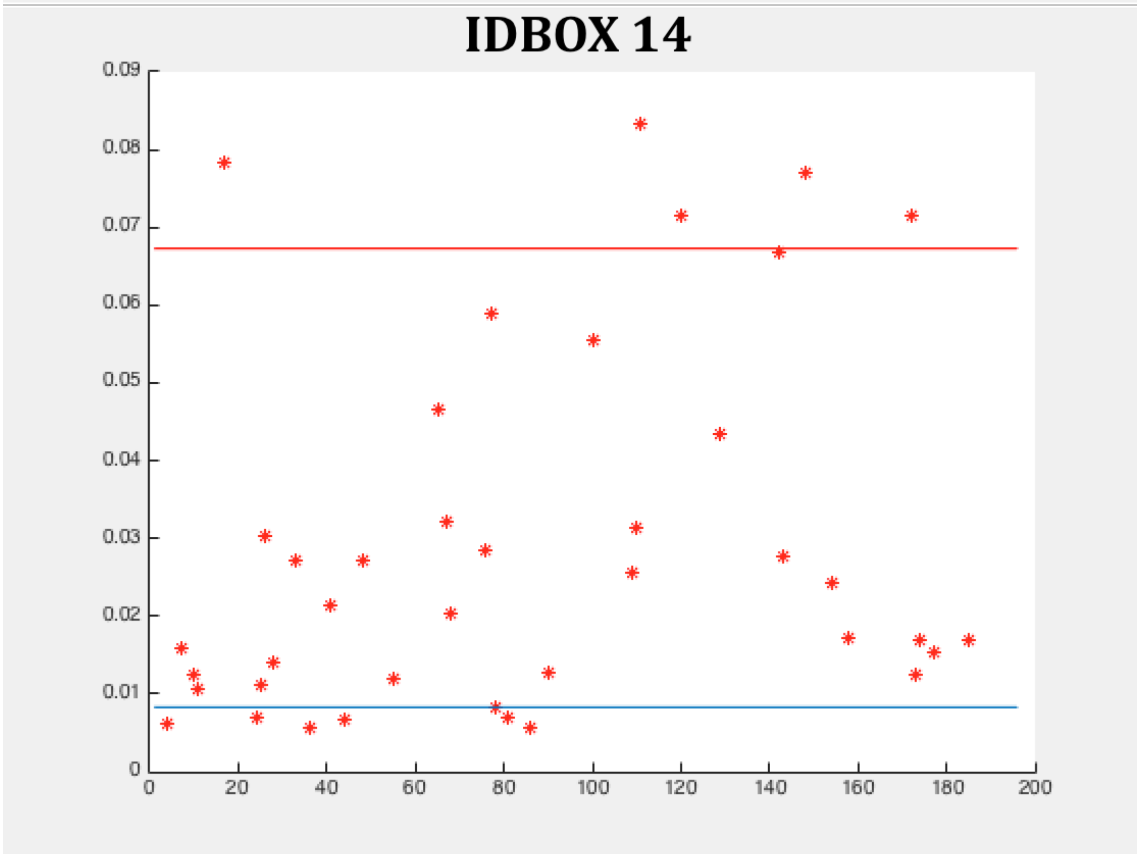

IDBOX 9
0.2
0.18
0.16
0.14
0.12
0.1
0.08
0.06
0.04
0.02
0
0
20
40
60
80
100
120
140
160
180
200

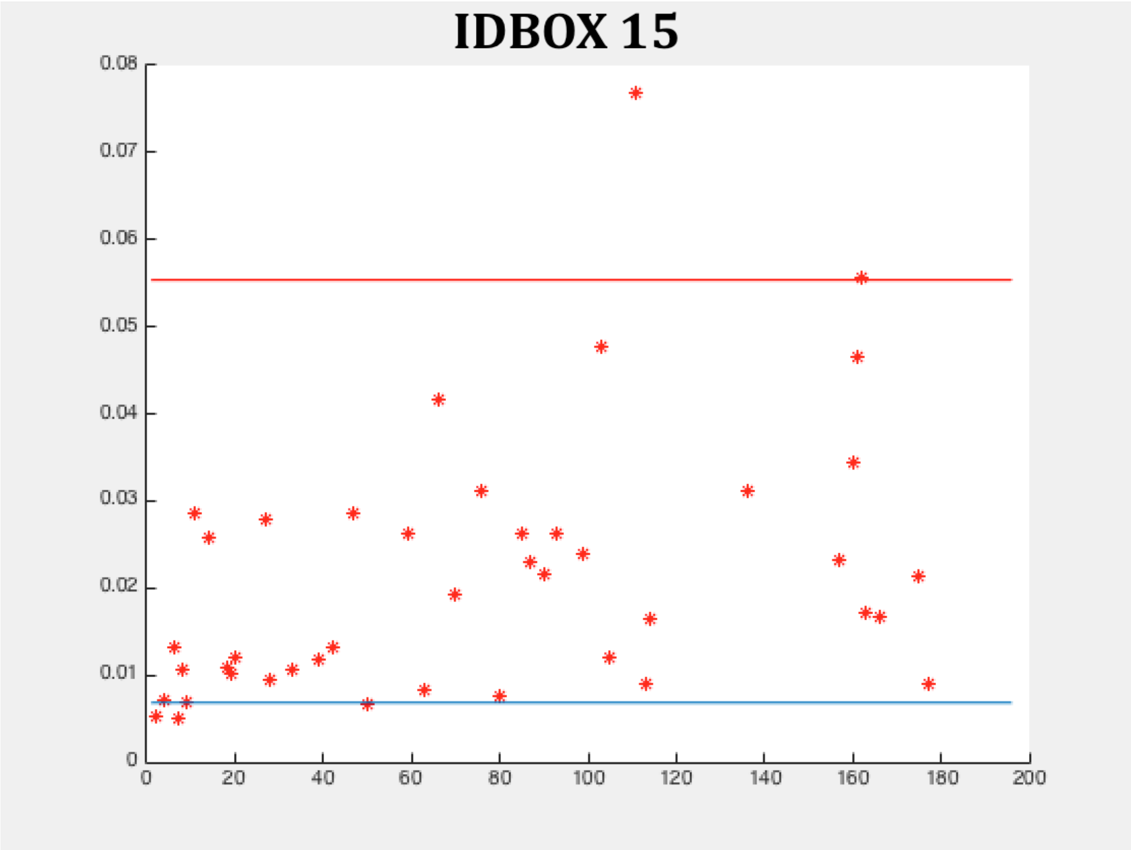

IDBOX 10
0.2
0.18
0.16
0.14
0.12
0.1
0.08
0.06
0.04
0.02
0
0
20
40
60
80
100
120
140
160
180
200

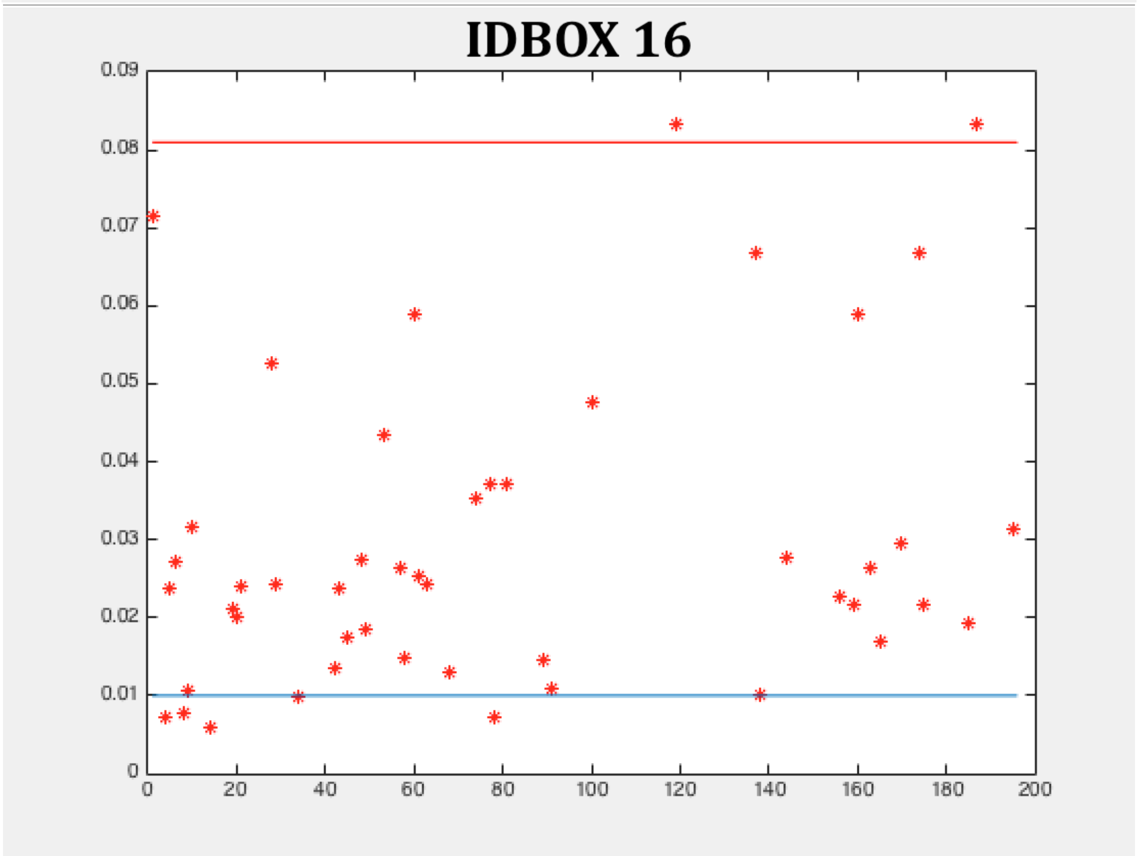

IDBOX 11
0.09
0.08
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0
0
20
40
60
80
100
120
140
160
180
200


IDBOX 12
0.08
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0
0
20
40
60
80
100
120
140
160
180
200

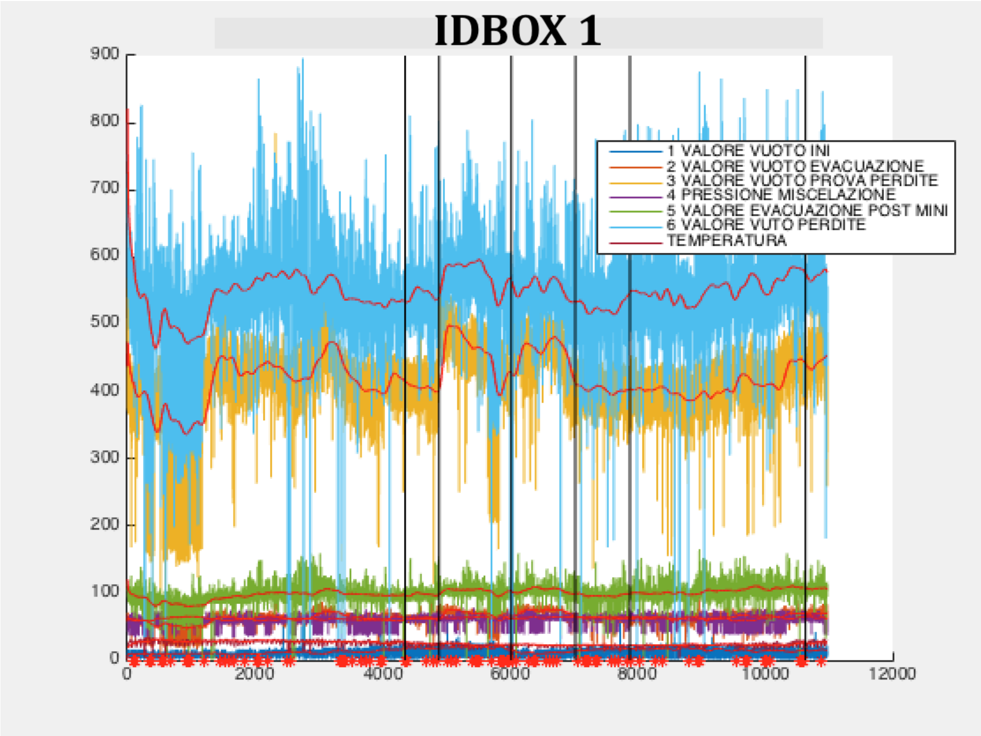

IDBOX 13
0.15
0.1
0.05
0
0
20
40
60
80
100
120
140
160
180
200

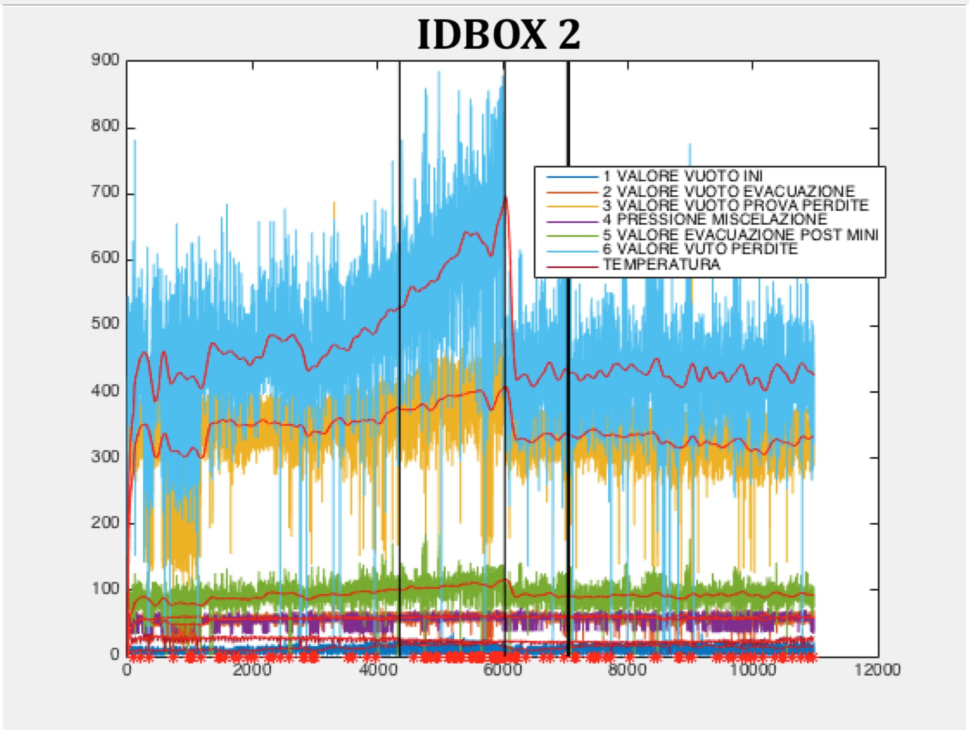

IDBOX 14
0.09
0.08
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0
0
20
40
60
80
100
120
140
160
180
200


IDBOX 15
0.08
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0
0 20 40 60 80 100 120 140 160 180 200

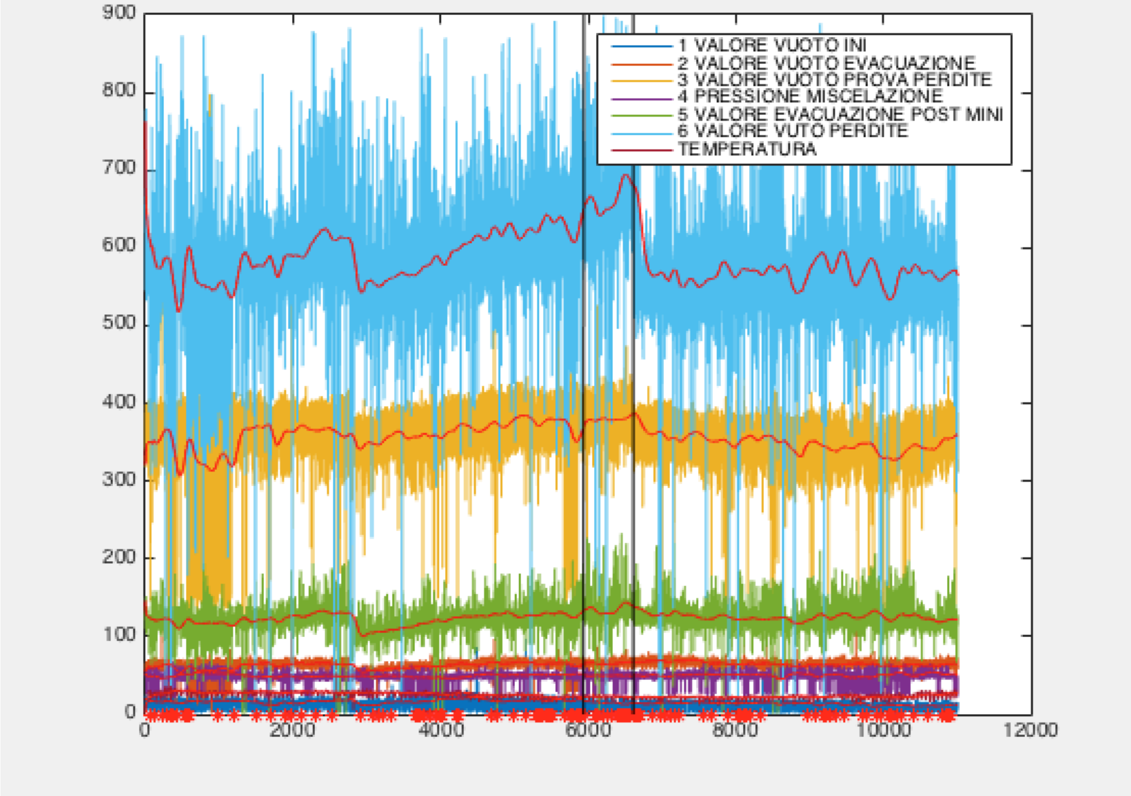

IDBOX 16
0.09
0.08
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0
0 20 40 60 80 100 120 140 160 180 200

ANNEX II

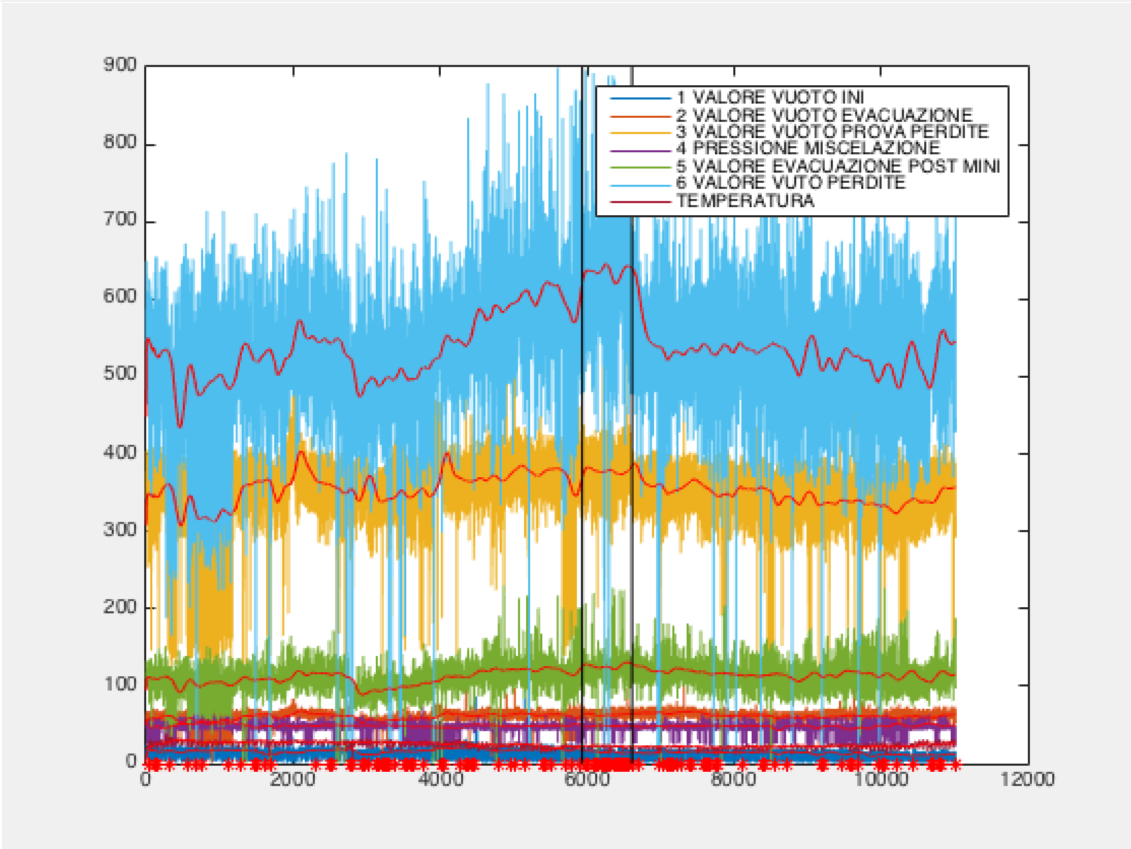

IDBOX 1
900
800
700
600
500
400
300
200
100
0
0
2000
4000
6000
8000
10000
12000
1 VALORE VUOTO INI
2 VALORE VUOTO EVACUAZIONE
3 VALORE VUOTO PROVA PERDITE
4 PRESSIONE MISCELAZIONE
5 VALORE EVACUAZIONE POST MINI
6 VALORE VUTO PERDITE
TEMPERATURA


IDBOX 2
900
800
700
600
500
400
300
200
100
0
0
2000
4000
6000
8000
10000
12000
1 VALORE VUOTO INI
2 VALORE VUOTO EVACUAZIONE
3 VALORE VUOTO PROVA PERDITE
4 PRESSIONE MISCELAZIONE
5 VALORE EVACUAZIONE POST MINI
6 VALORE VUTO PERDITE
TEMPERATURA

IDBOX 3

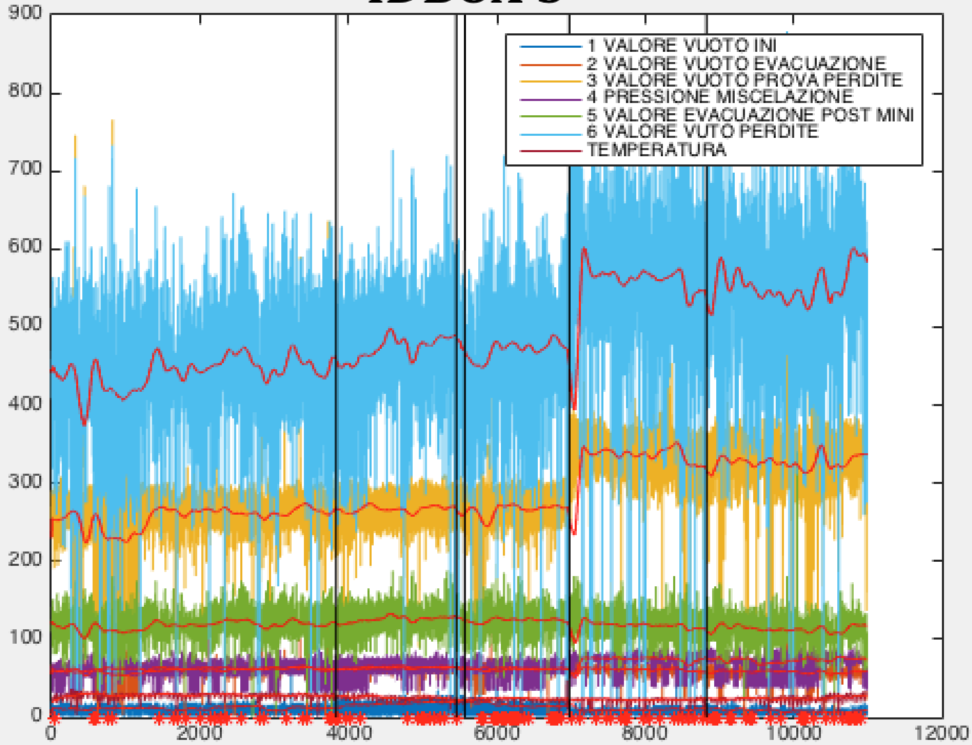


1 VALORE VUOTO INI
2 VALORE VUOTO EVACUAZIONE
3 VALORE VUOTO PROVA PERDITE
4 PRESSIONE MISCELAZIONE
5 VALORE EVACUAZIONE POST MINI
6 VALORE VUTO PERDITE
TEMPERATURA
900
800
700
600
500
400
300
200
100
0
0
2000
4000
6000
8000
10000
12000

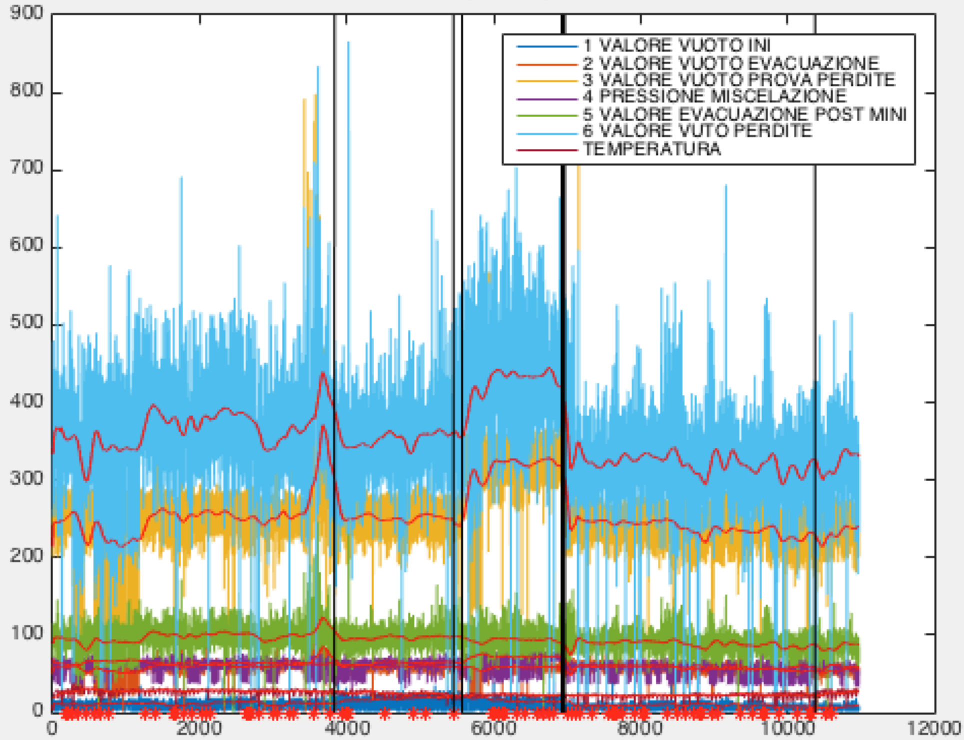


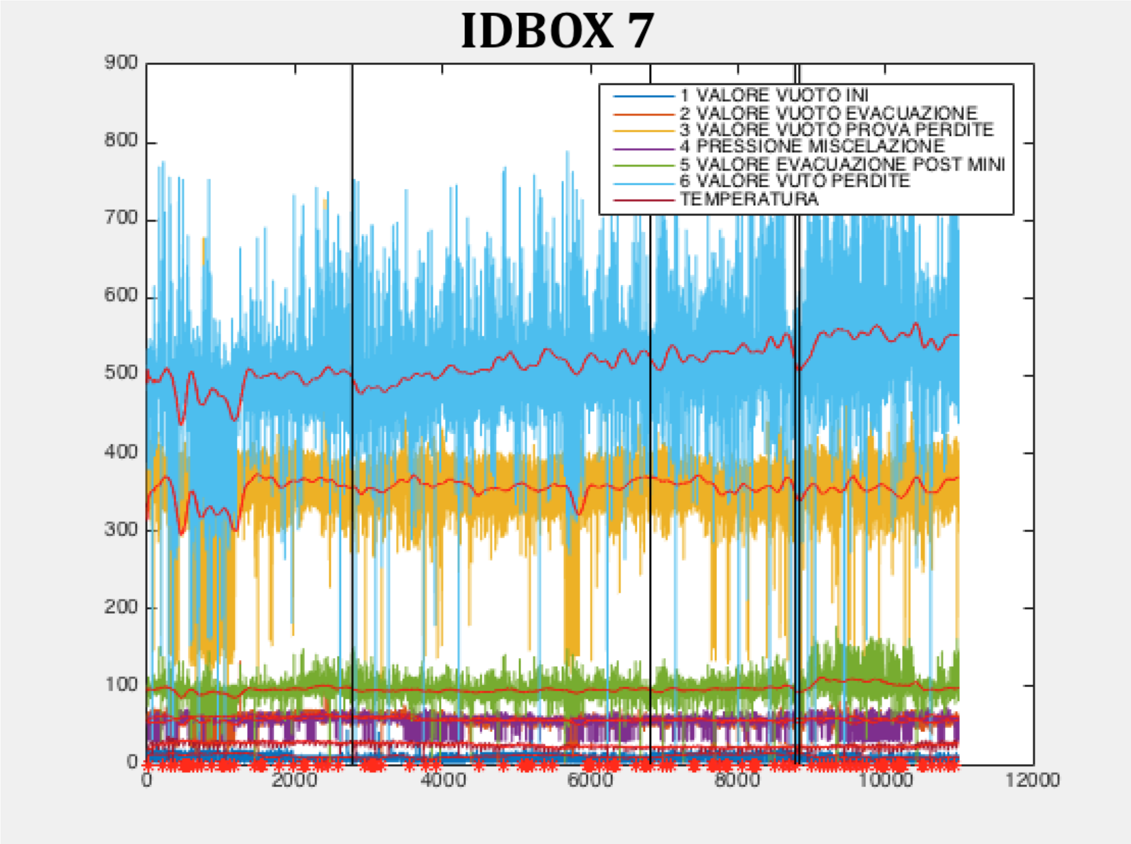
1 VALORE VUOTO INI
2 VALORE VUOTO EVACUAZIONE
3 VALORE VUOTO PROVA PERDITE
4 PRESSIONE MISCELAZIONE
5 VALORE EVACUAZIONE POST MINI
6 VALORE VUTO PERDITE
TEMPERATURA
900
800
700
600
500
400
300
200
100
0
0
2000
4000
6000
8000
10000
12000

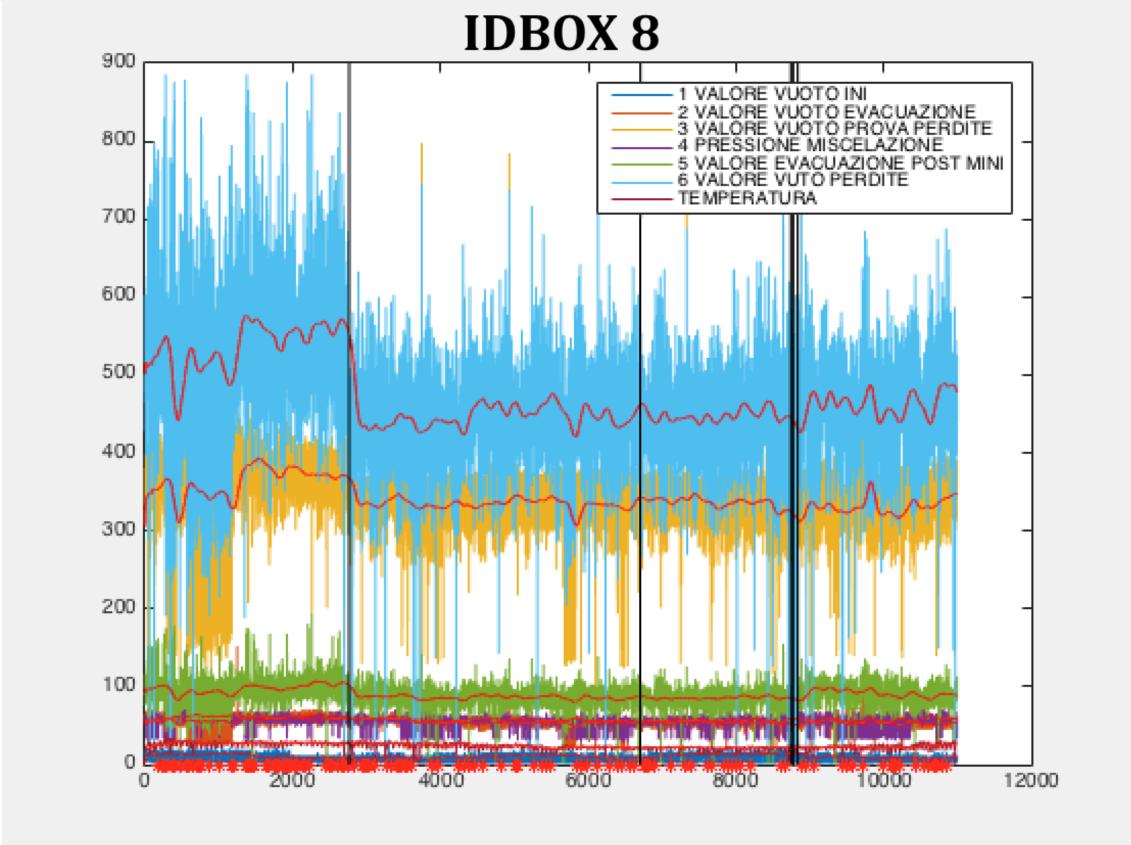
IDBOX 5

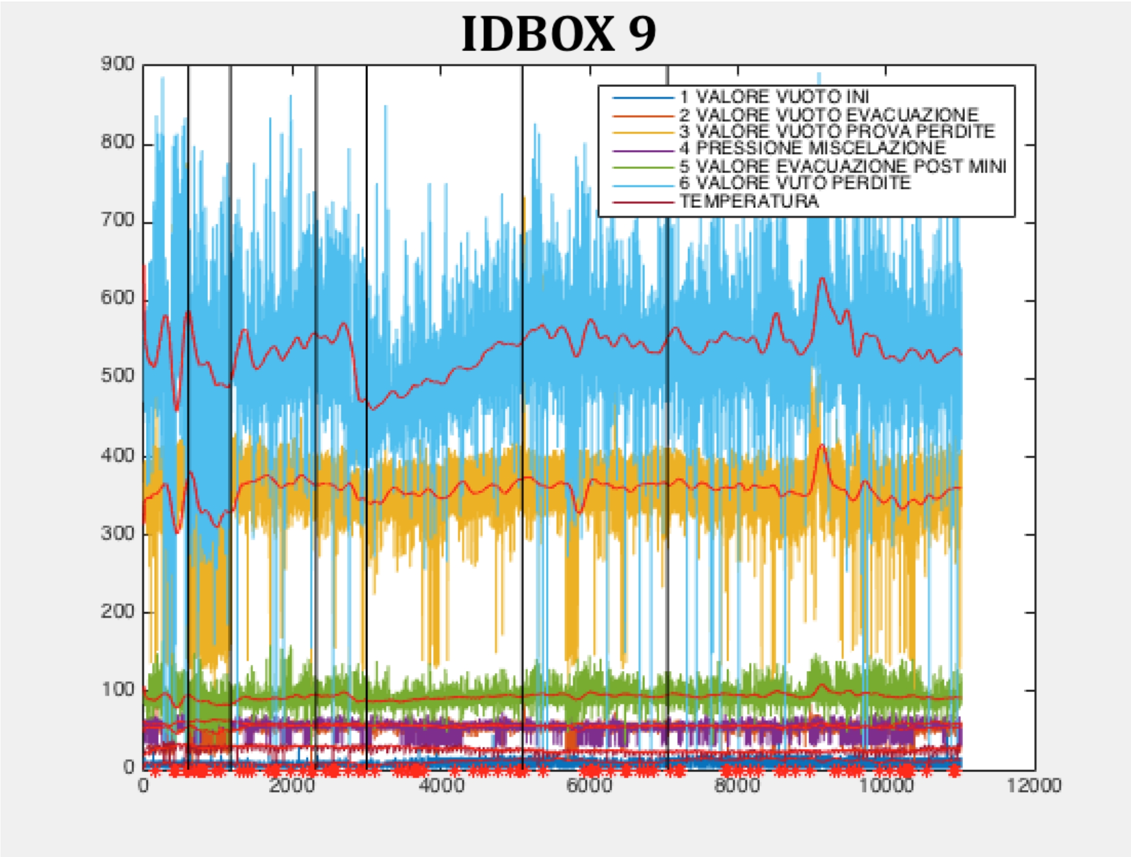


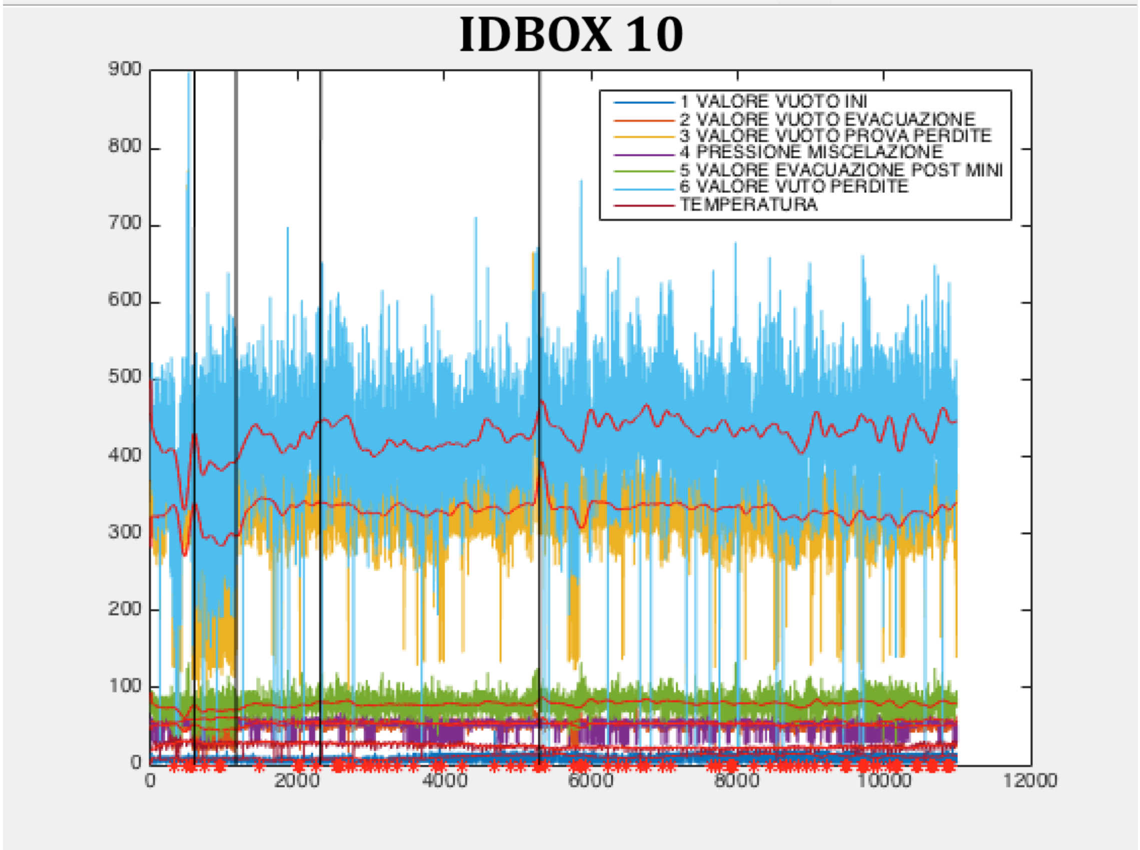
IDBOX 6

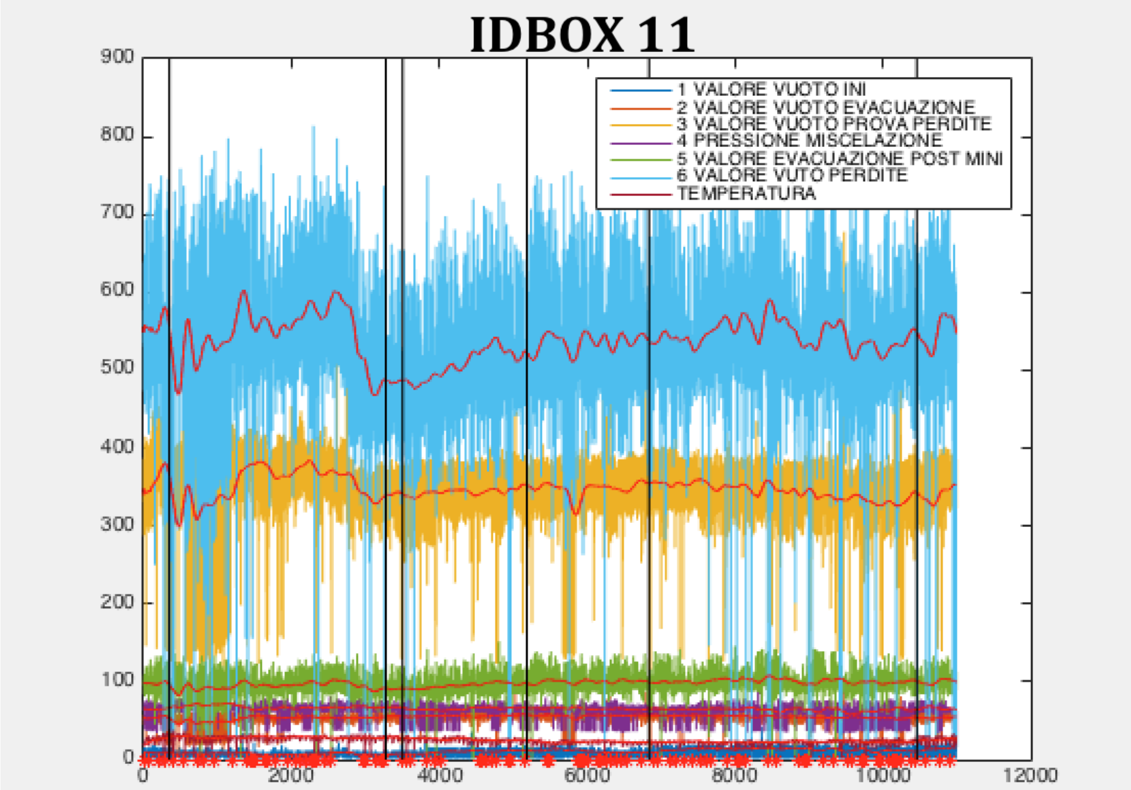


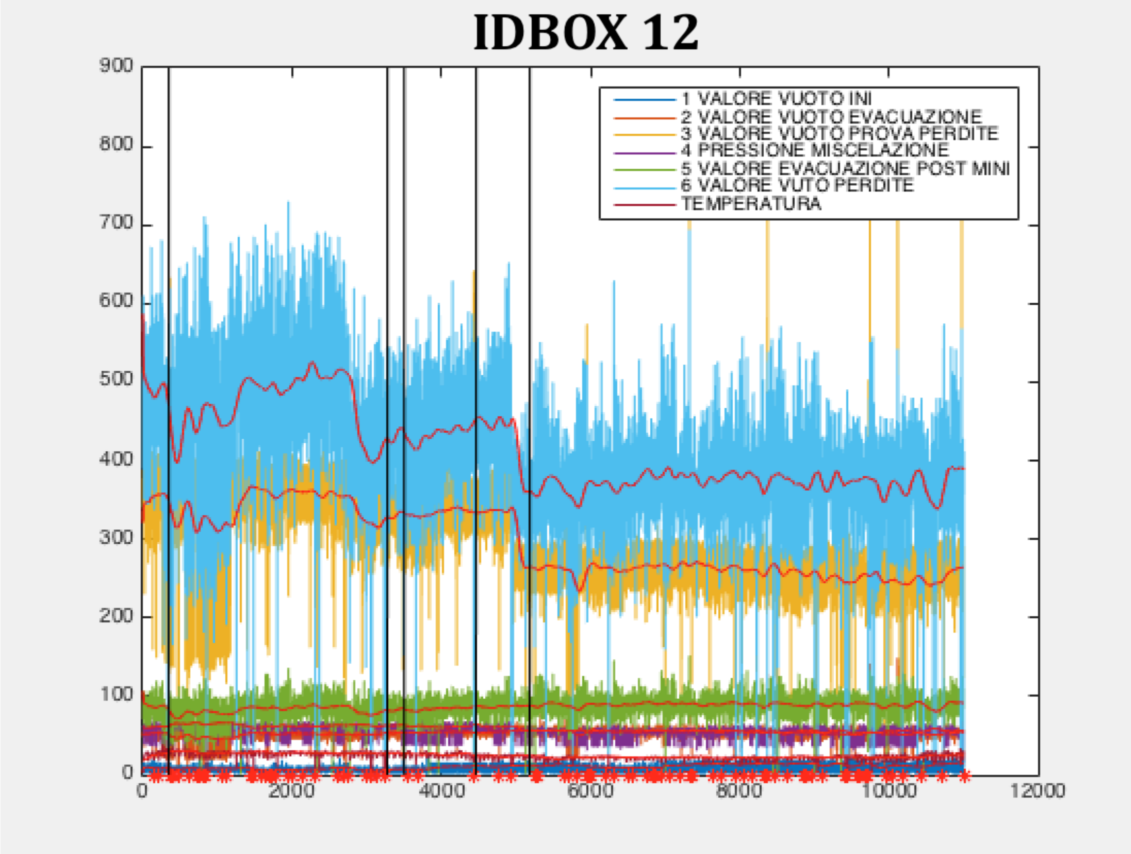

IDBOX 7
1 VALORE VUOTO INI
2 VALORE VUOTO EVACUAZIONE
3 VALORE VUOTO PROVA PERDITE
4 PRESSIONE MISCELAZIONE
5 VALORE EVACUAZIONE POST MINI
6 VALORE VUTO PERDITE
TEMPERATURA
0
100
200
300
400
500
600
700
800
900
2000
4000
6000
8000
10000
12000

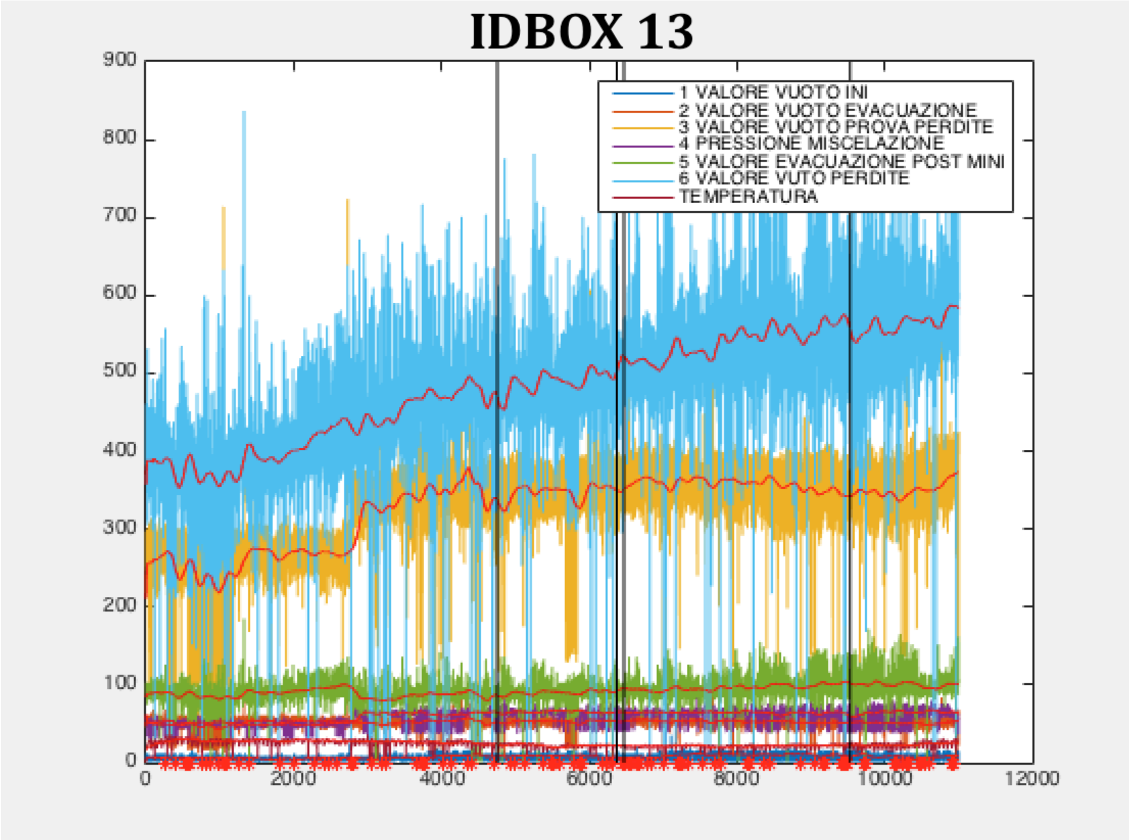

IDBOX 8
1 VALORE VUOTO INI
2 VALORE VUOTO EVACUAZIONE
3 VALORE VUOTO PROVA PERDITE
4 PRESSIONE MISCELAZIONE
5 VALORE EVACUAZIONE POST MINI
6 VALORE VUTO PERDITE
TEMPERATURA
0
100
200
300
400
500
600
700
800
900
2000
4000
6000
8000
10000
12000

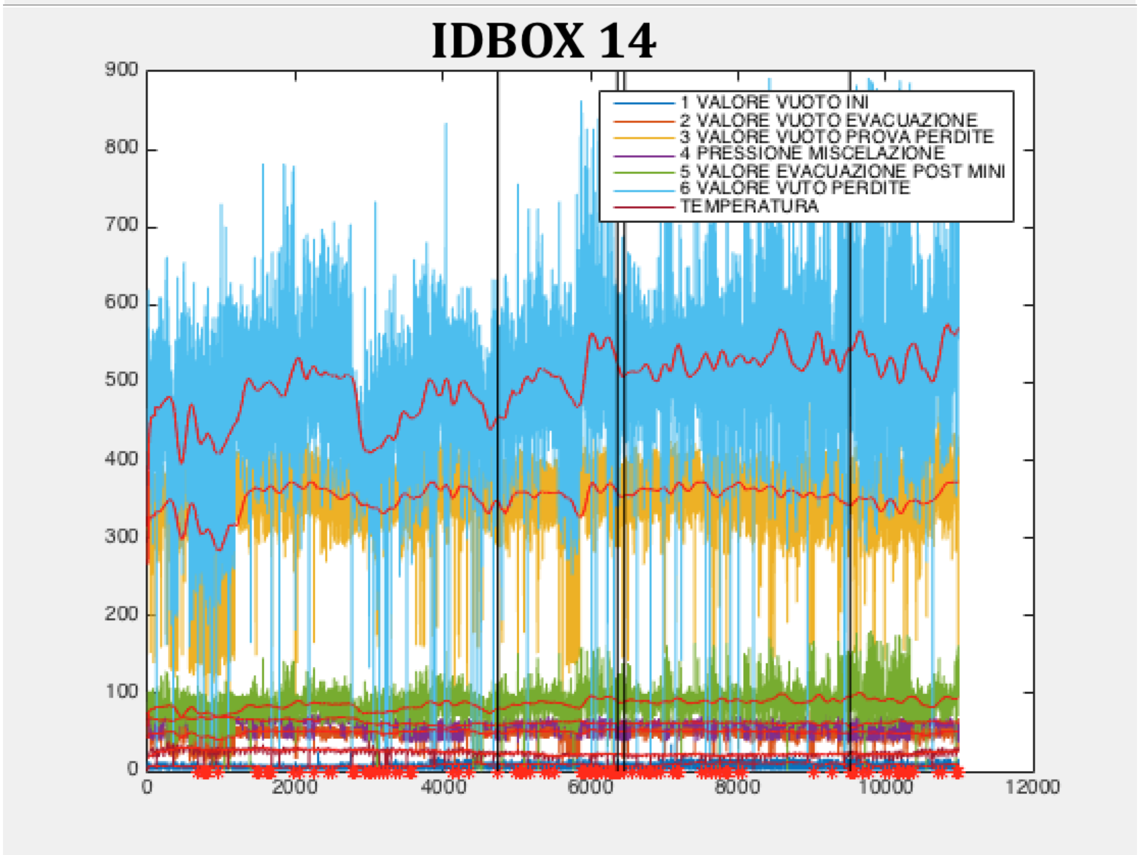

IDBOX 9
1 VALORE VUOTO INI
2 VALORE VUOTO EVACUAZIONE
3 VALORE VUOTO PROVA PERDITE
4 PRESSIONE MISCELAZIONE
5 VALORE EVACUAZIONE POST MINI
6 VALORE VUTO PERDITE
TEMPERATURA
900
800
700
600
500
400
300
200
100
0
2000
4000
6000
8000
10000
12000

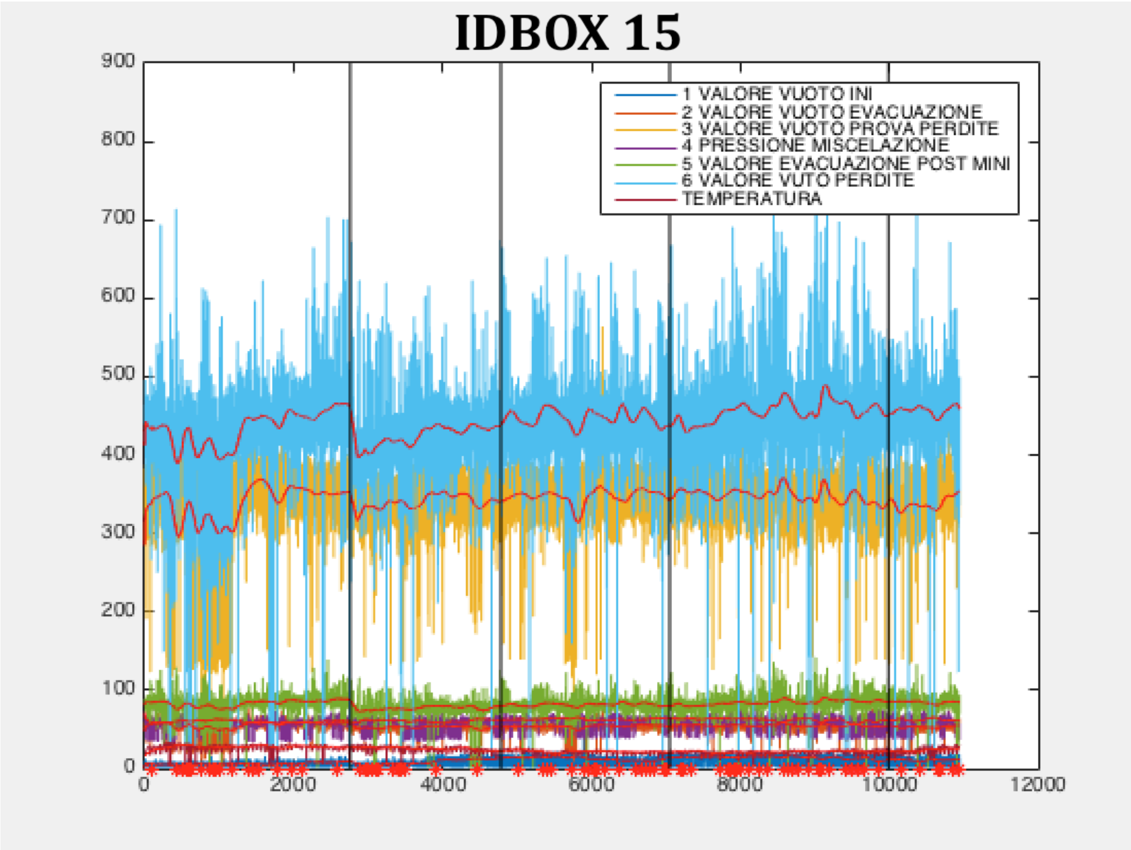

IDBOX 10
1 VALORE VUOTO INI
2 VALORE VUOTO EVACUAZIONE
3 VALORE VUOTO PROVA PERDITE
4 PRESSIONE MISCELAZIONE
5 VALORE EVACUAZIONE POST MINI
6 VALORE VUTO PERDITE
TEMPERATURA
900
800
700
600
500
400
300
200
100
0
2000
4000
6000
8000
10000
12000

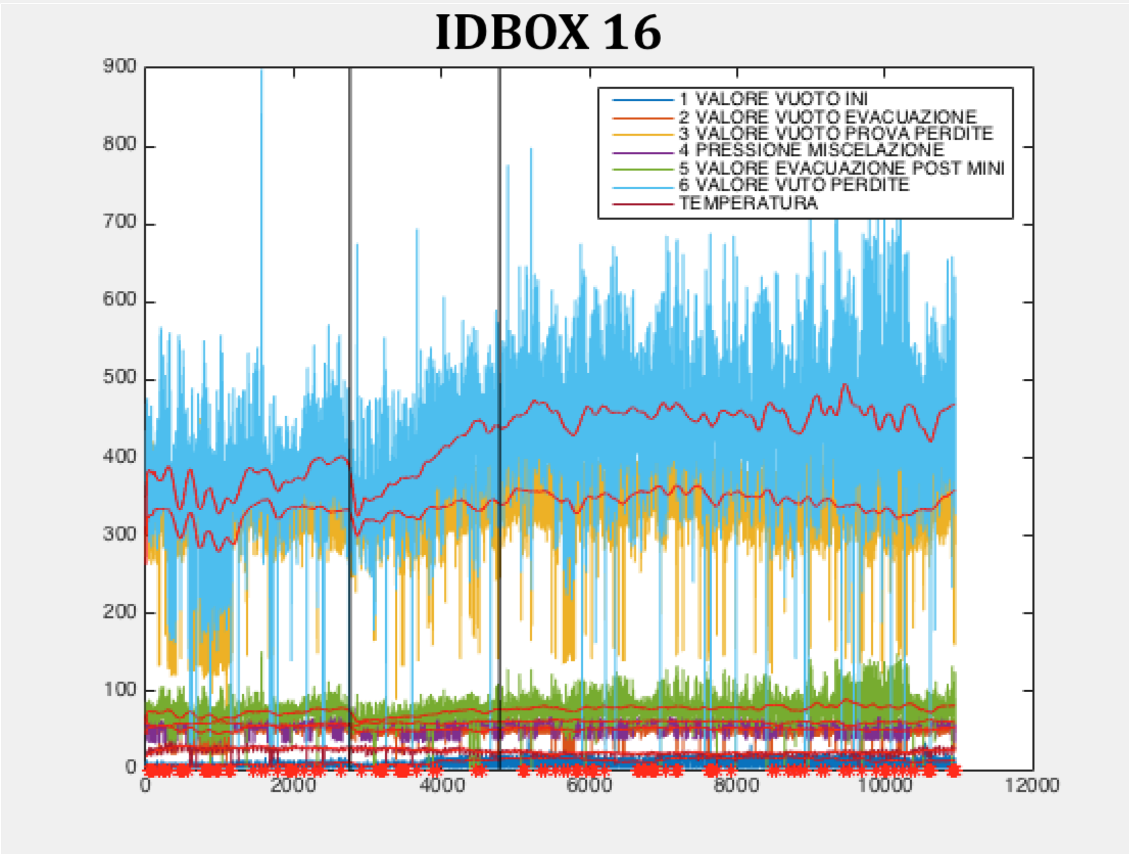

IDBOX 11
1 VALORE VUOTO INI
2 VALORE VUOTO EVACUAZIONE
3 VALORE VUOTO PROVA PERDITE
4 PRESSIONE MISCELAZIONE
5 VALORE EVACUAZIONE POST MINI
6 VALORE VUTO PERDITE
TEMPERATURA
900
800
700
600
500
400
300
200
100
0
0
2000
4000
6000
8000
10000
12000


IDBOX 12
1 VALORE VUOTO INI
2 VALORE VUOTO EVACUAZIONE
3 VALORE VUOTO PROVA PERDITE
4 PRESSIONE MISCELAZIONE
5 VALORE EVACUAZIONE POST MINI
6 VALORE VUTO PERDITE
TEMPERATURA
900
800
700
600
500
400
300
200
100
0
0
2000
4000
6000
8000
10000
12000

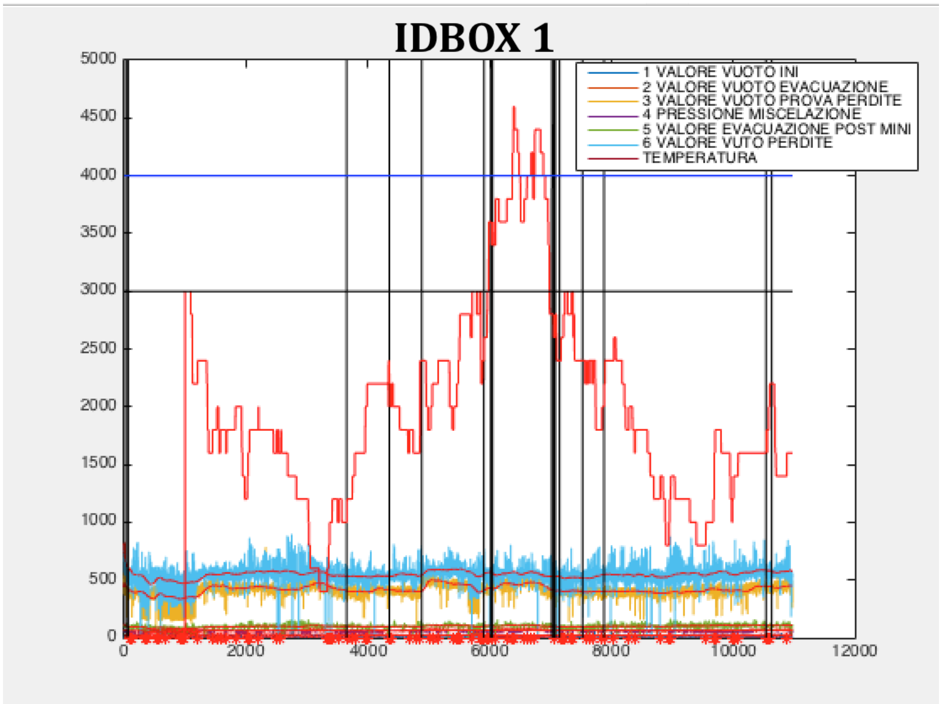

IDBOX 13
1 VALORE VUOTO INI
2 VALORE VUOTO EVACUAZIONE
3 VALORE VUOTO PROVA PERDITE
4 PRESSIONE MISCELAZIONE
5 VALORE EVACUAZIONE POST MINI
6 VALORE VUTO PERDITE
TEMPERATURA
900
800
700
600
500
400
300
200
100
0
0
2000
4000
6000
8000
10000
12000

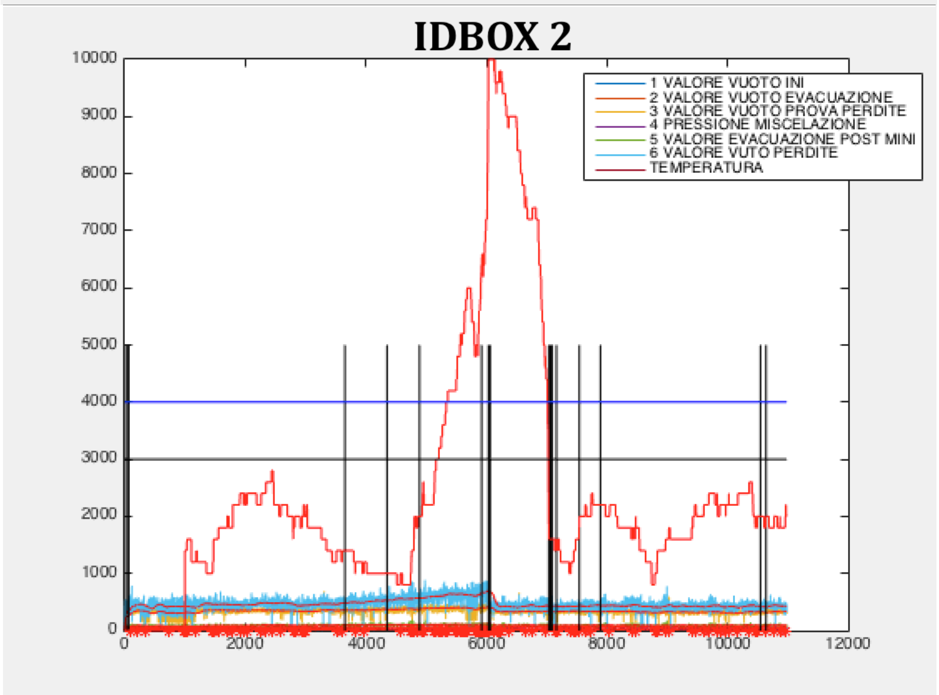

IDBOX 14
1 VALORE VUOTO INI
2 VALORE VUOTO EVACUAZIONE
3 VALORE VUOTO PROVA PERDITE
4 PRESSIONE MISCELAZIONE
5 VALORE EVACUAZIONE POST MINI
6 VALORE VUTO PERDITE
TEMPERATURA
900
800
700
600
500
400
300
200
100
0
0
2000
4000
6000
8000
10000
12000

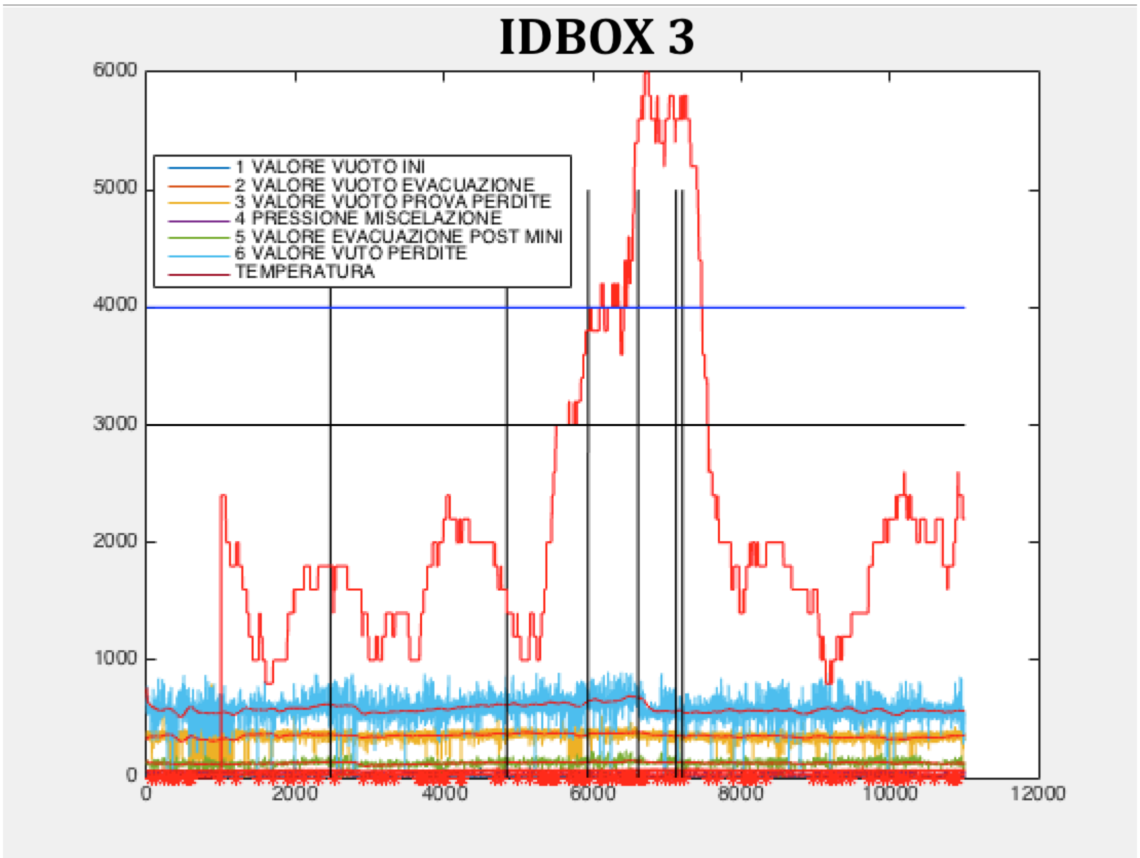

IDBOX 15
1 VALORE VUOTO INI
2 VALORE VUOTO EVACUAZIONE
3 VALORE VUOTO PROVA PERDITE
4 PRESSIONE MISCELAZIONE
5 VALORE EVACUAZIONE POST MINI
6 VALORE VUTO PERDITE
TEMPERATURA

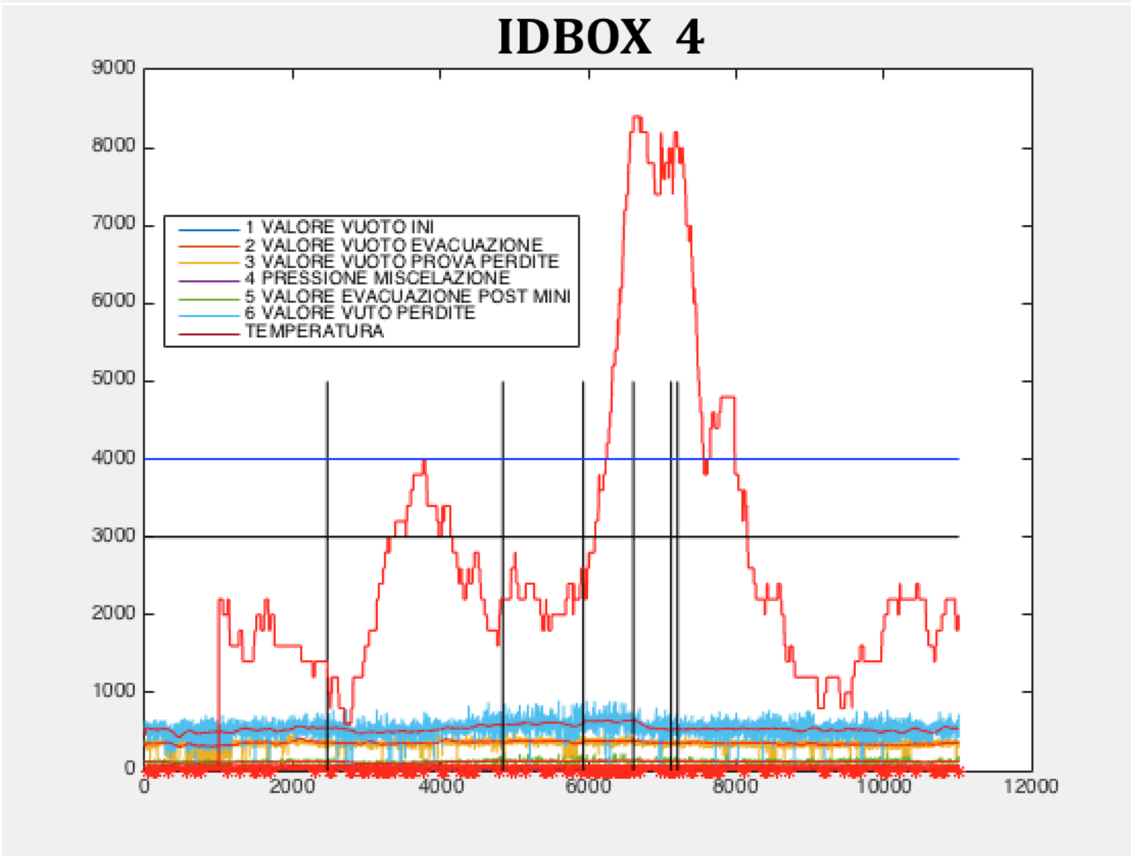

IDBOX 16
1 VALORE VUOTO INI
2 VALORE VUOTO EVACUAZIONE
3 VALORE VUOTO PROVA PERDITE
4 PRESSIONE MISCELAZIONE
5 VALORE EVACUAZIONE POST MINI
6 VALORE VUTO PERDITE
TEMPERATURA

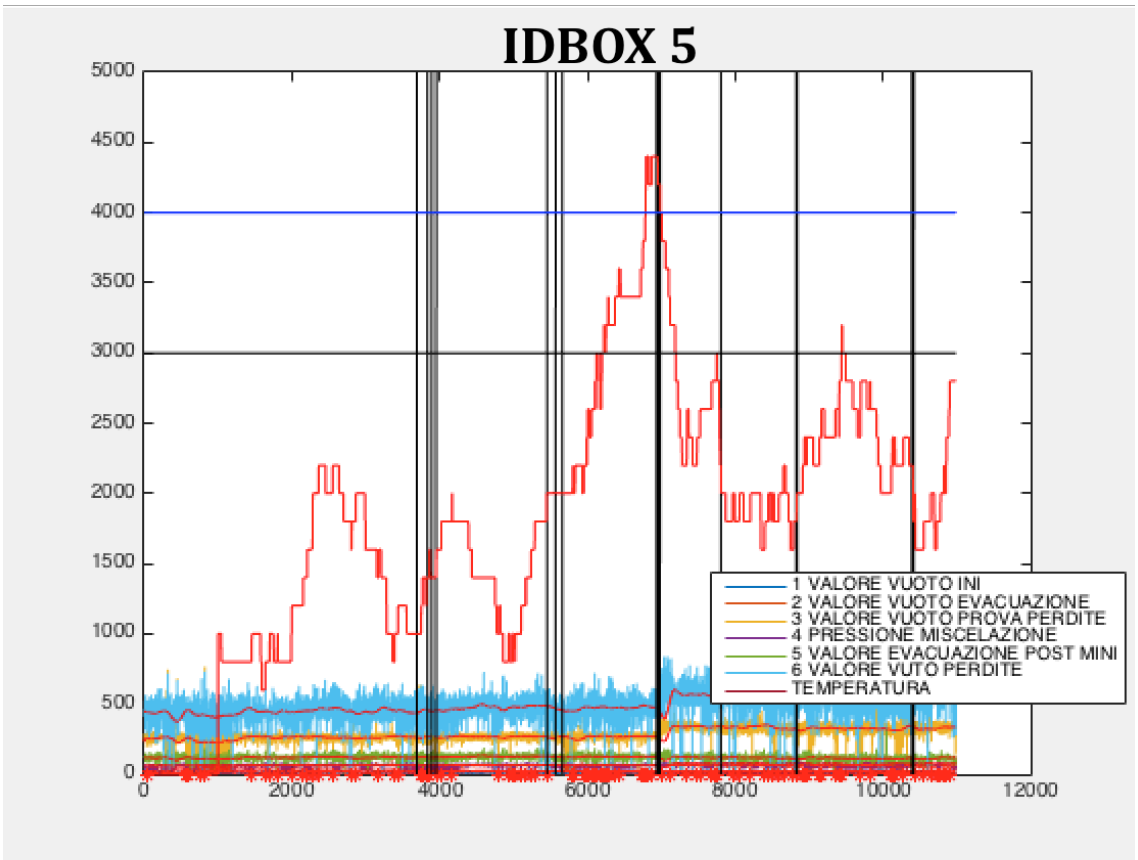
ANNEX III

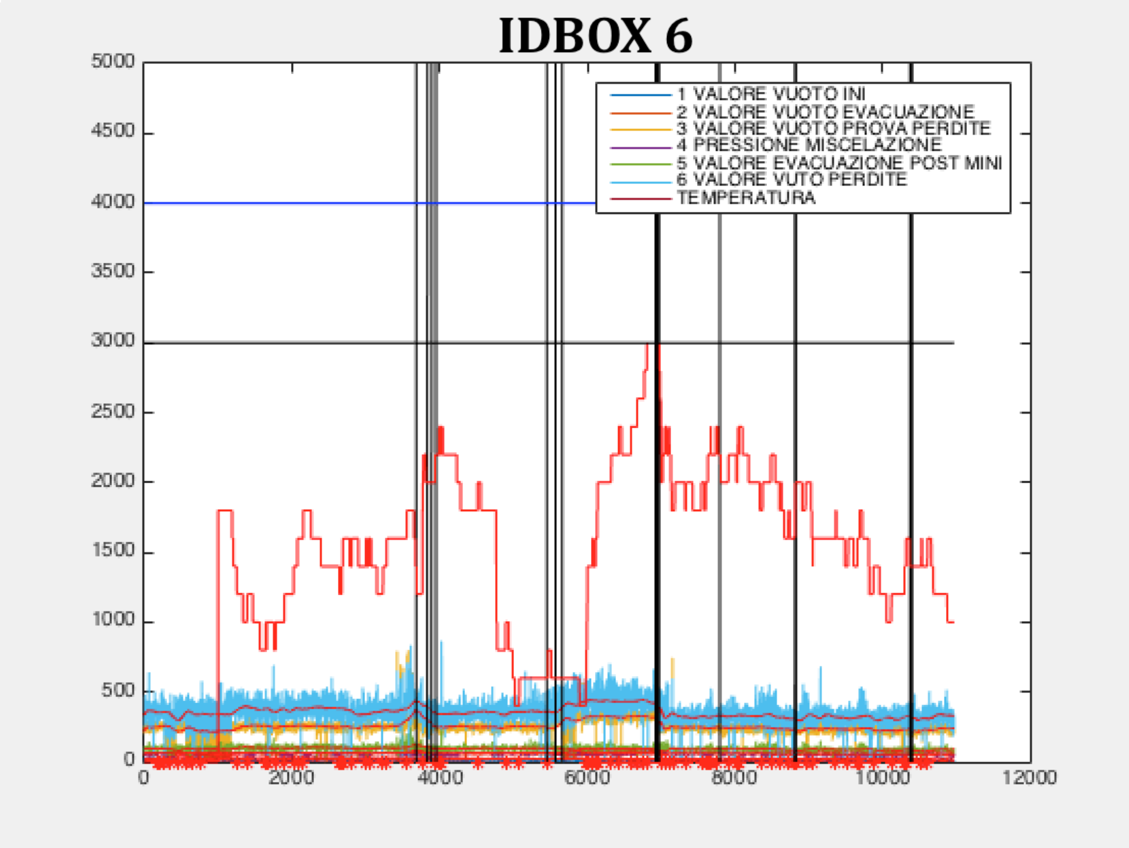

IDBOX 1
1 VALORE VUOTO INI
2 VALORE VUOTO EVACUAZIONE
3 VALORE VUOTO PROVA PERDITE
4 PRESSIONE MISCELAZIONE
5 VALORE EVACUAZIONE POST MINI
6 VALORE VUTO PERDITE
TEMPERATURA
5000
4500
4000
3500
3000
2500
2000
1500
1000
500
0
2000
4000
6000
8000
10000
12000

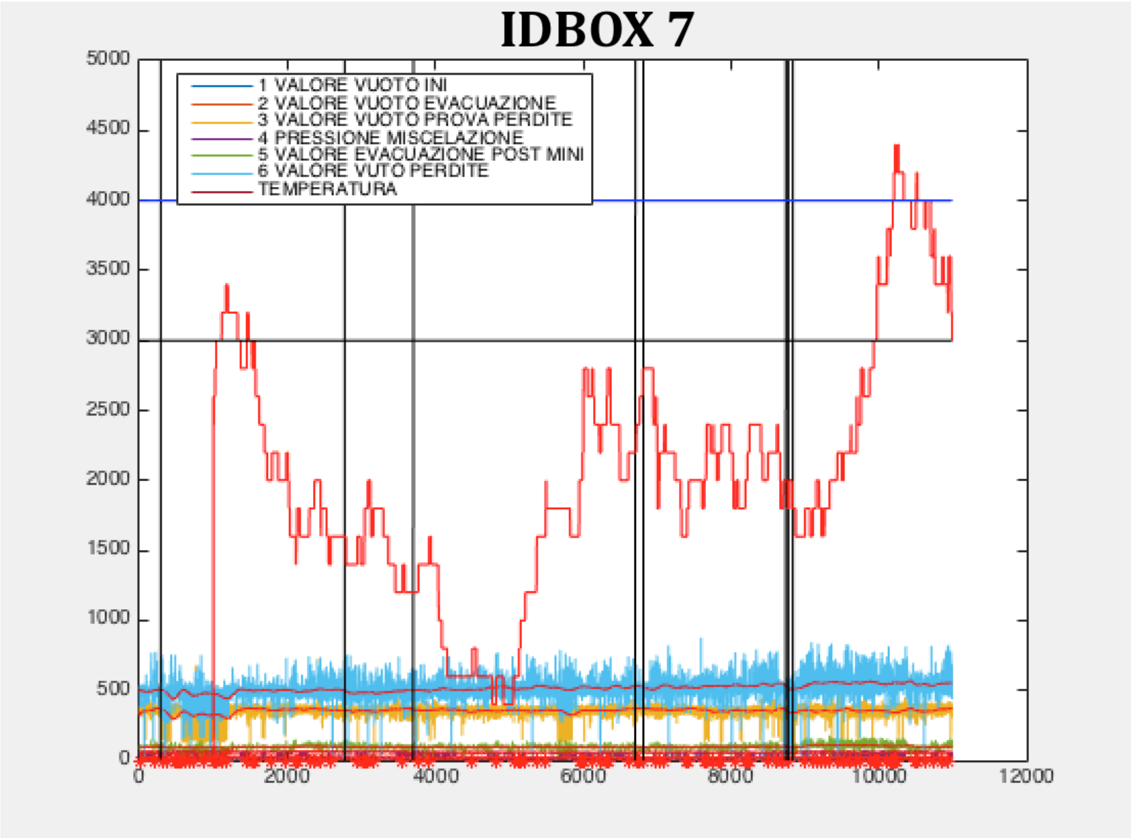

IDBOX 2
1 VALORE VUOTO INI
2 VALORE VUOTO EVACUAZIONE
3 VALORE VUOTO PROVA PERDITE
4 PRESSIONE MISCELAZIONE
5 VALORE EVACUAZIONE POST MINI
6 VALORE VUTO PERDITE
TEMPERATURA
10000
9000
8000
7000
6000
5000
4000
3000
2000
1000
0
2000
4000
6000
8000
10000
12000

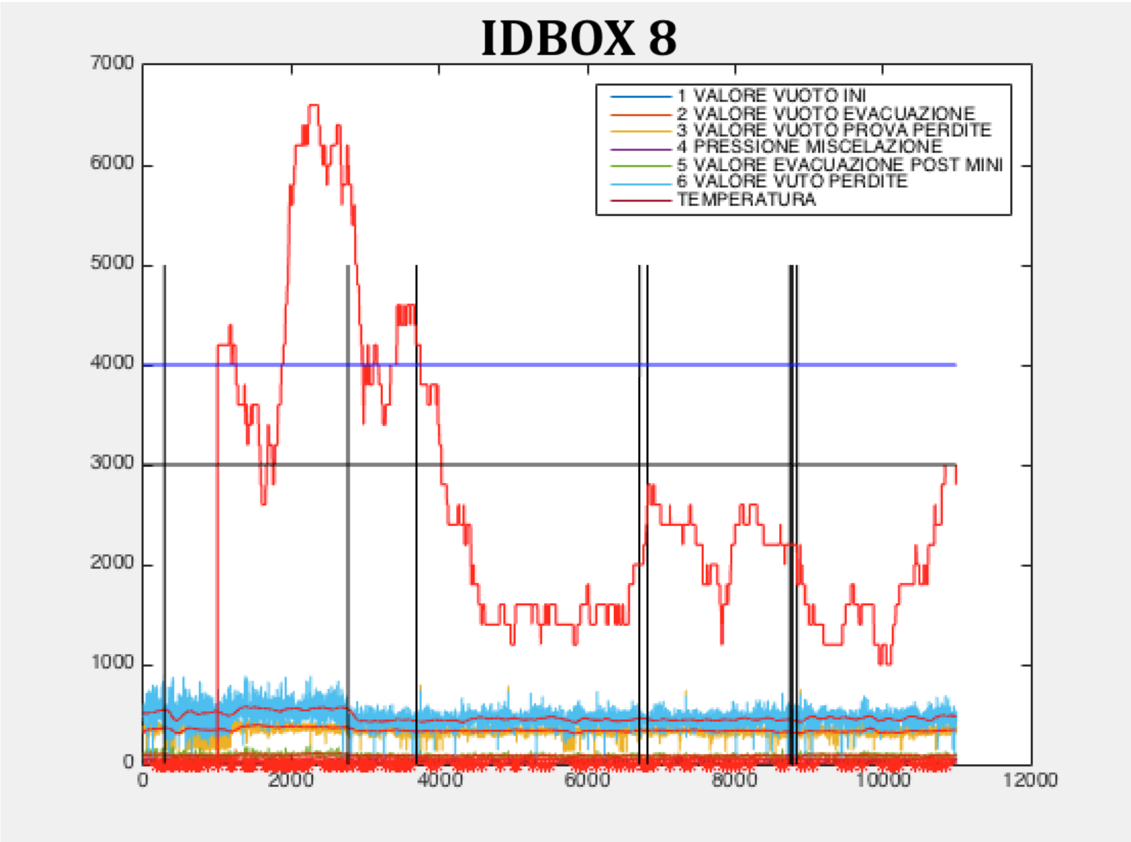

IDBOX 3
1 VALORE VUOTO INI
2 VALORE VUOTO EVACUAZIONE
3 VALORE VUOTO PROVA PERDITE
4 PRESSIONE MISCELAZIONE
5 VALORE EVACUAZIONE POST MINI
6 VALORE VUTO PERDITE
TEMPERATURA
6000
5000
4000
3000
2000
1000
0
0
2000
4000
6000
8000
10000
12000

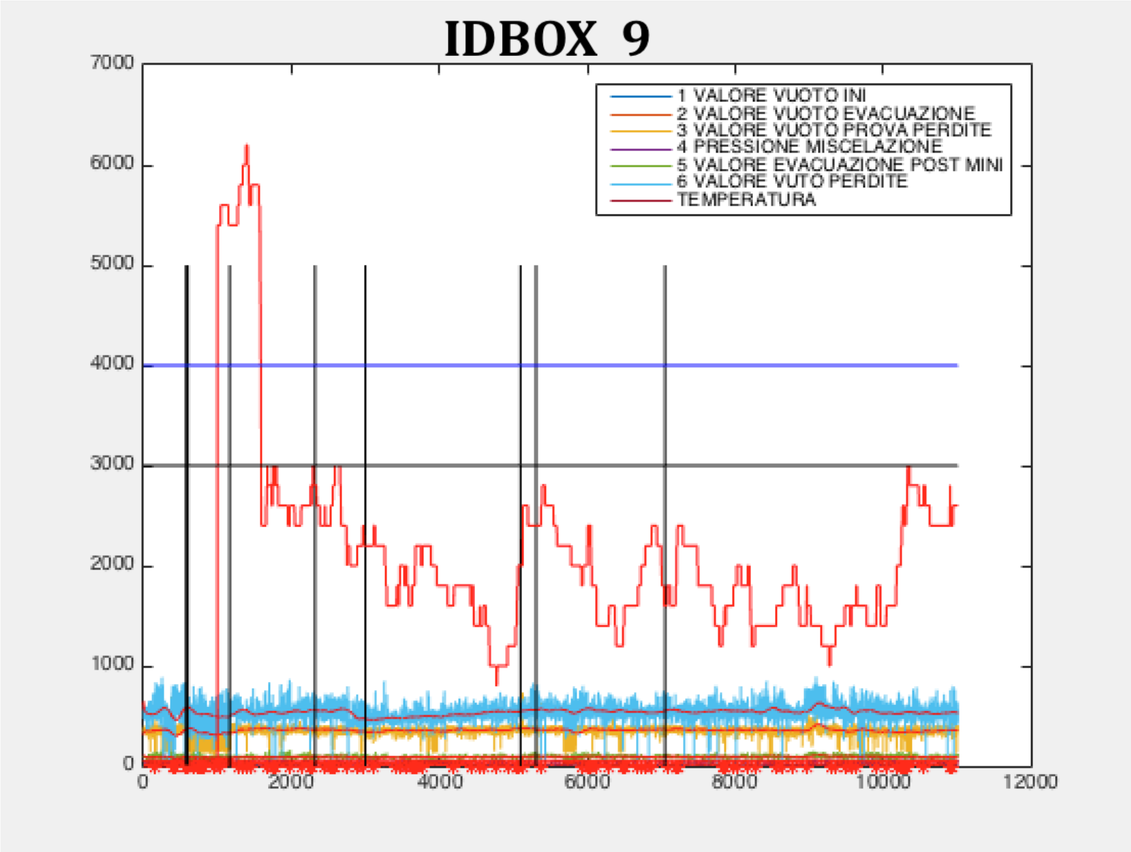

IDBOX 4
1 VALORE VUOTO INI
2 VALORE VUOTO EVACUAZIONE
3 VALORE VUOTO PROVA PERDITE
4 PRESSIONE MISCELAZIONE
5 VALORE EVACUAZIONE POST MINI
6 VALORE VUTO PERDITE
TEMPERATURA
9000
8000
7000
6000
5000
4000
3000
2000
1000
0
0
2000
4000
6000
8000
10000
12000

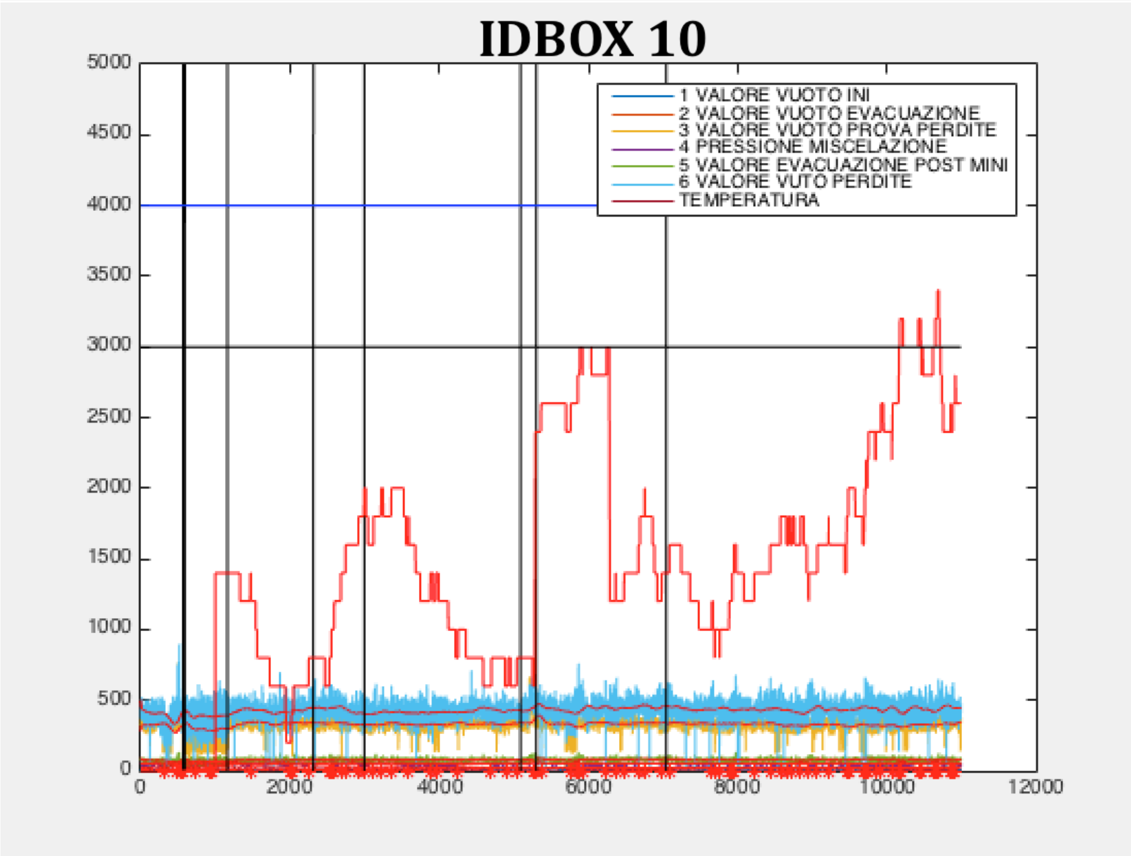

IDBOX 5
5000
4500
4000
3500
3000
2500
2000
1500
1000
500
0
0
2000
4000
6000
8000
10000
12000
1 VALORE VUOTO INI
2 VALORE VUOTO EVACUAZIONE
3 VALORE VUOTO PROVA PERDITE
4 PRESSIONE MISCELAZIONE
5 VALORE EVACUAZIONE POST MINI
6 VALORE VUTO PERDITE
TEMPERATURA


IDBOX 6
5000
4500
4000
3500
3000
2500
2000
1500
1000
500
0
0
2000
4000
6000
8000
10000
12000
1 VALORE VUOTO INI
2 VALORE VUOTO EVACUAZIONE
3 VALORE VUOTO PROVA PERDITE
4 PRESSIONE MISCELAZIONE
5 VALORE EVACUAZIONE POST MINI
6 VALORE VUTO PERDITE
TEMPERATURA

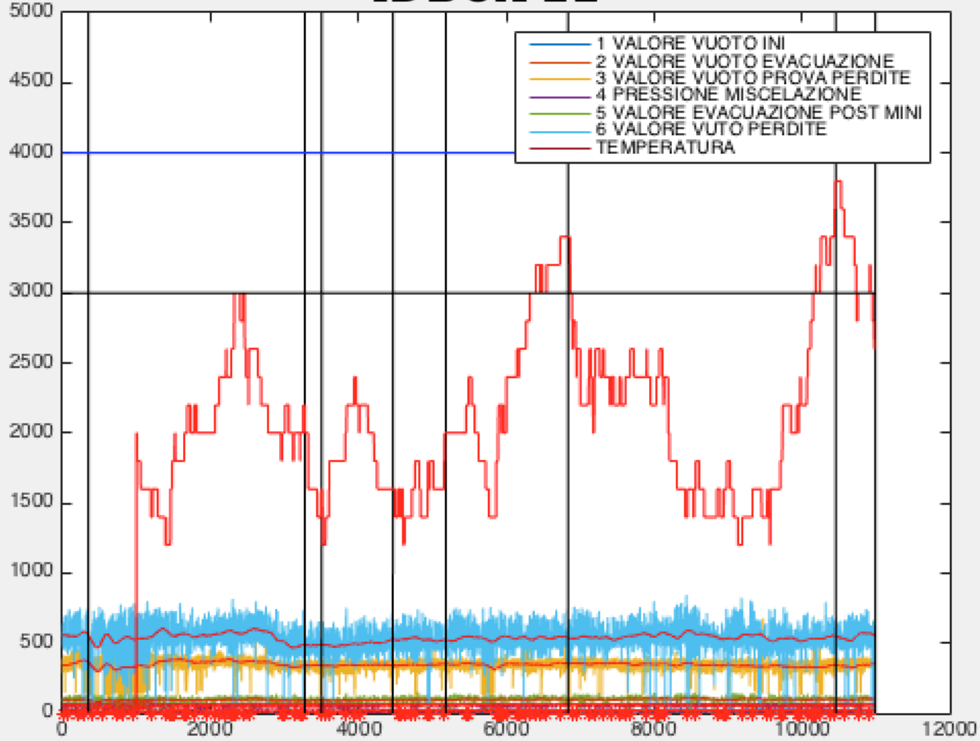

IDBOX 7
1 VALORE VUOTO INI
2 VALORE VUOTO EVACUAZIONE
3 VALORE VUOTO PROVA PERDITE
4 PRESSIONE MISCELAZIONE
5 VALORE EVACUAZIONE POST MINI
6 VALORE VUTO PERDITE
TEMPERATURA
5000
4500
4000
3500
3000
2500
2000
1500
1000
500
0
2000
4000
6000
8000
10000
12000


IDBOX 8
1 VALORE VUOTO INI
2 VALORE VUOTO EVACUAZIONE
3 VALORE VUOTO PROVA PERDITE
4 PRESSIONE MISCELAZIONE
5 VALORE EVACUAZIONE POST MINI
6 VALORE VUTO PERDITE
TEMPERATURA
7000
6000
5000
4000
3000
2000
1000
0
2000
4000
6000
8000
10000
12000

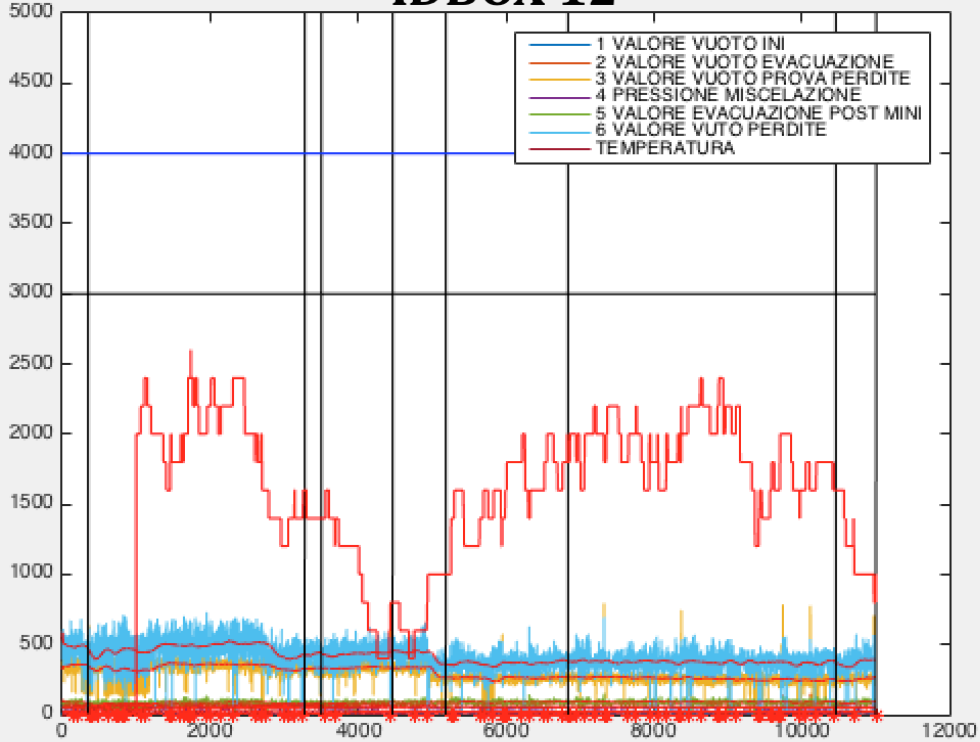

IDBOX 9
7000
6000
5000
4000
3000
2000
1000
0
0
2000
4000
6000
8000
10000
12000
1 VALORE VUOTO INI
2 VALORE VUOTO EVACUAZIONE
3 VALORE VUOTO PROVA PERDITE
4 PRESSIONE MISCELAZIONE
5 VALORE EVACUAZIONE POST MINI
6 VALORE VUTO PERDITE
TEMPERATURA

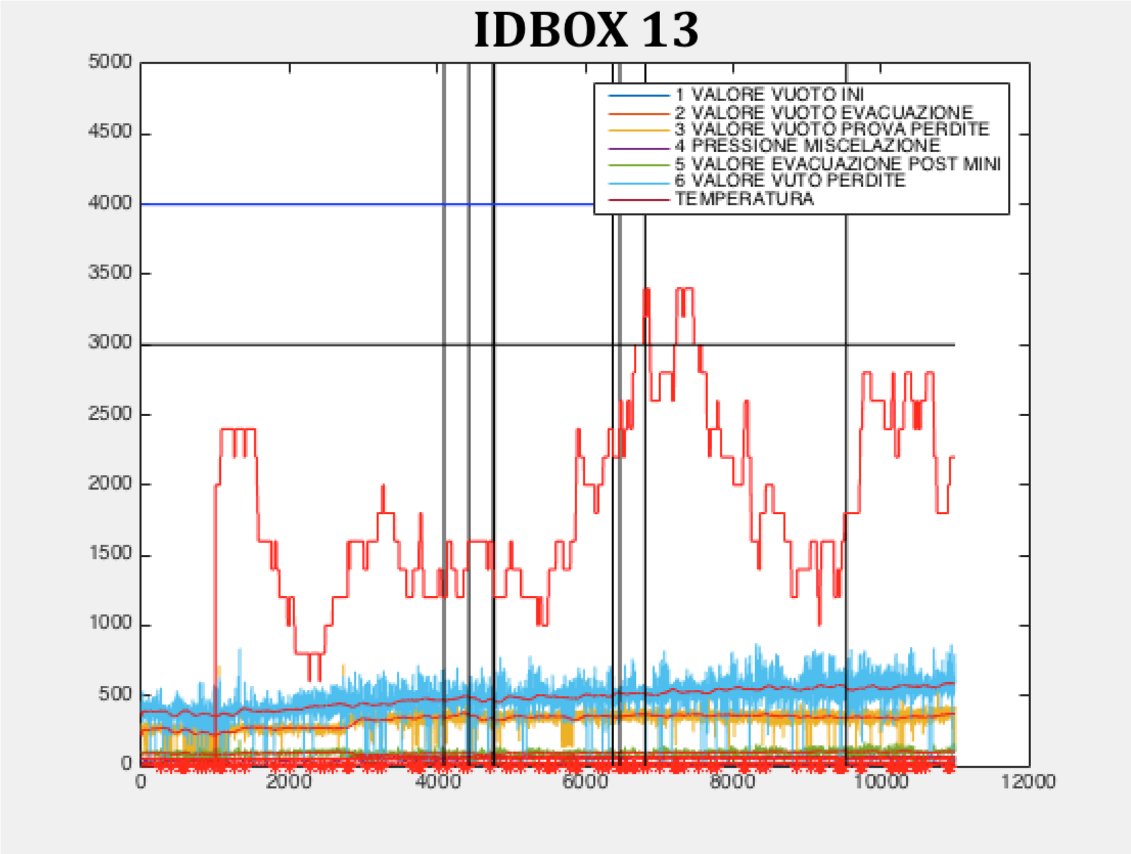

IDBOX 10
5000
4500
4000
3500
3000
2500
2000
1500
1000
500
0
0
2000
4000
6000
8000
10000
12000
1 VALORE VUOTO INI
2 VALORE VUOTO EVACUAZIONE
3 VALORE VUOTO PROVA PERDITE
4 PRESSIONE MISCELAZIONE
5 VALORE EVACUAZIONE POST MINI
6 VALORE VUTO PERDITE
TEMPERATURA

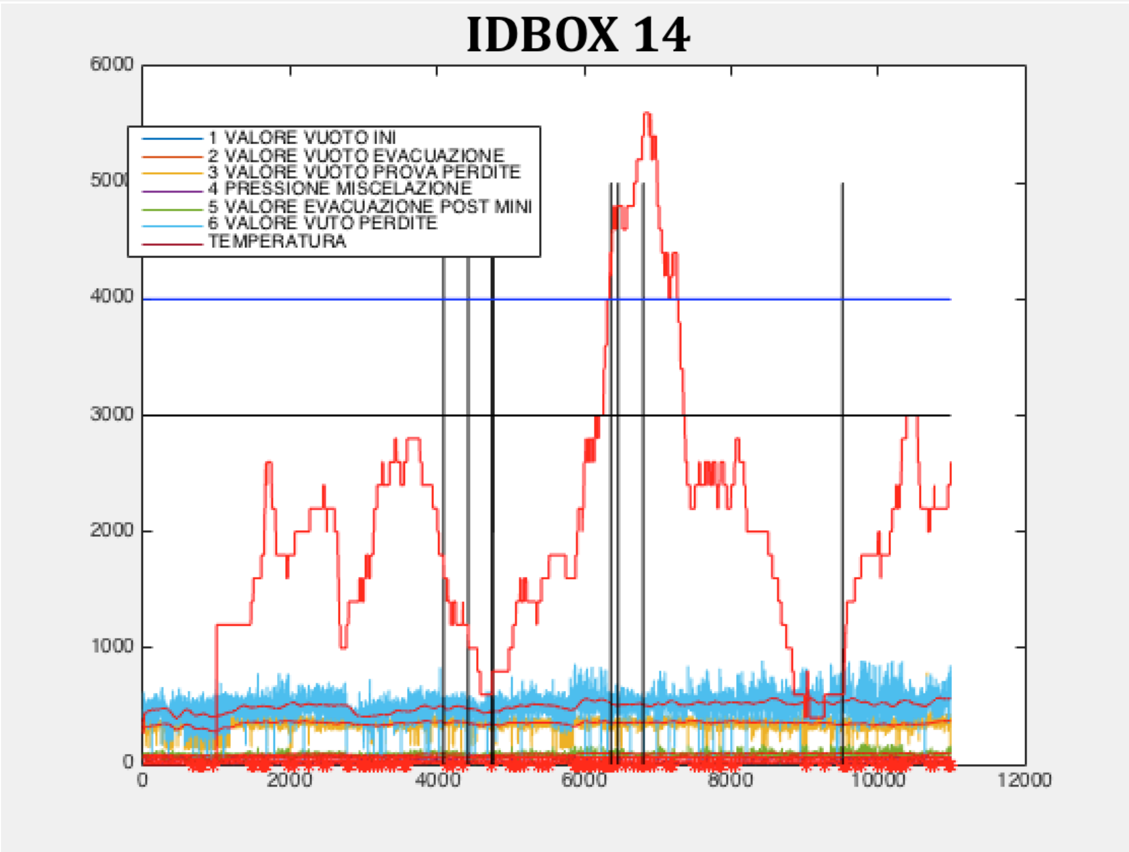
IDBOX 11



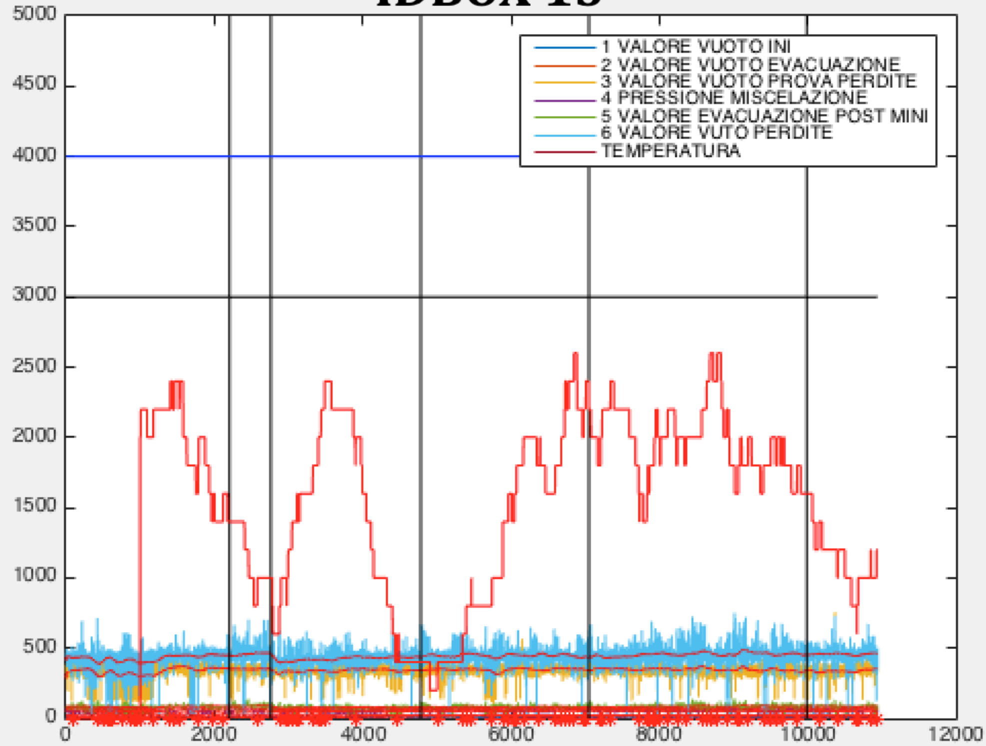
IDBOX 12



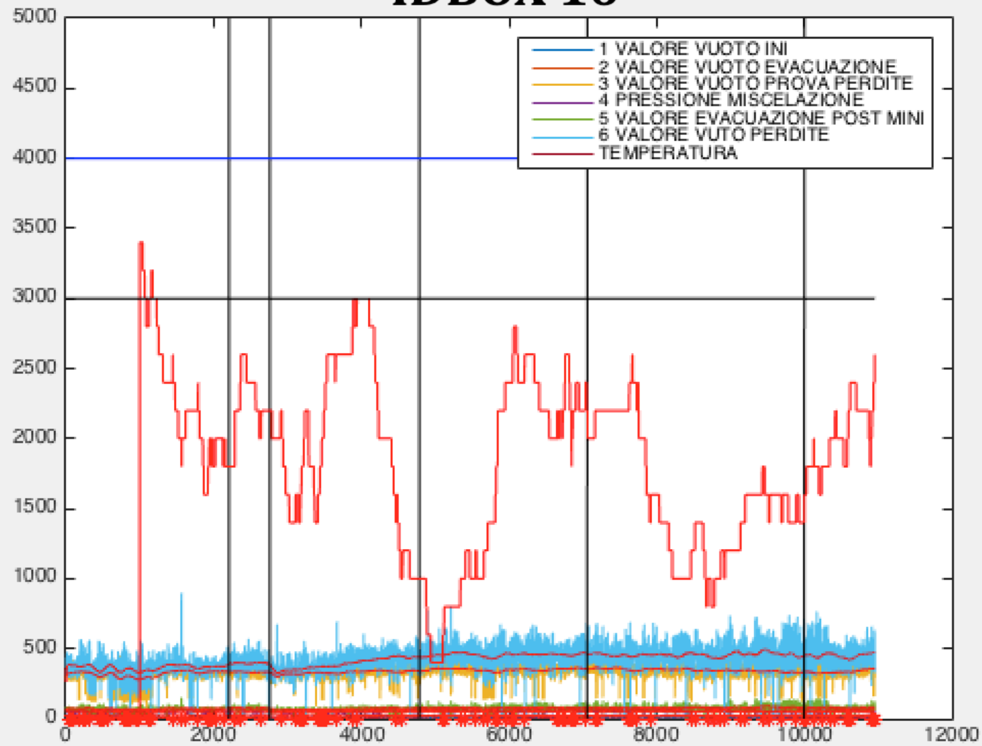

IDBOX 13
1 VALORE VUOTO INI
2 VALORE VUOTO EVACUAZIONE
3 VALORE VUOTO PROVA PERDITE
4 PRESSIONE MISCELAZIONE
5 VALORE EVACUAZIONE POST MINI
6 VALORE VUTO PERDITE
TEMPERATURA
5000
4500
4000
3500
3000
2500
2000
1500
1000
500
0
0
2000
4000
6000
8000
10000
12000


IDBOX 14
1 VALORE VUOTO INI
2 VALORE VUOTO EVACUAZIONE
3 VALORE VUOTO PROVA PERDITE
4 PRESSIONE MISCELAZIONE
5 VALORE EVACUAZIONE POST MINI
6 VALORE VUTO PERDITE
TEMPERATURA
6000
4000
3000
2000
1000
0
0
2000
4000
6000
8000
10000
12000

IDBOX 15



IDBOX 16



ANNEX IV - DATA PROCESSING AND ANALYSIS SCRIPT


```
%% INITIALITATION
clear all
clc
%%

B=xlsread('LB_Dati.xlsx');

%% DELETING ERROR
i=0;
k=0;
length(B)
while i<length(B)
    i=i+1;
    if B(i,7)==0 || B(i,2)==0
        k=k+1;
        B(i,:)=[];
        i=i-1;
    end
end

%%
rr=fopen('refrigerante.txt','r');
r=fscanf(rr,'%f');
fclose(rr)
%%
%%
length(B)
%%
%%
serial_number1=B(:,1);
serial_number2=B(:,2);
IDbox=B(:,4);
esito=B(:,6);
var1=B(:,7);
var2=B(:,8);
var3=B(:,9);
var4=B(:,10);
var5=B(:,11);
var6=B(:,12);
var7=B(:,16);
varT=B(:,15);
codice_fallo=B(:,18);
% SEPARATION BY IDBOXs
k1=0;
k2=0;
k3=0;
k4=0;
k5=0;
k6=0;
k7=0;
k8=0;
k9=0;
k10=0;
```

```
k11=0;
k12=0;
k13=0;
k14=0;
k15=0;
k16=0;


for i=1:length(B)
    if IDbox(i)==1
        k1=k1+1;
        ID1(k1,:)=B(i,:);
        refr1(k1)=r(i);
        posID1(k1)=i;
    elseif IDbox(i)==2
        k2=k2+1;
        ID2(k2,:)=B(i,:);
        refr2(k2)=r(i);
        posID2(k2)=i;
    elseif IDbox(i)==3
        k3=k3+1;
        ID3(k3,:)=B(i,:);
        refr3(k3)=r(i);
        posID3(k3)=i;
    elseif IDbox(i)==4
        k4=k4+1;
        ID4(k4,:)=B(i,:);
        refr4(k4)=r(i);
        posID4(k4)=i;
    elseif IDbox(i)==5
        k5=k5+1;
        ID5(k5,:)=B(i,:);
        refr5(k5)=r(i);
        posID5(k5)=i;
    elseif IDbox(i)==6
        k6=k6+1;
        ID6(k6,:)=B(i,:);
        refr6(k6)=r(i);
        posID6(k6)=i;
    elseif IDbox(i)==7
        k7=k7+1;
        ID7(k7,:)=B(i,:);
        refr7(k7)=r(i);
        posID7(k7)=i;
    elseif IDbox(i)==8
        k8=k8+1;
        ID8(k8,:)=B(i,:);
        refr8(k8)=r(i);
        posID8(k8)=i;
    elseif IDbox(i)==9
        k9=k9+1;
        ID9(k9,:)=B(i,:);
        refr9(k9)=r(i);
```

```
        posID9(k9)=i;
    elseif IDbox(i)==10
        k10=k10+1;
        ID10(k10,:)=B(i,:);
        refr10(k10)=r(i);
        posID10(k10)=i;
    elseif IDbox(i)==11
        k11=k11+1;
        ID11(k11,:)=B(i,:);
        refr11(k11)=r(i);
        posID11(k11)=i;
    elseif IDbox(i)==12
        k12=k12+1;
        ID12(k12,:)=B(i,:);
        refr12(k12)=r(i);
        posID12(k12)=i;
    elseif IDbox(i)==13
        k13=k13+1;
        ID13(k13,:)=B(i,:);
        refr13(k13)=r(i);
        posID13(k13)=i;
    elseif IDbox(i)==14
        k14=k14+1;
        ID14(k14,:)=B(i,:);
        refr14(k14)=r(i);
        posID14(k14)=i;
    elseif IDbox(i)==15
        k15=k15+1;
        ID15(k15,:)=B(i,:);
        refr15(k15)=r(i);
        posID15(k15)=i;
     elseif IDbox(i)==16
        k16=k16+1;
        ID16(k16,:)=B(i,:);
        refr16(k16)=r(i);
        posID16(k16)=i;
    end
end
%
k=0;
for i=1:length(B(:,2))-1
    if B(i,2)~=B(i+1,2)
        k=k+1;
        dat(k)=B(i,2);
        k=k+1;
        dat(k)=B(i+1,2);
    end
end

dat2(1)=dat(1);
for i=1:length(dat)
    n=0;
    j=0;
```

```
    i
    while j<length(dat2)
        j=j+1;
        if dat(i)~=dat2(j)
            n=n+1;
        end
        if n==length(dat2)
            dat2(j+1)=dat(i);
        end
    end
end
%%
%%
fault_code=2

NC=dat2;
%%
%% RIPARAZIONE
frigus1_1=[513 604 1130 58500 69677 78215 94724 96715 97195 112719 113209↙
113564 114731 120845 126431 168883 170397];
frigus1_2=[512 603 1131 58499 69676 78214 94725 96714 97196 112718 113210↙
113565 114730 120844 126432 168884 170396];

frigus2_3=[39432 77325 94723 105564 113722 115094];
frigus2_4=[39431 77324 94722 105563 113723 115093];

frigus3_5=[58866 61340 62342 63297 87233 89087 90534 110785 111223 111611↙
111802 124942 141321 141548 166319 166694];
frigus3_6=[58867 61339 62341 63298 87234 89086 90533 110786 111224 111612↙
111803 124941 141322 141549 166318 166695];

frigus4_7=[4470 44231 59050 107295 109057 139850 140378 141324];
frigus4_8=[4469 44230 59049 107296 109058 139849 140379 141323];

frigus5_9=[9085 9522 18615 37052 47927 81396 84846 112583];
frigus5_10=[9086 9521 18614 37051 47926 81411 84845 112582];

frigus6_11=[5577 52356 56020 71473 82873 109184 167177 175808];
frigus6_12=[5576 52355 56021 71474 82874 109185 167179 175809];

frigus7_13=[65273 70657 75983 76318 101896 103437 108936 152271];
frigus7_14=[65274 70658 75982 76317 101895 103438 108935 152270];

frigus8_15=[35476 44506 76969 113221 160391];
frigus8_16=[35477 44505 76970 113222 160390];
%% START
statisticated=0;
clear frigus
clear refrigerante
clear posID
help_variable1=0;
plotting=0;
for x=1:16
```

```
    x
    clear ID % data deleted by this is for all types of faults
    if x==1
        ID=ID1;
        posID=posID1;
        frigus=frigus1_1-1;
        refrigerante=refr1;
                m=length(frigus);
                for i=1:length(frigus)
                    p=m+1-i;
%                     if p==11 || %↙
p==1||p==3||p==2||p==4||p==7||p==9||p==12||p==13||p==14||p==16   ||      %↙
posiciones a eliminar de frigus
%                     frigus(p)=[];  %si p=11 solo afecta a la pistola 2
%                     end
                 end

    elseif x==2
        ID=ID2;
        posID=posID2;
        frigus=frigus1_2-1;
        refrigerante=refr2;
                m=length(frigus);
                for i=1:length(frigus)
                    p=m+1-i;
%                     if p==6||p==15||p==17%↙
p==1||p==3||p==2||p==4||p==7||p==9||p==12||p==13||p==14||p==16    ||↙
posiciones a eliminar de frigus
%                     frigus(p)=[];% P=6,15,17 solo afecta a pistola 1
%                     end
                 end

    elseif x==3
        ID=ID3;
        posID=posID3;
        frigus=frigus2_3-1;
        refrigerante=refr3;
                m=length(frigus);
                for i=1:length(frigus)
                    p=m+1-i;
                    %if p==1||p==2||p==5||p==6%posiciones a eliminar de↙
frigus
                    %frigus(p)=[];
                    %end
                 end
    elseif x==4
        ID=ID4;
        posID=posID4;
        refrigerante=refr4;
        frigus=frigus2_4-1;
                m=length(frigus);
                for i=1:length(frigus)
                    p=m+1-i;
```

```
                    %if p==1||p==2||p==5||p==6%posiciones a eliminar de↙
frigus
                    %frigus(p)=[];
                    %end
                 end
    elseif x==5
        ID=ID5;
        posID=posID5;
        refrigerante=refr5;
        frigus=frigus3_5-1;
                m=length(frigus);
                for i=1:length(frigus)
                    p=m+1-i;
%                     if p==8||p==9||p==15  %↙
p==1||p==3||p==4||p==7||p==11||p==12||p==14||p==16     %  posiciones a↙
eliminar de frigus
%                     frigus(p)=[];
%                     end
                 end
    elseif x==6
        ID=ID6;
        posID=posID6;
        refrigerante=refr6;
        frigus=frigus3_6-1;
                m=length(frigus);
                for i=1:length(frigus)
                    p=m+1-i;
%                     if p==13%↙
p==1||p==3||p==4||p==7||p==11||p==12||p==14||p==16     %  || posiciones a↙
eliminar de frigus
%                     frigus(p)=[];
%                     end
                 end
    elseif x==7
        ID=ID7;
        posID=posID7;
        refrigerante=refr7;
        frigus=frigus4_7-1;
                m=length(frigus);
                for i=1:length(frigus)
                    p=m+1-i;
%                     if p==4||p==6% p==1||p==3    %   ||posiciones a↙
eliminar de frigus
%                     frigus(p)=[];
%                     end
                 end
    elseif x==8
        ID=ID8;
        posID=posID8;
        refrigerante=refr8;
        frigus=frigus4_8-1;
                m=length(frigus);
                for i=1:length(frigus)
```

```
                    p=m+1-i;
%                     if p==5%p==1||p==3     % ||posiciones a eliminar de↙
frigus
%                     frigus(p)=[];
%                     end
                 end
    elseif x==9
        ID=ID9;
        posID=posID9;
        frigus=frigus5_9-1;
        refrigerante=refr9;
                m=length(frigus);
                for i=1:length(frigus)
                    p=m+1-i;
%                     if p==7%  p==1     % ||posiciones a eliminar de↙
frigus
%                     frigus(p)=[];
%                     end
                 end
    elseif x==10
        ID=ID10;
        posID=posID10;
        frigus=frigus5_10-1;
        refrigerante=refr10;
                m=length(frigus);
                for i=1:length(frigus)
                    p=m+1-i;
%                     if p==5||p==6||p==8 %  p==1    %   ||posiciones a↙
eliminar de frigus
%                     frigus(p)=[];
%                     end
                 end
    elseif x==11
        ID=ID11;
        posID=posID11;
        frigus=frigus6_11-1;
        refrigerante=refr11;
                m=length(frigus);
                for i=1:length(frigus)
                    p=m+1-i;
%                     if p==4||p==8 %posiciones a eliminar de frigus
%                     frigus(p)=[];
%                     end
                 end
    elseif x==12
        ID=ID12;
        posID=posID12;
        frigus=frigus6_12-1;
        refrigerante=refr12;
                m=length(frigus);
                for i=1:length(frigus)
                    p=m+1-i;
%                     if p==6||p==7||p==8 %posiciones a eliminar de frigus
```

```
%                     frigus(p)=[];
%                     end
                 end
    elseif x==13
        ID=ID13;
        posID=posID13;
        refrigerante=refr13;
        frigus=frigus7_13-1;
                 m=length(frigus);
                for i=1:length(frigus)
                    p=m+1-i;
%                     if p==1||p==2||p==7||p==4 %posiciones a eliminar de
frigus
%                     frigus(p)=[];
%                     end
                 end
    elseif x==14
        ID=ID14;
        posID=posID14;
        refrigerante=refr14;
        frigus=frigus7_14-1;
                m=length(frigus);
                for i=1:length(frigus)
                    p=m+1-i;
%                     if p==1||p==2||p==7||p==4 %posiciones a eliminar de
frigus
%                     frigus(p)=[];
%                     end
                 end
    elseif x==15
        ID=ID15;
        posID=posID15;
        refrigerante=refr15;
        frigus=frigus8_15-1;
                m=length(frigus);
                for i=1:length(frigus)
                    p=m+1-i;
%                     if p==1 % p==1posiciones a eliminar de frigus
%                     frigus(p)=[];
%                     end
                 end
    elseif x==16
        ID=ID16;
        posID=posID16;
        refrigerante=refr16;
        frigus=frigus8_16-1;
                m=length(frigus);
                for i=1:length(frigus)
                    p=m+1-i;
%                     if p==1       ||p==4||p==5 % p==1       ||posiciones
a eliminar de frigus
%                     frigus(p)=[];
%                     end
```

```
            end
    end

clear T1 T2 T3 T4 T5 T6 TT T8
clear array_data_1
clear media media1
clear faults
clear fault_type
clear faults_overall_position
clear pos_fallo_diferente
k=0;
m=0;
psd=0;
k_posf=0;
faults=0;
for i=1:length(ID)
    if ID(i,18)==fault_code || ID(i,18)==0
        k=k+1;
        fault_type(k)=ID(i,18); %fault_type IS A VECTOR OF FAULTS DECIDED↙
TO WITH FAULT CODE ONLY AND NO FAULTS
        % T IS THE FINAL MATRIX OF VARIABLES FOR EVERY SITUATION
        T1(k)=ID(i,7);
        T2(k)=ID(i,8);
        T3(k)=ID(i,9);
        T4(k)=ID(i,10);
        T5(k)=ID(i,11);
        T6(k)=ID(i,12);
        TT(k)=ID(i,15);
        T8(k)=ID(i,16);
        pos(x,k)=posID(i); %ANCESTRAL POSITION OF DATA IN THE INITIAL DATA↙
PLACE

        if ID(i,18)==fault_code
        k_posf=k_posf+1;
        faults_overall_position(k_posf)=posID(i);
        end

    else %IF IS NOT AN ERROR OF THE ONES WE ARE LOOKING FOR SO IT DOES NOT↙
HAS AN INFLUENCE
        k=k+1;
        fault_type(k)=-1;
        psd=psd+1;
        pos_fallo_diferente(psd)=posID(i);
        pos(x,k)=posID(i);
        if k>1
        T1(k)=T1(k-1);
        T2(k)=T1(k-1);
        T3(k)=T1(k-1);
        T4(k)=T1(k-1);
        T5(k)=T1(k-1);
        T6(k)=T1(k-1);
        TT(k)=T1(k-1);
        T8(k)=T1(k-1);
```

```
        end
        if k==1
        T1(k)=0;
        T2(k)=0;
        T3(k)=0;
        T4(k)=0;
        T5(k)=0;
        T6(k)=0;
        TT(k)=0;
        T8(k)=0;
        end

    end
end

T=[T1' T2' T3' T4' T5' T6' TT' T8'];
%timming2=toc

%%
k=0;
%%
for i=1:length(fault_type)
    if fault_type(i)==fault_code;
        k=k+1;
        faults(1,k)=ID(i,18);
        faults(2,k)=i;
    end
end
if faults~=0
help_variable1=length(faults)+help_variable1;
             porcentaje_error=length(faults)/length(T)*100
[m,nT]=size(T);
%% ARRAY_DATA is the most importan variable in the code together with the
array faults
for n=1:nT %array_data_1 has all the infrmation of the measures, all the
data already cleaned and processes
    for k=1:length(T) % faults contains for each type of fault and Idbox,
the instants where fault is detected
            array_data_1(k,n)=T(k,n);
    end
end
%%
i=0;
%%
[nn mm]=size(array_data_1);
media=zeros(nn,mm-1);
media1=zeros(nn,mm-1);
   for i=1:length(array_data_1)
       for j=1:length(faults(1,:))
           if faults(2,j)==i
               if x==2 && i==1
               array_data_1(i,:)=0;
               else
```

```
            array_data_1(i,:)=media1(i-1,:);
               end
            end
        end
         for n=1:nT
             if i==1
                 media(i,n)=array_data_1(i,n);
             elseif i>1 && i<=100
                 media(i,n)=((media(i-1,n)*(i-1))+array_data_1(i,n))/i;
             elseif i>10
                 media(i,n)=(sum(array_data_1(i-100:i,n)))/100;
             end

             if i==1
                 media1(i,n)=array_data_1(i,n);
             elseif i>1 && i<=100
                 media1(i,n)=((media1(i-1,n)*(i-1))+media(i,n))/i;
             elseif i>10
                 media1(i,n)=(sum(media(i-100:i,n)))/100; %poner por 10↙
para ver mejor
             end
         end
    end
%%
%% IS A SECTION TO GRAPH ALL THE DIFFERENT INTERVALS BETWEEN 2 CONSECUTIVE↙
FAILURES
    uta=0;
    for i=2:50
        inicio=fallos(2,i-1)+1;
        final=fallos(2,i);
        uta=uta+1;
        figure(uta)
        for n=1:8
            plot(inicio:final,array_datos_1(inicio:final,n))
            hold on
        end
        hold off
    end
%%
for work=1
for lavoro_del_programa=1
plotting=1;
clear C12NC
C12NC=ID(1,2);
for i=1:length(ID)
    count=0;
    for j=1:length(C12NC)
        if ID(i,2)~=C12NC(j)
            count=count+1;
        end
    end
    if count==length(C12NC)
        C12NC(length(C12NC)+1)=ID(i,2);
```

```
    end
end


%%
%clear estadistica
clear a_datos_NCs
clear ID_soloaqui
tt=0;
cantidad_c12nc=length(C12NC)
for i=1:length(C12NC)
    cc=0;
    t=0;
    for j=1:length(ID(:,2))
        if C12NC(i)==ID(j,2)
            for k=1:length(pos_fallo_diferente)
                if pos(x,j)==pos_fallo_diferente(k)
                    t=1;
                end
            end
            if t~=1
            cc=cc+1;
            ID_soloaqui(j,:)=[array_data_1(j,:) pos(x,j)];
            a_datos_NCs(cc,:,i)=[ID_soloaqui(j,:)]; %a_ID_NCs significa↙
que es un ARRAY de las filas del vector ID
            %dividido en los diferentes valores de NC en orden de↙
actuacion
            %pero separado en bloques de NC
            else
            end
        end
    end
end

aa=0;
for i=1:length(C12NC)
    if statisticated==0
        estadistica(1,i)=1;
    end
    if estadistica(1,i)~=0
    f=0;
    l=length(a_datos_NCs(:,9,i));
    for j=1:length(a_datos_NCs(:,9,i))
        if a_datos_NCs(j,9,i)==0
        l=j-1;
        break;
        end
    end

    if l>10
    %figure(i)
    for j2=1:l
        for k2=1:length(faults_overall_position)
```

```
            if a_datos_NCs(j2,9,i)==faults_overall_position(k2)
                f=f+1;
                %plot(length(a_datos_NCs(1:j2,:,i)),0,'r*')
                hold on
            end
        end
        for k3=1:length(frigus)
            if a_datos_NCs(j2,9,i)==frigus(k3)
                %plot(j2:j2+1,0:900:900,'k');
                aa=aa+1;
                i;
                %hold on
            end
        end
    end
    for n=1:7
    %plot(1:l,a_datos_NCs(1:l,n,i))
   % hold on
    end
   % hold off
    end
    if statisticated==0
estadistica(1,i)=f/l; %correr primero esto y luego todo lo demas ya que
%estadisitica existe antes, pero es necesario ponerlo aqui
estadistica(2,i)=l;
    end
    end
end
statisticated=1;
end
%%
for lavoro_del_programa=1
plotting=1;
clear C12NC
C12NC=ID(1,2);
for i=1:length(ID)
    count=0;
    for j=1:length(C12NC)
        if ID(i,2)~=C12NC(j)
            count=count+1;
        end
    end
    if count==length(C12NC)
        C12NC(length(C12NC)+1)=ID(i,2);
    end
end


%%
%clear estadistica
clear a_datos_NCs
clear ID_soloaqui
tt=0;
```

```
for i=1:length(C12NC)
    cc=0;
    t=0;
    for j=1:length(ID(:,2))
        if C12NC(i)==ID(j,2)
            for k=1:length(pos_fallo_diferente)
                if pos(x,j)==pos_fallo_diferente(k)
                    t=1;
                end
            end
            if t~=1
            cc=cc+1;
            ID_soloaqui(j,:)=[array_data_1(j,:) pos(x,j)];
            a_datos_NCs(cc,:,i)=[ID_soloaqui(j,:)]; %a_ID_NCs significa↙
que es un ARRAY de las filas del vector ID
            %dividido en los diferentes valores de NC en orden de↙
actuacion
            %pero separado en bloques de NC
            else
            end
        end
    end
end

aa=0;
for i=1:length(C12NC)
    if statisticated==0
        estadistica(1,i)=1;
    end
    if estadistica(1,i)~=0
    f=0;
    l=length(a_datos_NCs(:,9,i));
    for j=1:length(a_datos_NCs(:,9,i))
        if a_datos_NCs(j,9,i)==0
        l=j-1;
        break;
        end
    end

    if l>10
    %figure(i)
    for j2=1:l
        for k2=1:length(faults_overall_position)
            if a_datos_NCs(j2,9,i)==faults_overall_position(k2)
                f=f+1;
                %plot(length(a_datos_NCs(1:j2,:,i)),0,'r*')
                hold on
            end
        end
        for k3=1:length(frigus)
            if a_datos_NCs(j2,9,i)==frigus(k3)
                %plot(j2:j2+1,0:900:900,'k');
                aa=aa+1;
```

```
                i;
                %hold on
            end
        end
    end
    for n=1:7
    %plot(1:l,a_datos_NCs(1:l,n,i))
   % hold on
    end
   % hold off
    end
    if statisticated==0
estadistica(1,i)=f/l; %correr primero esto y luego todo lo demas ya que
%estadisitica existe antes, pero es necesario ponerlo aqui
estadistica(2,i)=l;
    end
    end
end
statisticated=1;
end
%%
for estadisticas_err=1
statisticated=0;
m_e=0;
oro=0;
no_zero=0;
no_zero_v=0;
kj=0;
kjj=0;

%figure(1000*x+12)

for i=1:length(estadistica)
    if estadistica(2,i)>10
        if estadistica(1,i)~=0
       %plot(length(estadistica(1,1:i)),estadistica(1,i),'r*')
        no_zero=no_zero+1;
        no_zero_v=no_zero_v+estadistica(1,i);
        end
        m_e=m_e+estadistica(1,i);
        oro=oro+1;
        if estadistica(1,i)>=0.05
            kjj=kjj+1;
            no_sirven_out(kjj)=i;
        end
    %hold on
    else
        kj=kj+1;
        no_sirven_small(kj)=i;
    end
end
hold on
%m_e2=m_e/length(estadistica);
```

```
m_e=m_e/oro;
vf=ones(1,length(estadistica));
m_e=m_e*vf;
max_m=0.05*vf;
%m_e2=m_e2*vf;
plot(1:length(estadistica),m_e)
hold on
%plot(1:length(estadistica),m_e2)
%plot(1:length(estadistica),max_m,'r')
%hold off

k=0;
for i=1:length(estadistica)
    if estadistica(1,i)>=m_e(1)*8+0.1*m_e(1) || estadistica(2,i)<=5
        k=k+1;
        no_sirve(k)=i;
    end
end
max_m=(m_e(1)*8+0.1*m_e(1))*vf;
plot(1:length(estadistica),max_m,'r')
hold off
k1=0;
k2=0;
clear dato_si
clear dato_no
for i=1:length(C12NC)
      l=length(a_datos_NCs(:,9,i));
    for j=1:length(a_datos_NCs(:,9,i))
        if a_datos_NCs(j,9,i)==0
        l=j-1;
        break;
        end
    end
    on=0;
    for k=1:length(no_sirve)
        if i==no_sirve(k)
            for ll=1:l
                k1=k1+1;
                dato_no(k1)=a_datos_NCs(ll,9,i);
            end
        end
    end
     if on==0;
            for ll=1:l
                k2=k2+1;
                dato_si(k2)=a_datos_NCs(ll,9,i);
            end
     end

end
end
end
```

```
for manu=1 %this secction studies the importance of the refrigerant in the↙
results
% %%
% j=0;
% for i=1:length(refrigerante)
%     if refrigerante(i)==134
%         j=j+1;
%         R134a(j)=i;
%     end
% end
% r134=0;
% r600=0;
% for i=1:length(R134a)
%     for j=1:length(faults(2,:))
%     if R134a(i)==faults(2,j)
%         r134=r134+1;
%     else
%         r600=r600+1;
%     end
%     end
% end
% r1=0;
% r6=0;
% for i=1:length(refrigerante)
%     if refrigerante(i)==134
%         r1=r1+1;
%     else
%         r6=r6+1;
%     end
% end
% RRR(1,x)=r134;
% RRR(2,x)=r600;
% RRR(3,x)=length(faults(2,:));
% RRR(4,x)=r1;
% RRR(5,x)=r6;
% x
% end
% RRR'
% RRR(1,:)./RRR(4,:);
end %nothing useful, only things of the refrigerant
%%
if plotting==1
figure(x)
title('1')
for n=1:nT-1 %until 8 variables beacuse the 9th variable is strange and↙
does not hasve any influence on the system
    plot(1:length(array_data_1),array_data_1(:,n));
    hold on
end
 for n=1:nT-1
     plot(media1(:,n),'r')
     hold on
```

```
  end
 end
  k=0;
  if plotting==0
  for i=1:length(faults)
      k=k+1;
      %plot(faults(2,k),0,'r*')
      %hold on
  end
  end
  %
 if plotting==1
   for i=1:length(faults(2,:))
    k=k+1;
    no_esta=0;
%       plot(faults(2,k),0,'r*')
%       hold on
      for xjx=1:length(dato_no)
          if faults_overall_position(k)==dato_no(xjx)
              plot(faults(2,k),0,'r*') %esto es para marcar los no_faults
              hold on
          else
              no_esta=no_esta+1;
          end
      end
      if no_esta==length(dato_no)
          plot(faults(2,k),0,'r*')
          hold on
      end
   end
  end
  %%
  m=0;
  if plotting==1
  for i=1:length(pos(x,:))
      for j=1:length(frigus)
          if pos(x,i)==frigus(j)
              m=m+1;
              j;
              plot(i:i+1,0:5000:5000,'k')
              hold on
          end
      end
  end
 legend('1 VALORE VUOTO INI','2 VALORE VUOTO EVACUAZIONE','3 VALORE VUOTO↙
 PROVA PERDITE','4 PRESSIONE MISCELAZIONE','5 VALORE EVACUAZIONE POST↙
 MINI','6 VALORE VUTO PERDITE','TEMPERATURA')%,'7 PRES. REFR')
 %hold of
  end
 %%
 %% DENSITY MA(1000)
 clear densidad
 faults_densidades=faults(2,:);
```

```
hold on
[long, var]=size(array_data_1);
in=1;
fin_in=1000;
for k=0:long-fin_in;
    in=1+k;
    fin=fin_in+k;
cont=0;
for i=in:fin
    j=0;
    for j=1:length(faults_densidades)
        if faults_densidades(j)==i
            cont=cont+1;
            break;
        end
    end
end
densidad(:,k+fin_in)=[fin;cont*200];
end

plot(1:length(array_data_1),densidad(2,:),'r')
hold on
%% THRESHOLDS
limit=ones(length(array_data_1),1)*3000;
plot(1:length(array_data_1),limit,'k')
hold on
limit=ones(length(array_data_1),1)*4000;
plot(1:length(array_data_1),limit,'b')
hold off
end
end
```

ANNEX V - NEURAL NETWORK SCRIPT


```
% Solve a Pattern Recognition Problem with a Neural Network
%
% This script assumes these variables are defined:
%
%   matrix - input data.
%   fallo - target data.

x = matrix;
t = fallo;

% Create a Pattern Recognition Network
hiddenLayerSize = 10;
net = patternnet(hiddenLayerSize);


% Setup Division of Data for Training, Validation, Testing
net.divideParam.trainRatio = 70/100;
net.divideParam.valRatio = 15/100;
net.divideParam.testRatio = 15/100;


% Train the Network
[net,tr] = train(net,x,t);

% Test the Network
y = net(x);
e = gsubtract(t,y);
tind = vec2ind(t);
yind = vec2ind(y);
percentErrors = sum(tind ~= yind)/numel(tind);
performance = perform(net,t,y)

% View the Network
view(net)

% Plots
% Uncomment these lines to enable various plots.
%figure, plotperform(tr)
%figure, plottrainstate(tr)
%figure, plotconfusion(t,y)
%figure, plotroc(t,y)
%figure, ploterrhist(e)
```

ANNEX VI - INTERFACE SCRIPT


```
function varargout = prueba1_gui(varargin)
%PRUEBA1_GUI M-file for prueba1_gui.fig
%      PRUEBA1_GUI, by itself, creates a new PRUEBA1_GUI or raises the
existing
%      singleton*.
%
%      H = PRUEBA1_GUI returns the handle to a new PRUEBA1_GUI or the
handle to
%      the existing singleton*.
%
%      PRUEBA1_GUI('Property','Value',...) creates a new PRUEBA1_GUI using
the
%      given property value pairs. Unrecognized properties are passed via
%      varargin to prueba1_gui_OpeningFcn.  This calling syntax produces a
%      warning when there is an existing singleton*.
%
%      PRUEBA1_GUI('CALLBACK') and PRUEBA1_GUI('CALLBACK',hObject,...)
call the
%      local function named CALLBACK in PRUEBA1_GUI.M with the given input
%      arguments.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only
one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help prueba1_gui

% Last Modified by GUIDE v2.5 01-Feb-2017 00:20:31

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @prueba1_gui_OpeningFcn, ...
                   'gui_OutputFcn',  @prueba1_gui_OutputFcn, ...
                   'gui_LayoutFcn',  [], ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
   gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT


% --- Executes just before prueba1_gui is made visible.
```

```
function prueba1_gui_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   unrecognized PropertyName/PropertyValue pairs from the
%            command line (see VARARGIN)

% Choose default command line output for prueba1_gui
handles.output = hObject;
%% INITIALITATION
for box=1:16
    if box==1
    file=fopen('IDBOX1.txt','r+');
    vector=fscanf(file,'%d');
elseif box==2
    file=fopen('IDBOX2.txt','r+');
    vector=fscanf(file,'%d');
elseif box==3
    file=fopen('IDBOX3.txt','r+');
    vector=fscanf(file,'%d');
elseif box==4
    file=fopen('IDBOX4.txt','r+');
    vector=fscanf(file,'%d');
elseif box==5
    file=fopen('IDBOX5.txt','r+');
    vector=fscanf(file,'%d');
elseif box==6
    file=fopen('IDBOX6.txt','r+');
    vector=fscanf(file,'%d');
elseif box==7
    file=fopen('IDBOX7.txt','r+');
    vector=fscanf(file,'%d');
elseif box==8
    file=fopen('IDBOX8.txt','r+');
    vector=fscanf(file,'%d');
elseif box==9
    file=fopen('IDBOX9.txt','r+');
    vector=fscanf(file,'%d');
elseif box==10
    file=fopen('IDBOX10.txt','r+');
    vector=fscanf(file,'%d');
elseif box==11
    file=fopen('IDBOX11.txt','r+');
    vector=fscanf(file,'%d');
elseif box==12
    file=fopen('IDBOX12.txt','r+');
    vector=fscanf(file,'%d');
elseif box==13
    file=fopen('IDBOX13.txt','r+');
    vector=fscanf(file,'%d');
elseif box==14
    file=fopen('IDBOX14.txt','r+');
```

```
    vector=fscanf(file,'%d');
elseif box==15
    file=fopen('IDBOX15.txt','r+');
    vector=fscanf(file,'%d');
elseif box==16
    file=fopen('IDBOX16.txt','r+');
    vector=fscanf(file,'%d');
    end
%
if length(vector)<=30
    m=0;
    for i=1:length(vector)
        if vector(i)==1;
            m=m+1;
        end
    end
    if m>=10
       for ok=1
       if box==1
       set(handles.p1,'BackgroundColor','red');
       end
       if box==2
       set(handles.p2,'BackgroundColor','red');
       end
       if box==3
       set(handles.p3,'BackgroundColor','red');
       end
       if box==4
       set(handles.p4,'BackgroundColor','red');
       end
       if box==5
       set(handles.p5,'BackgroundColor','red');
       end
       if box==6
       set(handles.p6,'BackgroundColor','red');
       end
       if box==7
       set(handles.p7,'BackgroundColor','red');
       end
       if box==8
       set(handles.p8,'BackgroundColor','red');
       end
       if box==9
       set(handles.p9,'BackgroundColor','red');
       end
       if box==10
       set(handles.p10,'BackgroundColor','red');
       end
       if box==11
       set(handles.p11,'BackgroundColor','red');
       end
       if box==12
       set(handles.p12,'BackgroundColor','red');
```

```
        end
        if box==13
        set(handles.p13,'BackgroundColor','red');
        end
        if box==14
        set(handles.p14,'BackgroundColor','red');
        end
        if box==15
        set(handles.p15,'BackgroundColor','red');
        end
        if box==16
        set(handles.p16,'BackgroundColor','red');
        end
        end % es todo el cambio de color ROJO
    elseif m>=5
        for ok=1
        if box==1
        set(handles.p1,'BackgroundColor','yellow');
        end
        if box==2
        set(handles.p2,'BackgroundColor','yellow');
        end
        if box==3
        set(handles.p3,'BackgroundColor','yellow');
        end
        if box==4
        set(handles.p4,'BackgroundColor','yellow');
        end
        if box==5
        set(handles.p5,'BackgroundColor','yellow');
        end
        if box==6
        set(handles.p6,'BackgroundColor','yellow');
        end
        if box==7
        set(handles.p7,'BackgroundColor','yellow');
        end
        if box==8
        set(handles.p8,'BackgroundColor','yellow');
        end
        if box==9
        set(handles.p9,'BackgroundColor','yellow');
        end
        if box==10
        set(handles.p10,'BackgroundColor','yellow');
        end
        if box==11
        set(handles.p11,'BackgroundColor','yellow');
        end
        if box==12
        set(handles.p12,'BackgroundColor','yellow');
        end
        if box==13
```

```
        set(handles.p13,'BackgroundColor','yellow');
        end
        if box==14
        set(handles.p14,'BackgroundColor','yellow');
        end
        if box==15
        set(handles.p15,'BackgroundColor','yellow');
        end
        if box==16
        set(handles.p16,'BackgroundColor','yellow');
        end
        end % es todo el cambio de color AMARILLO
    else
          for ok=1
        if box==1
        set(handles.p1,'BackgroundColor','green');
        end
        if box==2
        set(handles.p2,'BackgroundColor','green');
        end
        if box==3
        set(handles.p3,'BackgroundColor','green');
        end
        if box==4
        set(handles.p4,'BackgroundColor','green');
        end
        if box==5
        set(handles.p5,'BackgroundColor','green');
        end
        if box==6
        set(handles.p6,'BackgroundColor','green');
        end
        if box==7
        set(handles.p7,'BackgroundColor','green');
        end
        if box==8
        set(handles.p8,'BackgroundColor','green');
        end
        if box==9
        set(handles.p9,'BackgroundColor','green');
        end
        if box==10
        set(handles.p10,'BackgroundColor','green');
        end
        if box==11
        set(handles.p11,'BackgroundColor','green');
        end
        if box==12
        set(handles.p12,'BackgroundColor','green');
        end
        if box==13
        set(handles.p13,'BackgroundColor','green');
        end
```

```
        if box==14
        set(handles.p14,'BackgroundColor','green');
        end
        if box==15
        set(handles.p15,'BackgroundColor','green');
        end
        if box==16
        set(handles.p16,'BackgroundColor','green');
        end
        end % es todo el cambio de color VERDE
    end
elseif length(vector)<=1000
    m=0;
    for i=1:length(vector)
         if vector(i)==1
             m=m+1;
         end
    end
    if m>=20
        for ok=1
        if box==1
        set(handles.p1,'BackgroundColor','red');
        end
        if box==2
        set(handles.p2,'BackgroundColor','red');
        end
        if box==3
        set(handles.p3,'BackgroundColor','red');
        end
        if box==4
        set(handles.p4,'BackgroundColor','red');
        end
        if box==5
        set(handles.p5,'BackgroundColor','red');
        end
        if box==6
        set(handles.p6,'BackgroundColor','red');
        end
        if box==7
        set(handles.p7,'BackgroundColor','red');
        end
        if box==8
        set(handles.p8,'BackgroundColor','red');
        end
        if box==9
        set(handles.p9,'BackgroundColor','red');
        end
        if box==10
        set(handles.p10,'BackgroundColor','red');
        end
        if box==11
        set(handles.p11,'BackgroundColor','red');
        end
```

```
        if box==12
        set(handles.p12,'BackgroundColor','red');
        end
        if box==13
        set(handles.p13,'BackgroundColor','red');
        end
        if box==14
        set(handles.p14,'BackgroundColor','red');
        end
        if box==15
        set(handles.p15,'BackgroundColor','red');
        end
        if box==16
        set(handles.p16,'BackgroundColor','red');
        end
        end % es todo el cambio de color ROJO
    elseif m>=15
        for ok=1
        if box==1
        set(handles.p1,'BackgroundColor','yellow');
        end
        if box==2
        set(handles.p2,'BackgroundColor','yellow');
        end
        if box==3
        set(handles.p3,'BackgroundColor','yellow');
        end
        if box==4
        set(handles.p4,'BackgroundColor','yellow');
        end
        if box==5
        set(handles.p5,'BackgroundColor','yellow');
        end
        if box==6
        set(handles.p6,'BackgroundColor','yellow');
        end
        if box==7
        set(handles.p7,'BackgroundColor','yellow');
        end
        if box==8
        set(handles.p8,'BackgroundColor','yellow');
        end
        if box==9
        set(handles.p9,'BackgroundColor','yellow');
        end
        if box==10
        set(handles.p10,'BackgroundColor','yellow');
        end
        if box==11
        set(handles.p11,'BackgroundColor','yellow');
        end
        if box==12
        set(handles.p12,'BackgroundColor','yellow');
```

```
        end
        if box==13
        set(handles.p13,'BackgroundColor','yellow');
        end
        if box==14
        set(handles.p14,'BackgroundColor','yellow');
        end
        if box==15
        set(handles.p15,'BackgroundColor','yellow');
        end
        if box==16
        set(handles.p16,'BackgroundColor','yellow');
        end
        end % es todo el cambio de color AMARILLO
    else
         for ok=1
        if box==1
        set(handles.p1,'BackgroundColor','green');
        end
        if box==2
        set(handles.p2,'BackgroundColor','green');
        end
        if box==3
        set(handles.p3,'BackgroundColor','green');
        end
        if box==4
        set(handles.p4,'BackgroundColor','green');
        end
        if box==5
        set(handles.p5,'BackgroundColor','green');
        end
        if box==6
        set(handles.p6,'BackgroundColor','green');
        end
        if box==7
        set(handles.p7,'BackgroundColor','green');
        end
        if box==8
        set(handles.p8,'BackgroundColor','green');
        end
        if box==9
        set(handles.p9,'BackgroundColor','green');
        end
        if box==10
        set(handles.p10,'BackgroundColor','green');
        end
        if box==11
        set(handles.p11,'BackgroundColor','green');
        end
        if box==12
        set(handles.p12,'BackgroundColor','green');
        end
        if box==13
```

```
        set(handles.p13,'BackgroundColor','green');
        end
        if box==14
        set(handles.p14,'BackgroundColor','green');
        end
        if box==15
        set(handles.p15,'BackgroundColor','green');
        end
        if box==16
        set(handles.p16,'BackgroundColor','green');
        end
        end % es todo el cambio de color VERDE
    end
else
    m=0;
    for i=length(vector)-1000:length(vector)
        if vector(i)==1
            m=m+1;
        end
    end
    if m>=20
        for ok=1
        if box==1
        set(handles.p1,'BackgroundColor','red');
        end
        if box==2
        set(handles.p2,'BackgroundColor','red');
        end
        if box==3
        set(handles.p3,'BackgroundColor','red');
        end
        if box==4
        set(handles.p4,'BackgroundColor','red');
        end
        if box==5
        set(handles.p5,'BackgroundColor','red');
        end
        if box==6
        set(handles.p6,'BackgroundColor','red');
        end
        if box==7
        set(handles.p7,'BackgroundColor','red');
        end
        if box==8
        set(handles.p8,'BackgroundColor','red');
        end
        if box==9
        set(handles.p9,'BackgroundColor','red');
        end
        if box==10
        set(handles.p10,'BackgroundColor','red');
        end
        if box==11
```

```
        set(handles.p11,'BackgroundColor','red');
        end
        if box==12
        set(handles.p12,'BackgroundColor','red');
        end
        if box==13
        set(handles.p13,'BackgroundColor','red');
        end
        if box==14
        set(handles.p14,'BackgroundColor','red');
        end
        if box==15
        set(handles.p15,'BackgroundColor','red');
        end
        if box==16
        set(handles.p16,'BackgroundColor','red');
        end
        end % es todo el cambio de color ROJO
     elseif m>=15
        for ok=1
        if box==1
        set(handles.p1,'BackgroundColor','yellow');
        end
        if box==2
        set(handles.p2,'BackgroundColor','yellow');
        end
        if box==3
        set(handles.p3,'BackgroundColor','yellow');
        end
        if box==4
        set(handles.p4,'BackgroundColor','yellow');
        end
        if box==5
        set(handles.p5,'BackgroundColor','yellow');
        end
        if box==6
        set(handles.p6,'BackgroundColor','yellow');
        end
        if box==7
        set(handles.p7,'BackgroundColor','yellow');
        end
        if box==8
        set(handles.p8,'BackgroundColor','yellow');
        end
        if box==9
        set(handles.p9,'BackgroundColor','yellow');
        end
        if box==10
        set(handles.p10,'BackgroundColor','yellow');
        end
        if box==11
        set(handles.p11,'BackgroundColor','yellow');
        end
```

```
        if box==12
        set(handles.p12,'BackgroundColor','yellow');
        end
        if box==13
        set(handles.p13,'BackgroundColor','yellow');
        end
        if box==14
        set(handles.p14,'BackgroundColor','yellow');
        end
        if box==15
        set(handles.p15,'BackgroundColor','yellow');
        end
        if box==16
        set(handles.p16,'BackgroundColor','yellow');
        end
        end % es todo el cambio de color AMARILLO
    else
         for ok=1
        if box==1
        set(handles.p1,'BackgroundColor','green');
        end
        if box==2
        set(handles.p2,'BackgroundColor','green');
        end
        if box==3
        set(handles.p3,'BackgroundColor','green');
        end
        if box==4
        set(handles.p4,'BackgroundColor','green');
        end
        if box==5
        set(handles.p5,'BackgroundColor','green');
        end
        if box==6
        set(handles.p6,'BackgroundColor','green');
        end
        if box==7
        set(handles.p7,'BackgroundColor','green');
        end
        if box==8
        set(handles.p8,'BackgroundColor','green');
        end
        if box==9
        set(handles.p9,'BackgroundColor','green');
        end
        if box==10
        set(handles.p10,'BackgroundColor','green');
        end
        if box==11
        set(handles.p11,'BackgroundColor','green');
        end
        if box==12
        set(handles.p12,'BackgroundColor','green');
```

```
        end
        if box==13
        set(handles.p13,'BackgroundColor','green');
        end
        if box==14
        set(handles.p14,'BackgroundColor','green');
        end
        if box==15
        set(handles.p15,'BackgroundColor','green');
        end
        if box==16
        set(handles.p16,'BackgroundColor','green');
        end
        end % es todo el cambio de color VERDE
    end
end
 fclose(file);
end

%%
% Update handles structure
guidata(hObject, handles);

% UIWAIT makes prueba1_gui wait for user response (see UIRESUME)
% uiwait(handles.figure1);


% --- Outputs from this function are returned to the command line.
function varargout = prueba1_gui_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

%% HERE IS WHERE WE PUT WHAT WHEN PRESSING THE BUTTON SHOULD HAPPEN

% --- Executes on button press in p.
function p_Callback(hObject, eventdata, handles)
% hObject    handle to p (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
%ES LO QUE CAMBIA DE STRING INTRODUCIDO A NUMERO PARA PODER OPERAR EN
%CALCULOS
fallo=str2num(get(handles.edit1,'string'));
box=str2num(get(handles.edit2,'string'));
if (fallo~=0 && fallo~=1) || box<0 || box>16
   set(handles.text1,'string','ERROR, THE CODE TO BE INTRODUCED MUST BE↙
"1" FOR MACHINE FAIL OR "0" FOR MACHINE CORRECT WORKING AND IDBOX VALUE IS↙
BETWEEN 1 AND 16');
else
```

```
    set(handles.text1,'string','')
%selection of the working IDbox
if box==1
     file=fopen('IDBOX1.txt','r+');
     vector=fscanf(file,'%d');
elseif box==2
     file=fopen('IDBOX2.txt','r+');
     vector=fscanf(file,'%d');
elseif box==3
     file=fopen('IDBOX3.txt','r+');
     vector=fscanf(file,'%d');
elseif box==4
     file=fopen('IDBOX4.txt','r+');
     vector=fscanf(file,'%d');
elseif box==5
     file=fopen('IDBOX5.txt','r+');
     vector=fscanf(file,'%d');
elseif box==6
     file=fopen('IDBOX6.txt','r+');
     vector=fscanf(file,'%d');
elseif box==7
     file=fopen('IDBOX7.txt','r+');
     vector=fscanf(file,'%d');
elseif box==8
     file=fopen('IDBOX8.txt','r+');
     vector=fscanf(file,'%d');
elseif box==9
     file=fopen('IDBOX9.txt','r+');
     vector=fscanf(file,'%d');
elseif box==10
     file=fopen('IDBOX10.txt','r+');
     vector=fscanf(file,'%d');
elseif box==11
     file=fopen('IDBOX11.txt','r+');
     vector=fscanf(file,'%d');
elseif box==12
     file=fopen('IDBOX12.txt','r+');
     vector=fscanf(file,'%d');
elseif box==13
     file=fopen('IDBOX13.txt','r+');
     vector=fscanf(file,'%d');
elseif box==14
     file=fopen('IDBOX14.txt','r+');
     vector=fscanf(file,'%d');
elseif box==15
     file=fopen('IDBOX15.txt','r+');
     vector=fscanf(file,'%d');
elseif box==16
     file=fopen('IDBOX16.txt','r+');
     vector=fscanf(file,'%d');
end

%we add to the open file the new value
```

```
fprintf(file,'\n%d',fallo);
%we introduce the new value 0/1 to the vector
vector(length(vector)+1)=fallo;

if length(vector)<=30
    m=0;
    for i=1:length(vector)
        if vector(i)==1;
            m=m+1;
        end
    end
    if m>=10
       for ok=1
       if box==1
       set(handles.p1,'BackgroundColor','red');
       end
       if box==2
       set(handles.p2,'BackgroundColor','red');
       end
       if box==3
       set(handles.p3,'BackgroundColor','red');
       end
       if box==4
       set(handles.p4,'BackgroundColor','red');
       end
       if box==5
       set(handles.p5,'BackgroundColor','red');
       end
       if box==6
       set(handles.p6,'BackgroundColor','red');
       end
       if box==7
       set(handles.p7,'BackgroundColor','red');
       end
       if box==8
       set(handles.p8,'BackgroundColor','red');
       end
       if box==9
       set(handles.p9,'BackgroundColor','red');
       end
       if box==10
       set(handles.p10,'BackgroundColor','red');
       end
       if box==11
       set(handles.p11,'BackgroundColor','red');
       end
       if box==12
       set(handles.p12,'BackgroundColor','red');
       end
       if box==13
       set(handles.p13,'BackgroundColor','red');
       end
       if box==14
```

```
        set(handles.p14,'BackgroundColor','red');
        end
        if box==15
        set(handles.p15,'BackgroundColor','red');
        end
        if box==16
        set(handles.p16,'BackgroundColor','red');
        end
        end % es todo el cambio de color ROJO
     elseif m>=5
        for ok=1
        if box==1
        set(handles.p1,'BackgroundColor','yellow');
        end
        if box==2
        set(handles.p2,'BackgroundColor','yellow');
        end
        if box==3
        set(handles.p3,'BackgroundColor','yellow');
        end
        if box==4
        set(handles.p4,'BackgroundColor','yellow');
        end
        if box==5
        set(handles.p5,'BackgroundColor','yellow');
        end
        if box==6
        set(handles.p6,'BackgroundColor','yellow');
        end
        if box==7
        set(handles.p7,'BackgroundColor','yellow');
        end
        if box==8
        set(handles.p8,'BackgroundColor','yellow');
        end
        if box==9
        set(handles.p9,'BackgroundColor','yellow');
        end
        if box==10
        set(handles.p10,'BackgroundColor','yellow');
        end
        if box==11
        set(handles.p11,'BackgroundColor','yellow');
        end
        if box==12
        set(handles.p12,'BackgroundColor','yellow');
        end
        if box==13
        set(handles.p13,'BackgroundColor','yellow');
        end
        if box==14
        set(handles.p14,'BackgroundColor','yellow');
        end
```

```
        if box==15
        set(handles.p15,'BackgroundColor','yellow');
        end
        if box==16
        set(handles.p16,'BackgroundColor','yellow');
        end
        end % es todo el cambio de color AMARILLO
    else
          for ok=1
        if box==1
        set(handles.p1,'BackgroundColor','green');
        end
        if box==2
        set(handles.p2,'BackgroundColor','green');
        end
        if box==3
        set(handles.p3,'BackgroundColor','green');
        end
        if box==4
        set(handles.p4,'BackgroundColor','green');
        end
        if box==5
        set(handles.p5,'BackgroundColor','green');
        end
        if box==6
        set(handles.p6,'BackgroundColor','green');
        end
        if box==7
        set(handles.p7,'BackgroundColor','green');
        end
        if box==8
        set(handles.p8,'BackgroundColor','green');
        end
        if box==9
        set(handles.p9,'BackgroundColor','green');
        end
        if box==10
        set(handles.p10,'BackgroundColor','green');
        end
        if box==11
        set(handles.p11,'BackgroundColor','green');
        end
        if box==12
        set(handles.p12,'BackgroundColor','green');
        end
        if box==13
        set(handles.p13,'BackgroundColor','green');
        end
        if box==14
        set(handles.p14,'BackgroundColor','green');
        end
        if box==15
        set(handles.p15,'BackgroundColor','green');
```

```
        end
        if box==16
        set(handles.p16,'BackgroundColor','green');
        end
        end % es todo el cambio de color VERDE
    end
elseif length(vector)<=1000
    m=0;
    for i=1:length(vector)
        if vector(i)==1
            m=m+1;
        end
    end
    if m>=20
       for ok=1
       if box==1
       set(handles.p1,'BackgroundColor','red');
       end
       if box==2
       set(handles.p2,'BackgroundColor','red');
       end
       if box==3
       set(handles.p3,'BackgroundColor','red');
       end
       if box==4
       set(handles.p4,'BackgroundColor','red');
       end
       if box==5
       set(handles.p5,'BackgroundColor','red');
       end
       if box==6
       set(handles.p6,'BackgroundColor','red');
       end
       if box==7
       set(handles.p7,'BackgroundColor','red');
       end
       if box==8
       set(handles.p8,'BackgroundColor','red');
       end
       if box==9
       set(handles.p9,'BackgroundColor','red');
       end
       if box==10
       set(handles.p10,'BackgroundColor','red');
       end
       if box==11
       set(handles.p11,'BackgroundColor','red');
       end
       if box==12
       set(handles.p12,'BackgroundColor','red');
       end
       if box==13
       set(handles.p13,'BackgroundColor','red');
```

```
        end
        if box==14
        set(handles.p14,'BackgroundColor','red');
        end
        if box==15
        set(handles.p15,'BackgroundColor','red');
        end
        if box==16
        set(handles.p16,'BackgroundColor','red');
        end
        end % es todo el cambio de color ROJO
     elseif m>=15
        for ok=1
        if box==1
        set(handles.p1,'BackgroundColor','yellow');
        end
        if box==2
        set(handles.p2,'BackgroundColor','yellow');
        end
        if box==3
        set(handles.p3,'BackgroundColor','yellow');
        end
        if box==4
        set(handles.p4,'BackgroundColor','yellow');
        end
        if box==5
        set(handles.p5,'BackgroundColor','yellow');
        end
        if box==6
        set(handles.p6,'BackgroundColor','yellow');
        end
        if box==7
        set(handles.p7,'BackgroundColor','yellow');
        end
        if box==8
        set(handles.p8,'BackgroundColor','yellow');
        end
        if box==9
        set(handles.p9,'BackgroundColor','yellow');
        end
        if box==10
        set(handles.p10,'BackgroundColor','yellow');
        end
        if box==11
        set(handles.p11,'BackgroundColor','yellow');
        end
        if box==12
        set(handles.p12,'BackgroundColor','yellow');
        end
        if box==13
        set(handles.p13,'BackgroundColor','yellow');
        end
        if box==14
```

```
        set(handles.p14,'BackgroundColor','yellow');
        end
        if box==15
        set(handles.p15,'BackgroundColor','yellow');
        end
        if box==16
        set(handles.p16,'BackgroundColor','yellow');
        end
        end % es todo el cambio de color AMARILLO
    else
         for ok=1
        if box==1
        set(handles.p1,'BackgroundColor','green');
        end
        if box==2
        set(handles.p2,'BackgroundColor','green');
        end
        if box==3
        set(handles.p3,'BackgroundColor','green');
        end
        if box==4
        set(handles.p4,'BackgroundColor','green');
        end
        if box==5
        set(handles.p5,'BackgroundColor','green');
        end
        if box==6
        set(handles.p6,'BackgroundColor','green');
        end
        if box==7
        set(handles.p7,'BackgroundColor','green');
        end
        if box==8
        set(handles.p8,'BackgroundColor','green');
        end
        if box==9
        set(handles.p9,'BackgroundColor','green');
        end
        if box==10
        set(handles.p10,'BackgroundColor','green');
        end
        if box==11
        set(handles.p11,'BackgroundColor','green');
        end
        if box==12
        set(handles.p12,'BackgroundColor','green');
        end
        if box==13
        set(handles.p13,'BackgroundColor','green');
        end
        if box==14
        set(handles.p14,'BackgroundColor','green');
        end
```

```
        if box==15
        set(handles.p15,'BackgroundColor','green');
        end
        if box==16
        set(handles.p16,'BackgroundColor','green');
        end
        end % es todo el cambio de color VERDE
    end
else
    m=0;
    for i=length(vector)-1000:length(vector)
        if vector(i)==1
            m=m+1;
        end
    end
    if m>=20
        for ok=1
        if box==1
        set(handles.p1,'BackgroundColor','red');
        end
        if box==2
        set(handles.p2,'BackgroundColor','red');
        end
        if box==3
        set(handles.p3,'BackgroundColor','red');
        end
        if box==4
        set(handles.p4,'BackgroundColor','red');
        end
        if box==5
        set(handles.p5,'BackgroundColor','red');
        end
        if box==6
        set(handles.p6,'BackgroundColor','red');
        end
        if box==7
        set(handles.p7,'BackgroundColor','red');
        end
        if box==8
        set(handles.p8,'BackgroundColor','red');
        end
        if box==9
        set(handles.p9,'BackgroundColor','red');
        end
        if box==10
        set(handles.p10,'BackgroundColor','red');
        end
        if box==11
        set(handles.p11,'BackgroundColor','red');
        end
        if box==12
        set(handles.p12,'BackgroundColor','red');
        end
```

```
        if box==13
        set(handles.p13,'BackgroundColor','red');
        end
        if box==14
        set(handles.p14,'BackgroundColor','red');
        end
        if box==15
        set(handles.p15,'BackgroundColor','red');
        end
        if box==16
        set(handles.p16,'BackgroundColor','red');
        end
        end % es todo el cambio de color ROJO
    elseif m>=15
        for ok=1
        if box==1
        set(handles.p1,'BackgroundColor','yellow');
        end
        if box==2
        set(handles.p2,'BackgroundColor','yellow');
        end
        if box==3
        set(handles.p3,'BackgroundColor','yellow');
        end
        if box==4
        set(handles.p4,'BackgroundColor','yellow');
        end
        if box==5
        set(handles.p5,'BackgroundColor','yellow');
        end
        if box==6
        set(handles.p6,'BackgroundColor','yellow');
        end
        if box==7
        set(handles.p7,'BackgroundColor','yellow');
        end
        if box==8
        set(handles.p8,'BackgroundColor','yellow');
        end
        if box==9
        set(handles.p9,'BackgroundColor','yellow');
        end
        if box==10
        set(handles.p10,'BackgroundColor','yellow');
        end
        if box==11
        set(handles.p11,'BackgroundColor','yellow');
        end
        if box==12
        set(handles.p12,'BackgroundColor','yellow');
        end
        if box==13
        set(handles.p13,'BackgroundColor','yellow');
```

```
        end
        if box==14
        set(handles.p14,'BackgroundColor','yellow');
        end
        if box==15
        set(handles.p15,'BackgroundColor','yellow');
        end
        if box==16
        set(handles.p16,'BackgroundColor','yellow');
        end
        end % es todo el cambio de color AMARILLO
    else
         for ok=1
        if box==1
        set(handles.p1,'BackgroundColor','green');
        end
        if box==2
        set(handles.p2,'BackgroundColor','green');
        end
        if box==3
        set(handles.p3,'BackgroundColor','green');
        end
        if box==4
        set(handles.p4,'BackgroundColor','green');
        end
        if box==5
        set(handles.p5,'BackgroundColor','green');
        end
        if box==6
        set(handles.p6,'BackgroundColor','green');
        end
        if box==7
        set(handles.p7,'BackgroundColor','green');
        end
        if box==8
        set(handles.p8,'BackgroundColor','green');
        end
        if box==9
        set(handles.p9,'BackgroundColor','green');
        end
        if box==10
        set(handles.p10,'BackgroundColor','green');
        end
        if box==11
        set(handles.p11,'BackgroundColor','green');
        end
        if box==12
        set(handles.p12,'BackgroundColor','green');
        end
        if box==13
        set(handles.p13,'BackgroundColor','green');
        end
        if box==14
```

```
        set(handles.p14,'BackgroundColor','green');
        end
        if box==15
        set(handles.p15,'BackgroundColor','green');
        end
        if box==16
        set(handles.p16,'BackgroundColor','green');
        end
        end % es todo el cambio de color VERDE
    end
end


% val=num2str(length(vector));
% set(handles.text1,'string',val);

% m=0;
% m1=0;
% if IDbox==1;
%     m1(length(m1)+1)=n;
%     for i=1:length(m1)
%         if m1(i)==1;
%             m=m+1;
%         end
%     end
% end

%v=n;

% ESTO CAMBIA A STRING (PARA QUE PUEDA SACARSE EN PANTALLA)
%val=num2str(v);

% ESTO ES LO QUE HACE QUE CADA OUT ABSORBA UNO U OTRO VALOR
% if m>=1;
% set(handles.text1,'string','CODIGO NARANJA');
% end
% if m==0;
%     set(handles.text1,'string','CODIGO VERDE');
% end


fclose(file);
end %is the end of the first if
%%




function edit1_Callback(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

```
% Hints: get(hObject,'String') returns contents of edit1 as text
%        str2double(get(hObject,'String')) returns contents of edit1 as a↙
double


% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get↙
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


function edit2_Callback(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit2 as text
%        str2double(get(hObject,'String')) returns contents of edit2 as a↙
double


% --- Executes during object creation, after setting all properties.
function edit2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get↙
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on selection change in listbox2.
function listbox2_Callback(hObject, eventdata, handles)
% hObject    handle to listbox2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns listbox2↙
contents as cell array
%        contents{get(hObject,'Value')} returns selected item from↙
listbox2
```

```
% --- Executes during object creation, after setting all properties.
function listbox2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to listbox2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: listbox controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get↙
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


% --- Executes on button press in p2.
function p2_Callback(hObject, eventdata, handles)
% hObject    handle to p2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)


% --- Executes on button press in p6.
function p6_Callback(hObject, eventdata, handles)
% hObject    handle to p6 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of p6


% --- Executes on button press in p7.
function p7_Callback(hObject, eventdata, handles)
% hObject    handle to p7 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of p7


% --- Executes on button press in p10.
function p10_Callback(hObject, eventdata, handles)
% hObject    handle to p10 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of p10


% --- Executes on button press in p3.
function p3_Callback(hObject, eventdata, handles)
% hObject    handle to p3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
```

```
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of p3


% --- Executes on button press in p1.
function p1_Callback(hObject, eventdata, handles)
% hObject    handle to p1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of p1


% --- Executes on button press in p11.
function p11_Callback(hObject, eventdata, handles)
% hObject    handle to p11 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of p11


% --- Executes on button press in p5.
function p5_Callback(hObject, eventdata, handles)
% hObject    handle to p5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of p5


% --- Executes on button press in p4.
function p4_Callback(hObject, eventdata, handles)
% hObject    handle to p4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of p4


% --- Executes on button press in p12.
function p12_Callback(hObject, eventdata, handles)
% hObject    handle to p12 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of p12


% --- Executes on button press in p13.
function p13_Callback(hObject, eventdata, handles)
% hObject    handle to p13 (see GCBO)
```

```
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of p13


% --- Executes on button press in p14.
function p14_Callback(hObject, eventdata, handles)
% hObject    handle to p14 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of p14


% --- Executes on button press in p9.
function p9_Callback(hObject, eventdata, handles)
% hObject    handle to p9 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of p9


% --- Executes on button press in p8.
function p8_Callback(hObject, eventdata, handles)
% hObject    handle to p8 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of p8


% --- Executes on button press in p15.
function p15_Callback(hObject, eventdata, handles)
% hObject    handle to p15 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of p15


% --- Executes on button press in p16.
function p16_Callback(hObject, eventdata, handles)
% hObject    handle to p16 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of p16


% --- Executes on selection change in listbox3.
function listbox3_Callback(hObject, eventdata, handles)
```

```
% hObject    handle to listbox3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns listbox3↙
contents as cell array
%        contents{get(hObject,'Value')} returns selected item from↙
listbox3


% --- Executes during object creation, after setting all properties.
function listbox3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to listbox3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
```