Anexos

Anexos A

Estudio de los parámetros recogido del controlador de soldadura por haz láser

El primer análisis por realizar es conocer si el conjunto de datos recogido tiene valores nulos porque la forma de abordar los siguientes pasos variara en función de esto. En la gráfica A.1 se puede ver como en la mayoría de los parámetros la cantidad de valores que se recogen es inferior al 20 %, siendo prácticamente nulos en algunos casos

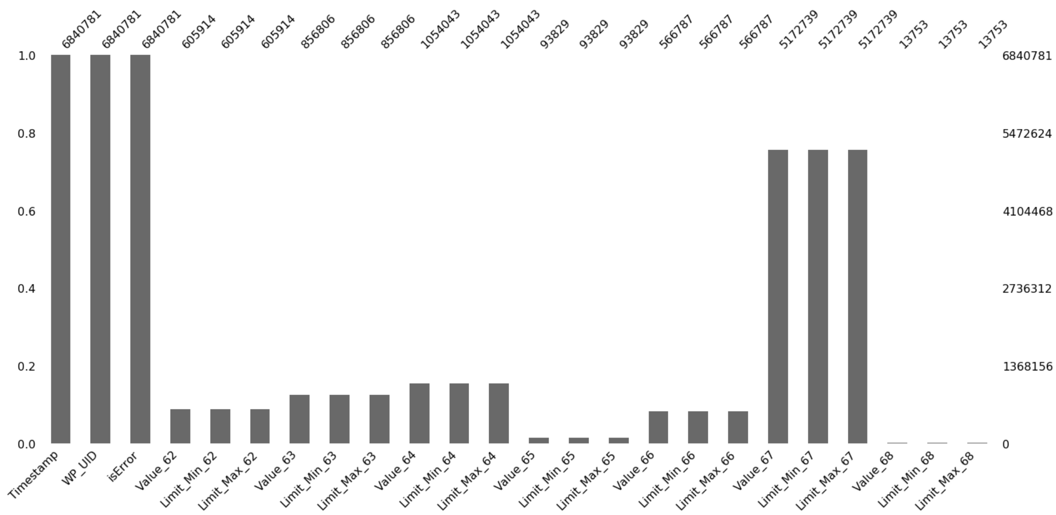


Figura A.1: Diagrama de barras de la cantidad de valores no nulos existentes en el dataset. A la izquierda se representa en porcentaje. A la derecha se presenta en formato numérico. Arriba el total de datos no nulos para cara característica.

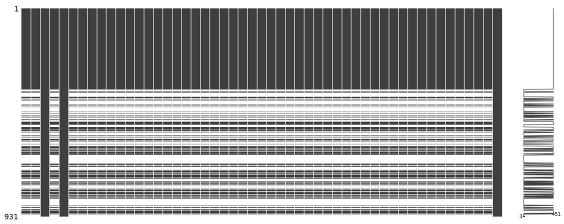
El segundo estudio que se realizó fue conocer la cantidad de fallos que se habían producido en el periodo durante el cual se recogieron los datos. El total filas es de 6840781 dentro de las cuales 3258 se asocian a fallos. Hay que tener en cuenta de que en el proyecto los datos recogidos se corresponden con muestras reales (a excepción del ruido añadido), lo que condiciona que en ocasiones las clases con las que se trabaja se encuentren tan desbalanceadas, como ocurre en este caso.

Con este pequeño análisis se puede concluir que hace falta recoger durante un periodo más largo para poder aplicar alguna técnica de rellenado de datos.

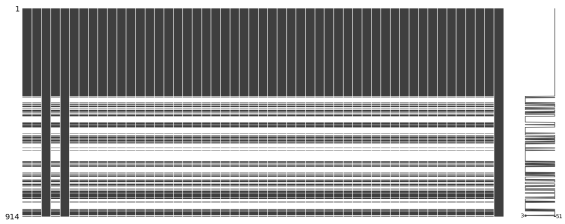
Anexos B

Estudio de los parámetros recogidos del controlador de soldadura de pernos

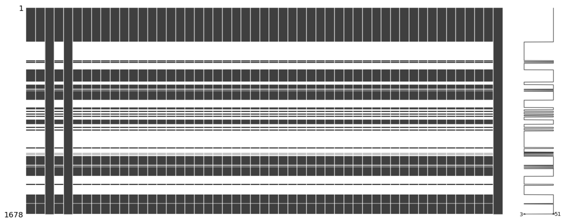
Tras unir la tabla que contiene los parámetros propios de la soldadura con el de fallos, se obtiene un dataset en el que existen gran cantidad de columnas con valores nulos así que en primer lugar se va a buscar el motivo de porqué ocurre esto. Como se puede ver en la figura B.1c se aprecian espacios temporales en blanco los cuales representan la presencia continuada de un fallo o , al estar la línea recién montada, también puede deberse a que se apagan los equipos. En las otras tres figuras B.1 también se puede apreciar esta condición pero en menor medida.



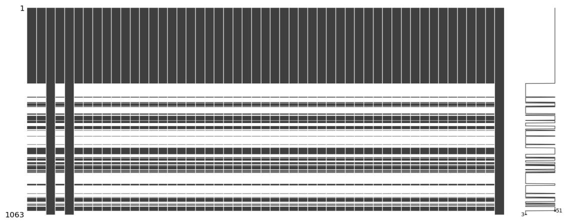
(a) Dispositivo número 1



(b) Dispositivo número 2



(c) Dispositivo número 3



(d) Dispositivo número 4

Figura B.1: Visualización de los valores nulos para cada parámetro a lo largo del tiempo de captura.

Teniendo en cuenta de que los dispositivos están situados en una línea de trabajo en la que todavía no se está en plena producción y realizando una inspección visual al momento de los fallos, se concluye que posiblemente el error se ocasiona como consecuencia del último punto soldado y el resto son el mismo error que se repite de forma continua hasta que un técnico acude a solventarlo.

En este punto, se contabilizan 147 columnas en el fichero de parámetros de soldadura y 25 en el fichero de fallos, lo que supone un total de 172 columnas. Se observa que ciertas columnas tienen el mismo valor siempre por lo que se descartan ya que los valores constantes no aportan un peso significativo a la hora de predecir. A continuación, nos seguíamos enfrentando ante un número considerable de columnas y no conocíamos cuanto aportarían a la tarea de predecir errores por lo que se realizó la relación por pares de cada columna con el objetivo de descartar aquellas que no mostrasen una buena división en cuanto a presencia de errores.

Se divide el dataset dedicando un 60 % para el enterramiento, un 20 % para la validación y el 20 % restante para un test final. En la siguiente tabla se muestra la cantidad de ceros (no existencia de fallos) y unos (presencia de fallos) que hay en cada subdivisión.

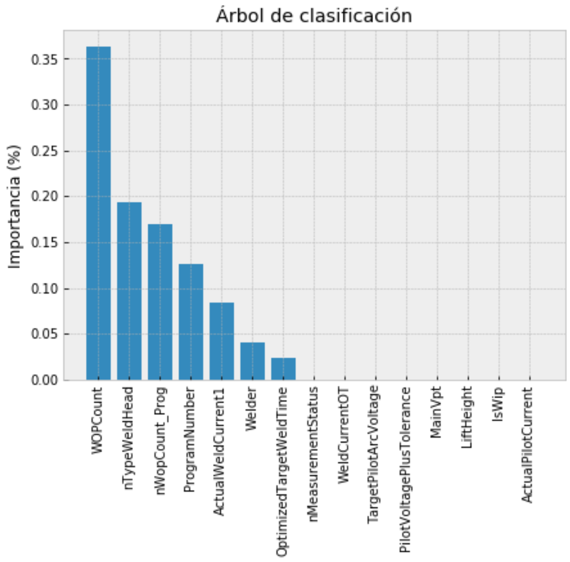
	Entrenamiento	Validación	Prueba final	Total	Porcentaje sobre el total (%)
Valor 0	1637	549	551	2737	94.73
Valor 1	96	29	27	152	5.27

Tabla B.1: Datos disponibles para realizar el estudio

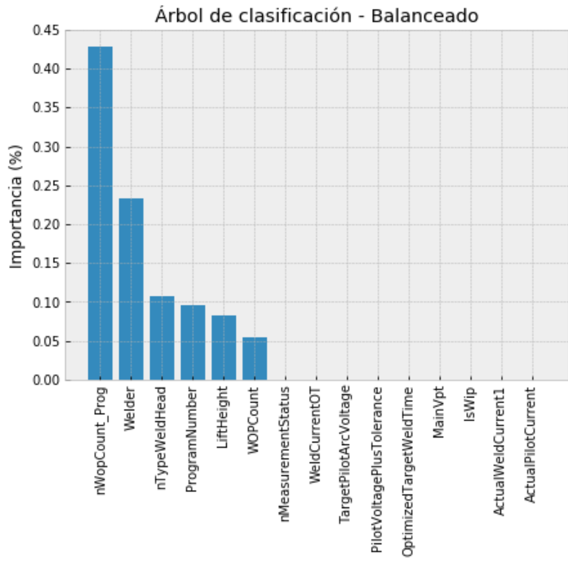
Finalmente, se aplican diferentes algoritmos de aprendizaje supervisado donde se busca detectar las características más importantes. Como consecuencia del bajo porcentaje de fallos existente, es muy fácil que el predictor acierte. La forma de intentar solventar este problema es utilizar las opciones de balanceado de clases que ofrecen los algoritmos.

Como ya se había mostrado en 4.2 se puede ver que tras aplicar el balanceo el porcentaje de acierto se reduce, esto podría parecer una desventaja pero en nuestro caso supone una aproximación a una realidad en la que la cantidad de muestras disponibles hubiera sido más grande.

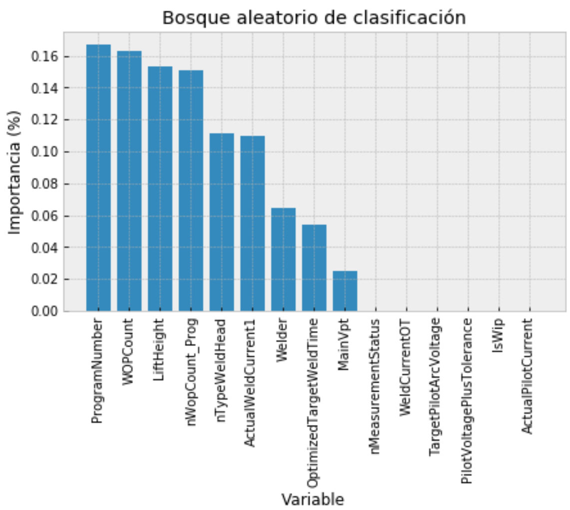
En las siguientes figuras se muestran los gráficos de barras generado a partir de los porcentajes que devuelven los algoritmos sobre las características más importantes.



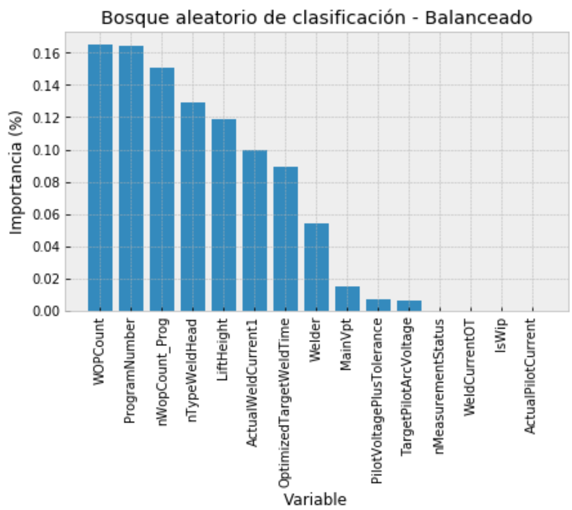
(a) Árbol de decisión sin hacer uso de la opción de balancear las clases



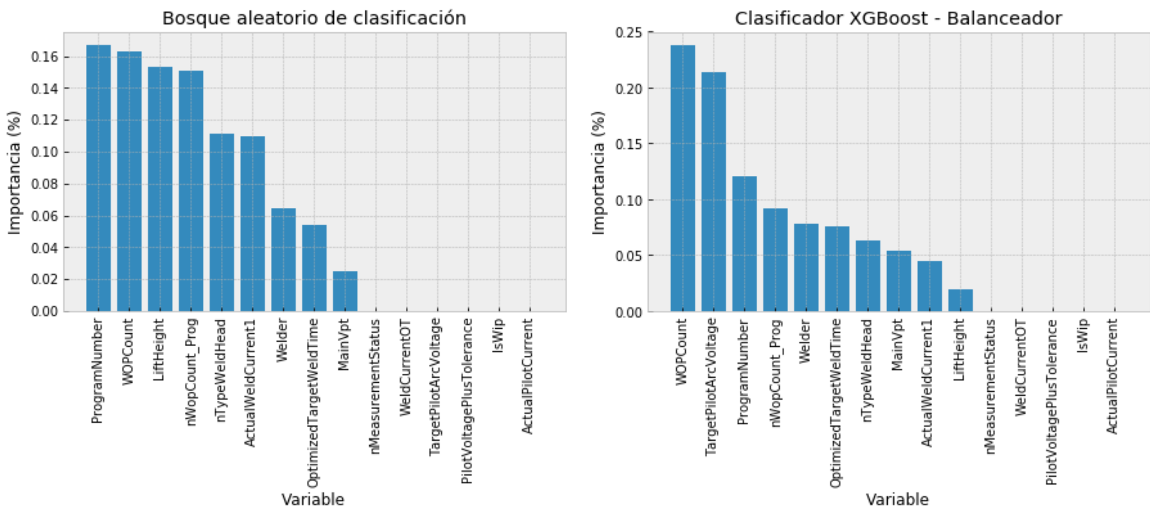
(b) Árbol de decisión haciendo uso de la opción de balancear las clases



(c) Bosque aleatorio sin hacer uso de la opción de balancear las clases



(d) Bosque aleatorio haciendo uso de la opción de balancear las clases



(e) Bosque aleatorio sin hacer uso de la opción de balancear las clases

(f) Bosque aleatorio haciendo uso de la opción de balancear las clases

Figura B.2: Diagrama de barras en función de los porcentajes de importancia de los parámetros para árbol de decisión.

Anexos C

Estudio de los parámetros recogido del controlador de unión por adhesivo

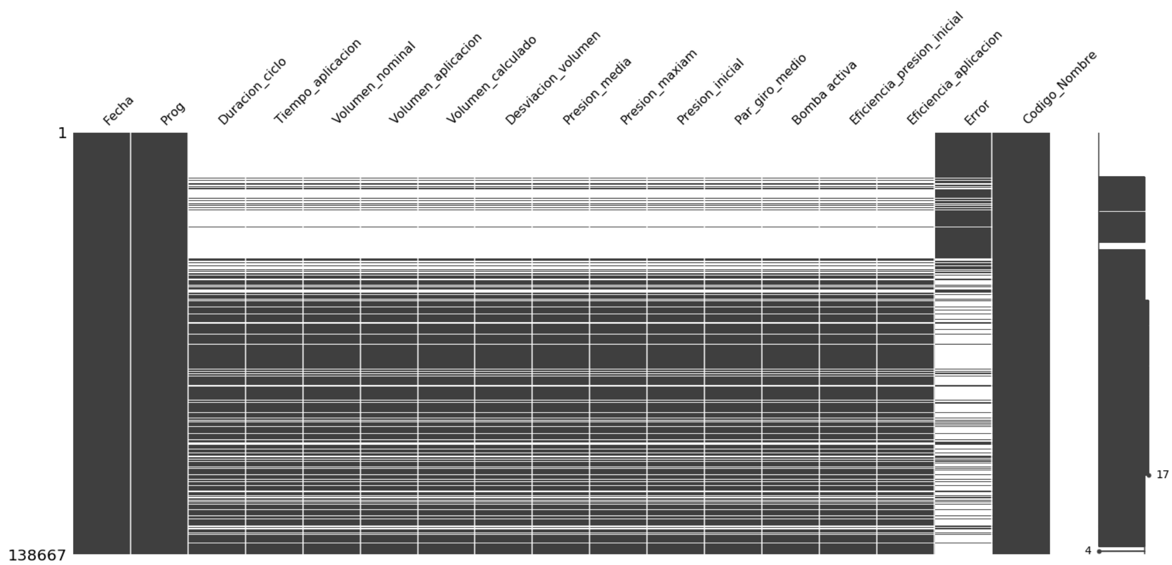


Figura C.1: Diagrama de barras de la cantidad de valores no nulos existentes en el dataset. A la izquierda se representa en porcentaje. A la derecha se presenta en formato numérico. Arriba el total de datos no nulos para cara característica.

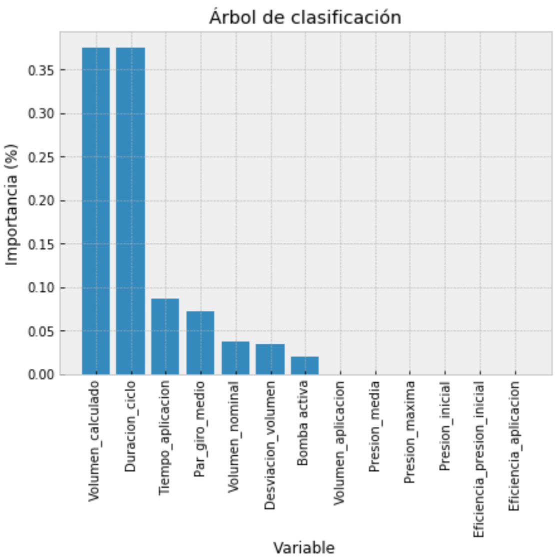
En este anexo se va a realizar un análisis más detallado de la búsqueda de las variables más importantes utilizando modelos de clasificación orientados a predecir fallos. Los algoritmos que van a utilizar son los siguientes: arboles de decisión, bosques aleatorios y XGBoost.

Una de las formas más visual para ver los parámetros más importantes es dibujar un diagrama de barras utilizando los valores que devuelve el modelo una vez se ha entrenado. En las figuras que se representan a continuación se puede ver como la influencia de la opción de balancear las clases es bastante fuerte ya que los parámetros que seleccionan los modelos son bastante diferentes en función de si se ha marcado la opción o no. En nuestro dataset resulta de interés ya que el porcentaje de filas que contiene fallos es de 27,29 %, de esta forma se crea un equilibrio entre las clases y el resultado es más fiel a la realidad, comparándolo con el caso de tener un dataset más completo. Después de visualizar las gráficas C.2, la conclusión que se obtiene en cuanto a relevancia de parámetros son los que se podrían descartar en futuros análisis ya que en todos los modelos *Bomba_activa* se encuentra en la última posición.

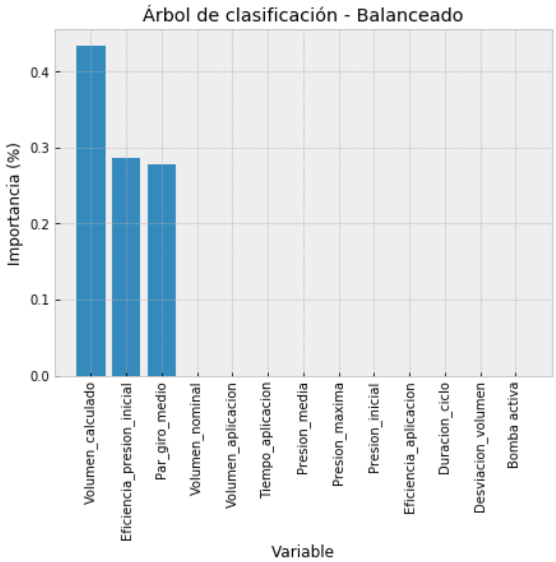
En los modelos de clasificación, el valor devuelto es binario, o cero o uno, por lo que se puede cuantificar la estimación que realiza. Pese a que el porcentaje que se obtiene es más bien representativo, sirve para concluir si es viable predecir los fallos a través de los parámetros que nos ofrece el controlador. Para este caso, se obtiene que en todos los modelos el porcentaje supera el 70 %, lo que indica que los parámetros si contiene información relevante. En la tabla 4.4 se pueden ver los valores obtenidos para cada modelo.

Otro factor que hay que tener en cuenta es que, como ya se vio en la descripción de la tabla de fallos que se realiza en el Anexo D se contemplan como errores los avisos que genera el visor Quiss. La acción de revisar esta alarma conllevaría muy poco tiempo por parte de los empleados de planta y, ante nosotros, se puede considerar como un falso negativo. Así se propone que los modelos regresión pueden ser una buena alternativa ya que permiten marcar el umbral a partir del cual debería ser considerado fallo. Se implementan para los mismos clasificadores elegidos antes. Al igual que en caso anterior devuelven los parámetros más importantes, como se puede ver en la figura C.3.

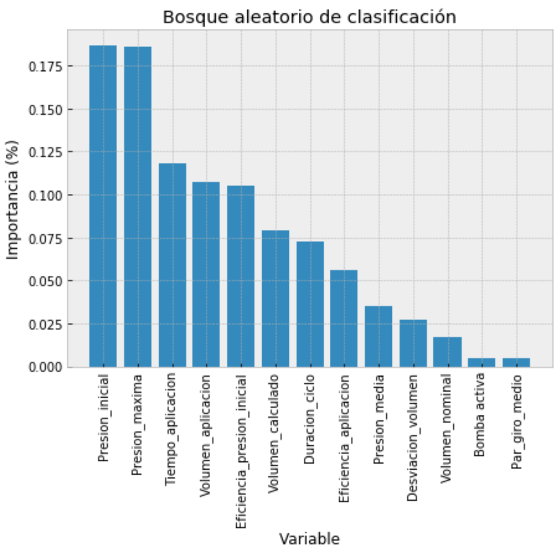
También una gráfica conocida como la curva ROC donde se representan la sensibilidad de falsos positivos frente a de verdaderos positivos según varía el umbral y el valor AUC (Área Bajo la Curva ROC) que se interpreta como la probabilidad de que el modelo clasifique de forma correcta cada etiqueta. Se observa que al igual que ocurría en el caso de clasificación, en regresión se obtiene un porcentaje del más del 70 % en todos los casos, por lo que se reafirma que los datos serán válidos para predecir errores.



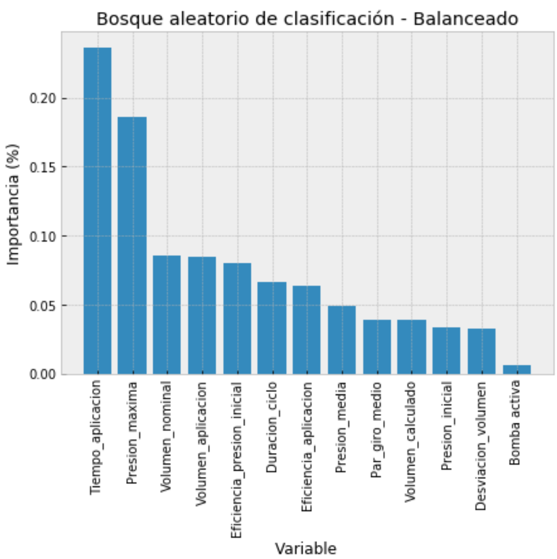
(a) Árbol de decisión sin hacer uso de la opción de balancear las clases



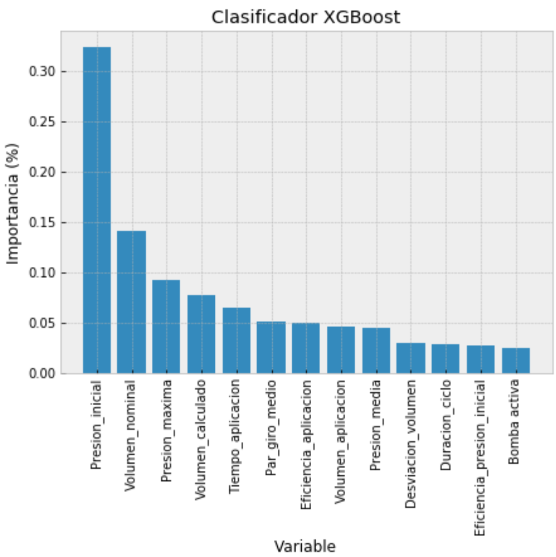
(b) Árbol de decisión haciendo uso de la opción de balancear las clases



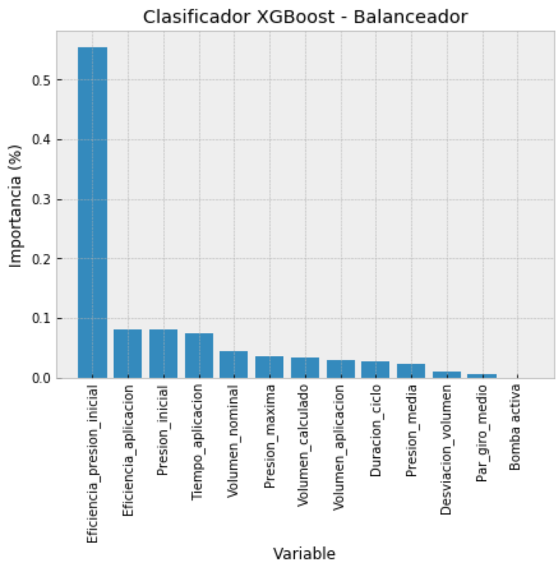
(c) Bosque aleatorio sin hacer uso de la opción de balancear las clases



(d) Bosque aleatorio haciendo uso de la opción de balancear las clases

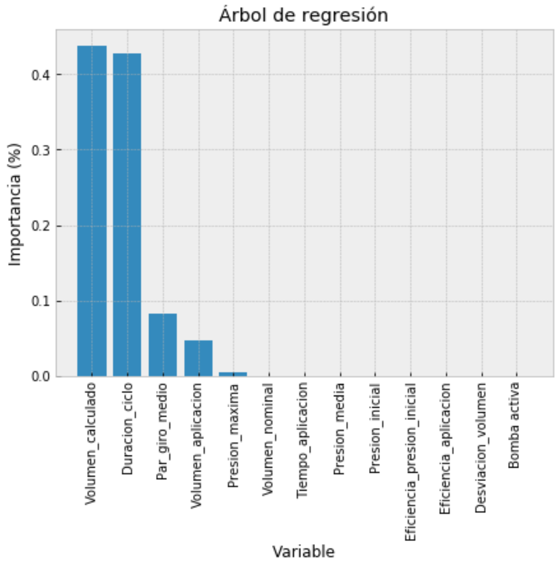


(e) XGBoost sin hacer uso de la opción de balancear las clases

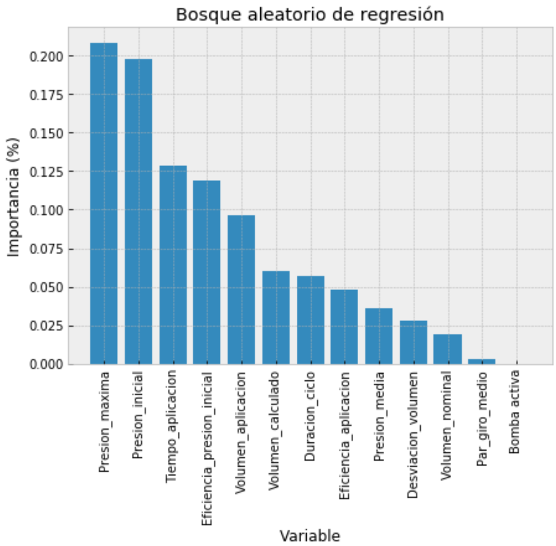


(f) XGBoost haciendo uso de la opción de balancear las clases

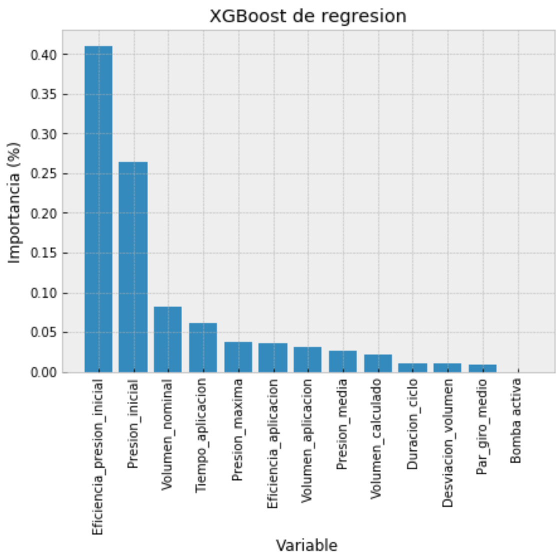
Figura C.2: Diagrama de barras en función de los porcentajes de importancia de los parámetros en función de los diferentes algoritmos de clasificación.



(a) Características más importantes en el algoritmo árbol de decisión.



(b) Características más importantes en el algoritmo bosque aleatorio.



(c) Características más importantes en el algoritmo XGBoost.

Figura C.3: Características más importantes de cada uno de los algoritmos probados

Anexos D

Informe de Opel

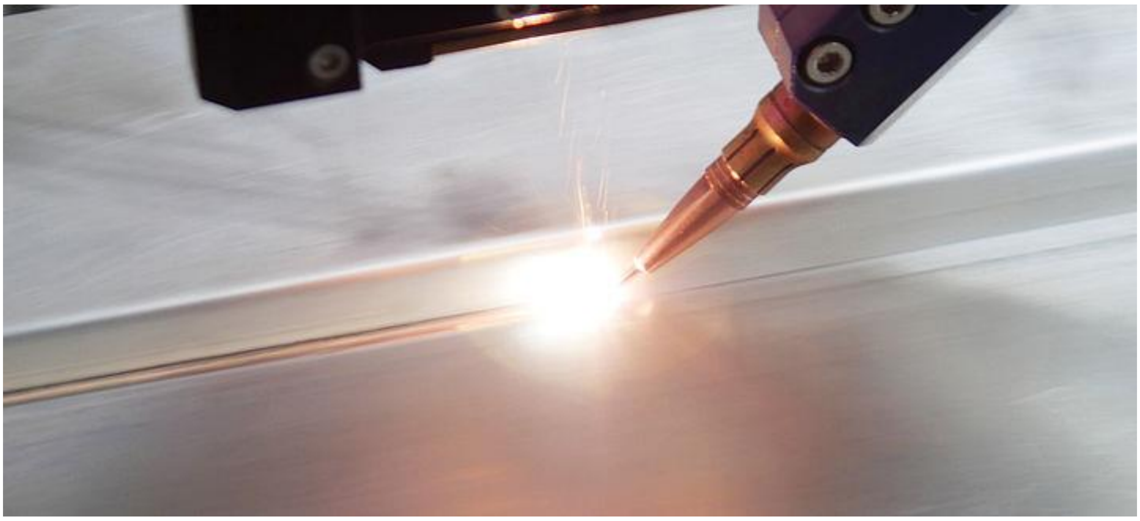
En las siguientes páginas se incluye el informe que se realizó para Opel.

Welding Wizard Phase 3

Device Analysis

Index

1 Laser brazing (Lessmüller)

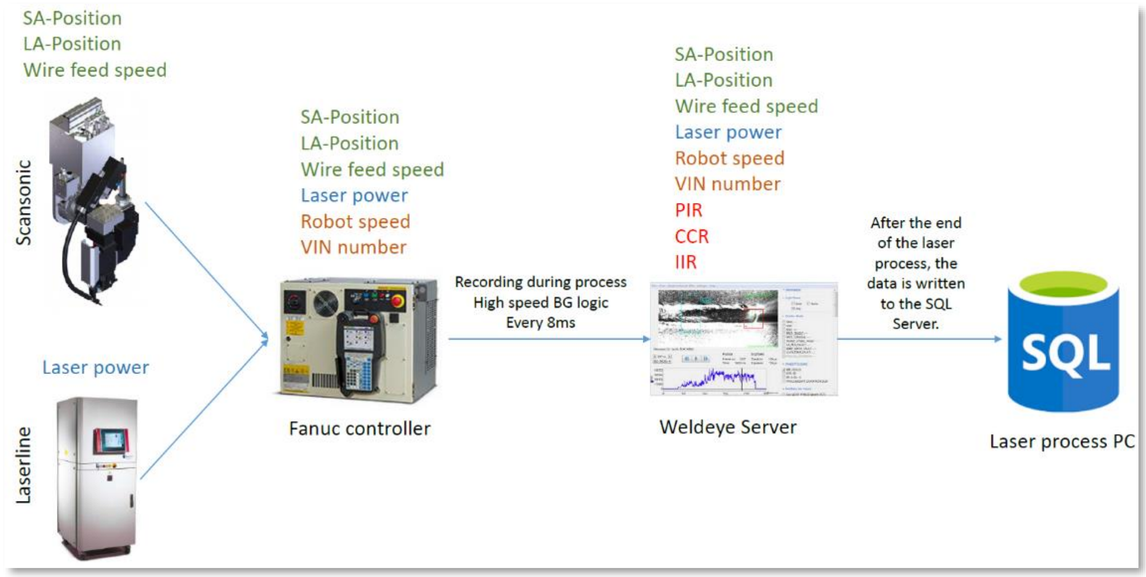


1.1 Introduction to the laser brazing technology

The laser brazing process uses a filler metal for joining without actually melting the base material.

This process allows to achieve high precision and minimal warpage on the workpiece with excellent cost effectiveness.

The laser brazing process includes different kind of devices connected in the way described on the next image:



LASER BRAZING - CONNECTING PRINCIPLE

From the perspective of this study, the key element is the Weldeye Server, where the information about the results of the process is stored.

1.2 Device inventory

1.2.1 Connectivity possibilities

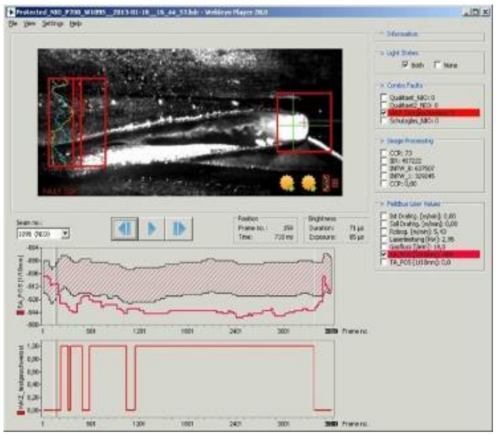
Every installation includes a Weldeyer Server, who oversees the communications through the field bus interface, the communication / input of the camera image and the analysis / registration of the the image data.

The software, developed by the company Lessmüller, allows to select the parameters of interest in the Weldeye Server and to send them to a SQL Server database in another computer.

1.3 Device access method

1.3.1 Software

Laser brazing involves some technical complexity. “Lessmüller Weldeye” is installed on every application to control and understand the behavior of the process.



Fa. Lessmüller developed an SQL interface to extract welding data out of Weldeye and store on an external SQL database located in a Process PC (like a MWS). The selection of the extracted data is configurable.

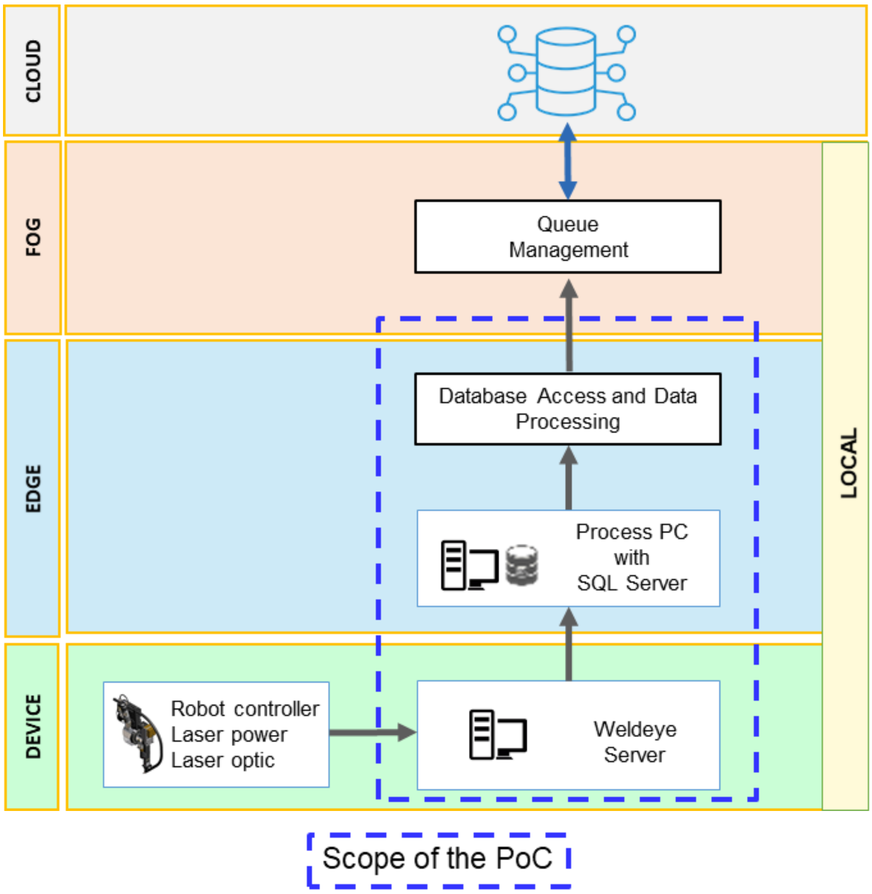
WW3 project makes use of the welding data from the SQL database of the MWS.

Se trata de un software que se instala en el Process PC y permite la extracción de los datos del Weldeye Server y su envío a una base de datos.

1.3.2 Proof of Concept

This software was acquired, installed and configured by a welding engineer

The Process PC has two communication cards; one to communicate with the Weldeye internal network and another to communicate with the manufacturing network. To get remote access to the database, it was necessary to enable the address of the second network card in SQL Server, which was not configured. The architecture used for proof of concept is presented below.



The sending of data from the Weldeye Server to the database can only be done while there is no welding, ie in the period from one body leaves after being welded until it enters the next, which is approximately 11 seconds. This time limitation is reflected in the fact that it will be necessary to select the most interesting parameters in the Weldeye software.

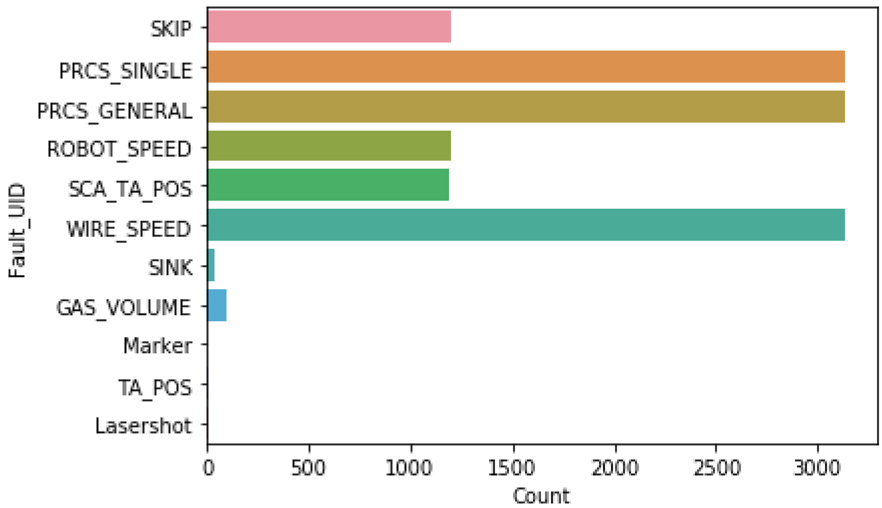
Because maintenance can enable or disable data capture, only one week was captured, which was enough time to store an exemplary sample of the data and tables that make up the database. During this period, the data was stored practically in real time considering that the sending is only done in certain instants.

1.4 Device Data Analysis

The database on the MWS contains seven tables:

- Two tables with analog values and their coding
  - Process_Execution_Data_Analogue
  - Process_Execution_Data_Index.
- Two tables with the faults and their coding
  - Combo_Fault_Index
  - Combo_Fault
- One table with the configuration
  - Parts
- One table with digital values (empty)
  - Process_Execution_Data_Digital
- One table with the configurations for different components/car models
  - WPs

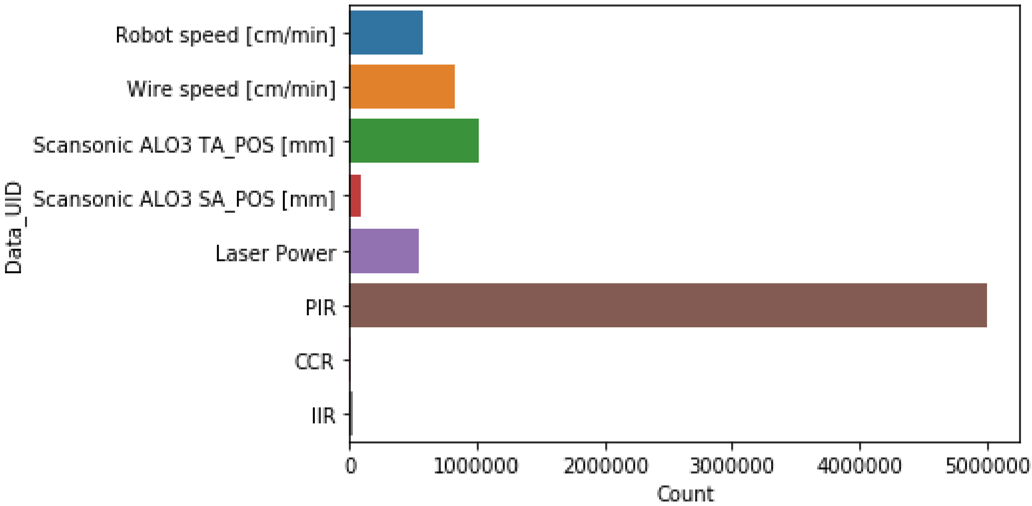
Data was stored between 2019-05-17 11:04:41.559 and 2019-05-24 23:50:32.148 for the table called Combo_Fault. During this time the following errors associated with the time they occurred were captured.



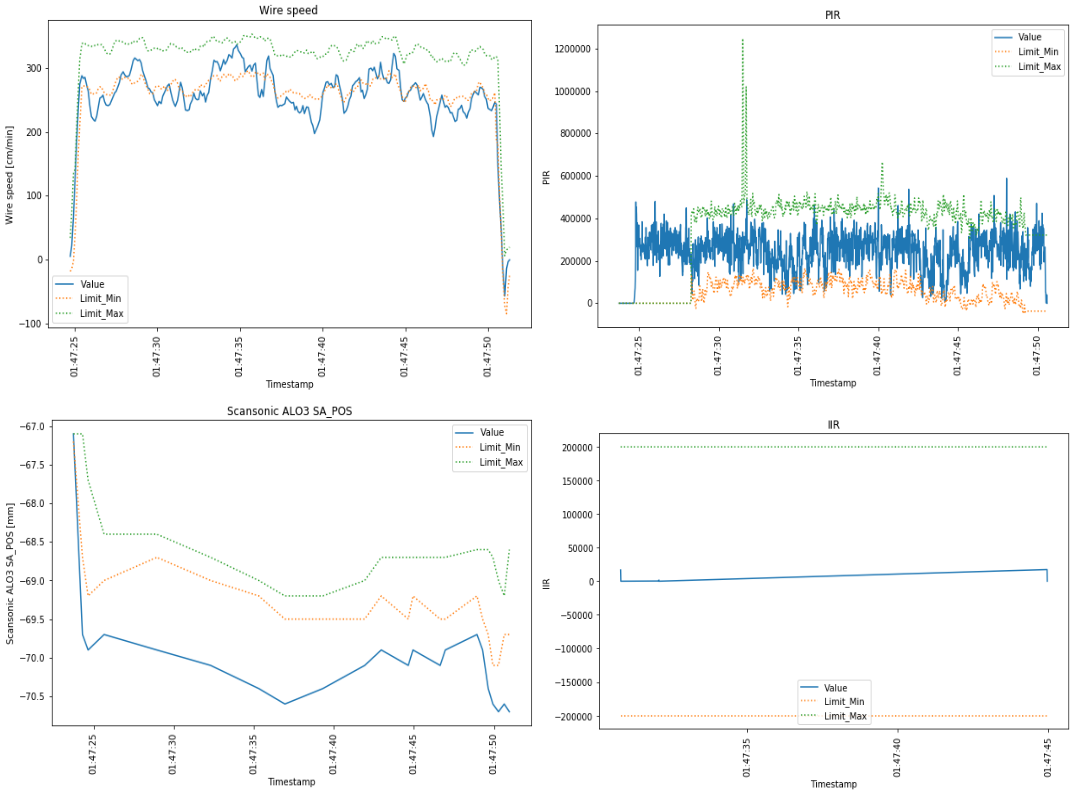
For the Process_Execution_Data_Analogue table, data was captured between 2019-05-17 11:04:13.898 and 2019-05-24 23:50:32.047. The fields that are stored in this table are the Timestamp of each weld, a field for the coding that is done of the parameters it generates and the maximum and minimum value. The name of the fields differs in the database from its real name so that in the table you can see the different assignments that each one has.

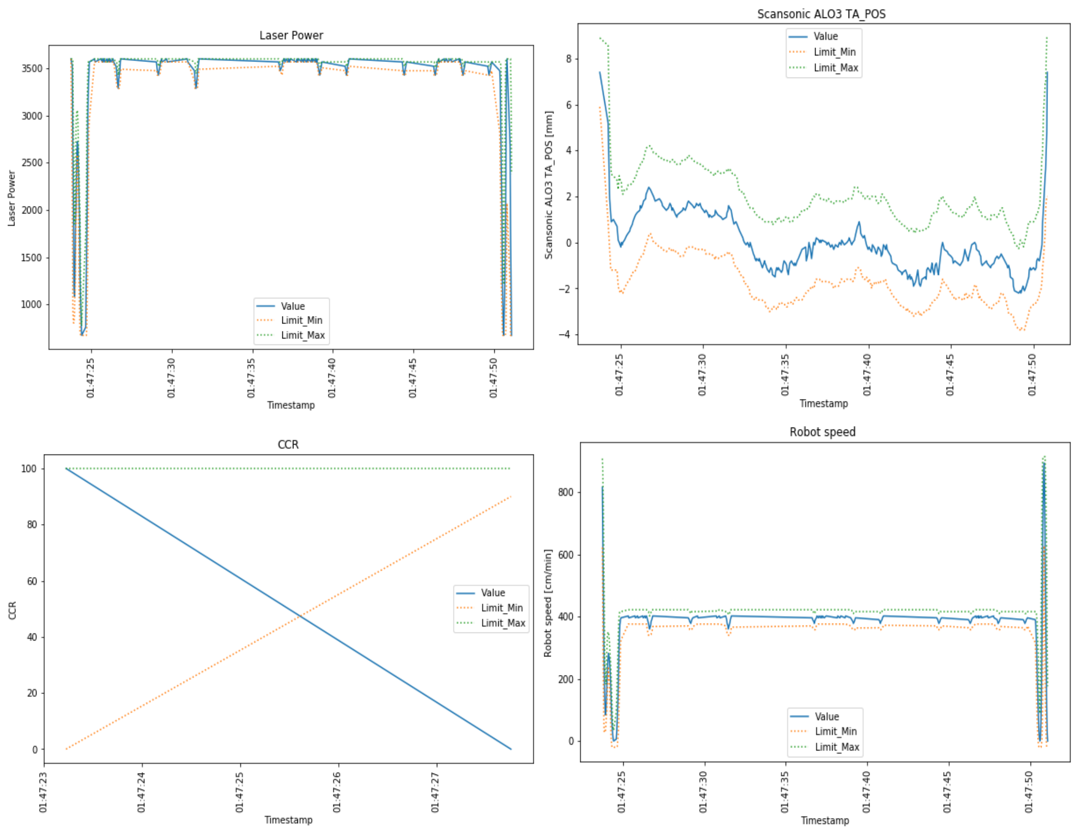
Name	Process_Execution_Data	Code
Robot speed [cm/min]	FIELDBUS_VALUE_1	62
Wire speed [cm/min]	FIELDBUS_VALUE_2	63
Scansonic ALO3 TA_POS [mm]	FIELDBUS_VALUE_4	64
Scansonic ALO3 SA_POS [mm]	FIELDBUS_VALUE_5	65
Laser Power	FIELDBUS_VALUE_6	66
PIR	PIR	67
CCR	CCR	68
IIR	IIR	69

The following table shows a count of each parameter along with the number of times it is received. The most frequent is the PIR that refers to the power of the light emitted by the welding process in the infrared band.



The following figure shows an example of all the curves that are received for a welding cycle.



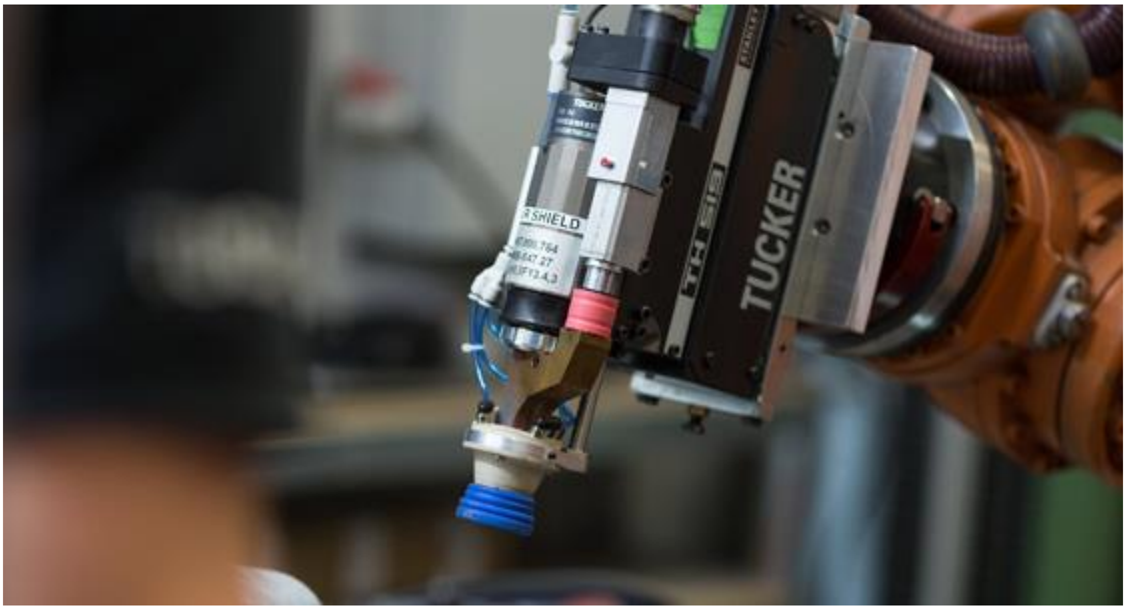


1.5 Database description

Accessing the database hosted in the Process PC you can see that the system is able to generate 7 tables, which are described in more detail below:

- Two tables with analog data and their corresponding coding: Pro-cess_Execution_Data_nalogue, Process_Execution_Data_Index.
- Two tables with the faults that have occurred and their corresponding coding: Combo_Fault_Index, Combo_Fault
- A table with the configuration: Parts
- A table with digital data that is empty: Process_Execution_Data_Digital
- One table with the configurations for different components/car models: WPs

2 Stud Welding (Tucker)



2.1 Introduction

Stud welding is responsible for joining studs by applying pressure after heating both elements sufficiently with an electric arc.

2.2 Device inventory

2.2.1 DCE Controllers

DCE is a family of stud welding devices from Fa. Stanley Tucker

DCE1500 and DCE1800 are basically the same device. The difference is the screen:

- DCE1500 has a monochrome screen
- DCE1800 has a colour touchscreen

2.2.2 TE Controllers

TE is a modern family of stud welding devices from Fa. Stanley Tucker, from which some TE1500 are installed on OV sites.

2.3 DCE Controllers

2.3.1 Connectivity possibilities

The way to collect the data generated by the controller is through the DCELink software of the company Tucker (Stanley) offering the possibility of establishing a network connection between the welding controllers and a client-server application.

2.3.2 Device access method

2.3.3 Software installation for DCE controllers

Tucker developed the software DCE link to connect with the DCE controllers. It is made up of three modules:

- **DCE-Link Server**
  Application that manages the communication between the controllers and the database by means of TCP/IP protocol. It needs to run permanently.
- **DCE-Link Manager**
  Application that runs on the background. It controls and monitors the status of DCE Link Server. It needs to run permanently.
- **DCE-Link Client**
  Graphical User Interface of DCE-Link Server. It does not need to run permanently. It connects through TCP/IP with the server, which provides access to the database.

The different functionalities offered by the software are the following: Monitorización del estado de la red de los dispositivos DCE

- Remote programming of DCE devices via a central computer
- Data Backup and Restore

The requirements to be able to install DCELink software are:

- Have access to a computer with a Windows operating system.
- Have a relational Microsoft SQL Server database system installed.
- Each DCE controller must have a unique IP address to identify them once it has been configured in the DCELink-Client application.

Although the software is compatible with the DCE1500 and DCE1800 models, it was not possible to establish a connection with the DCE 1800 devices because, according to information provided by a manufacturer's technician, the firmware version has a specific GM layer that prevented the completion of the Three-Way Handshake protocol. This meant that it was only possible to establish the connection with the DCE 1500 devices.

It is the responsibility of the departments of Manufacturing Engineering and Maintenance to investigate how to communicate the DCE 1800. A technician of the manufacturer suggested to install in the DCE1800 the same version of system that the DCE1500 have.

It is also important to mention that a second network card is required along with a fiber optic cable in the equipment, if not installed. This second card allows the sending of data through the communication enabled by DCELink.

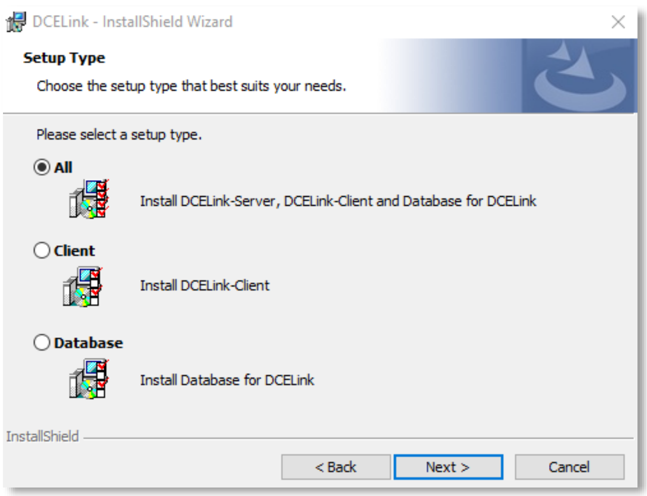
The following is a brief description of the DCELink software installation process

1. Verify that the computer on which the installation is to be performed can communicate with the driver via the ping command.

```
C:\Windows\system32>ping IP
Pinging IP address with 32 bytes of data:
Reply from IP: bytes=32 time=1ms TTL=25
Reply from IP: bytes=32 time=1ms TTL=25
Reply from IP: bytes=32 time=1ms TTL=25
Reply from IP: bytes=32 time=1ms TTL=25
Ping statistics for IP:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 1ms, Maximum = 1ms, Average = 1ms
```

2. Run the installer named InstallDCELink.exe

3. The appropriate installation option is chosen. In the Proof of Concept, the client, server and database were installed on the same computer.



4. In the following steps the instructions of the manual were followed.

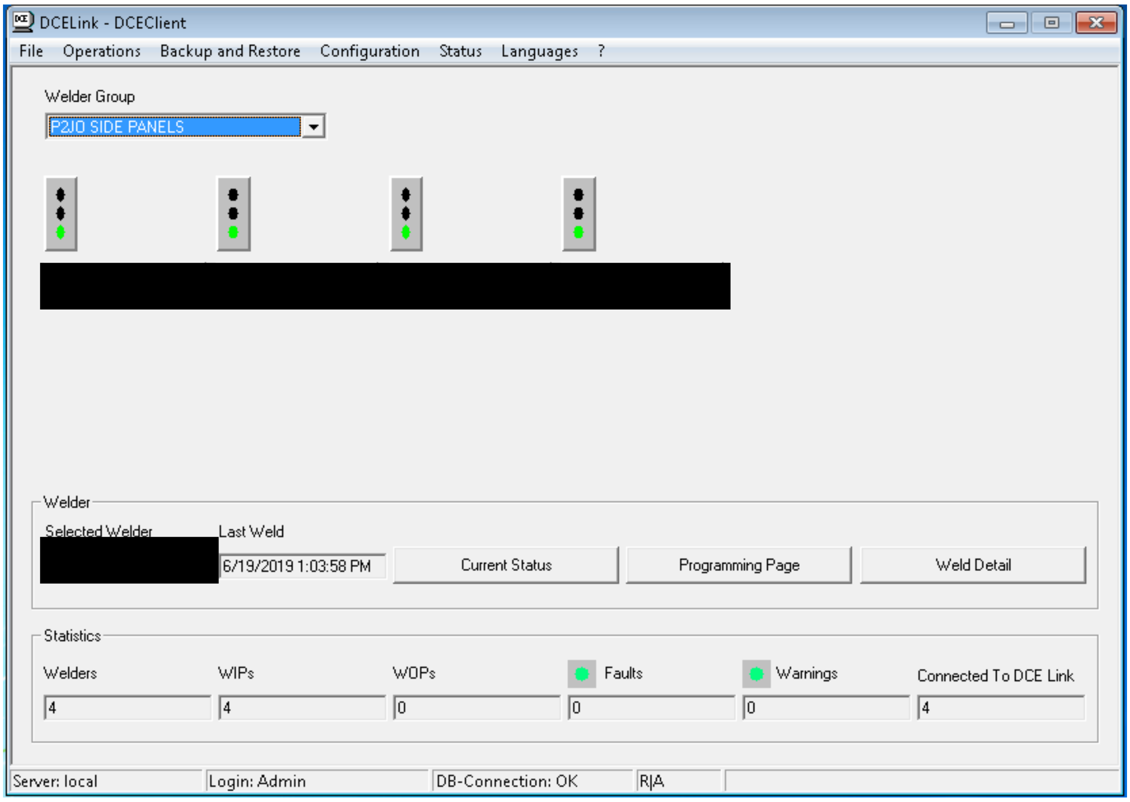
5. Finally, it is verified that the connection has been established with the following command: netstat -nao | find “87”.

```
C:\Windows\system32>netstat -nao | find "87"
  TCP    IP:49946    IP    ESTABLISHED    1420
  TCP    IP:49989    IP    ESTABLISHED    1420
  TCP    IP:56590    IP    ESTABLISHED    1420
  TCP    IP:56625    IP    ESTABLISHED    1420
```

If the connection status is different from "ESTABLISHED", it is necessary to check that port 87 is open. If it is not open, it will be necessary to open it since the communication with the server is done through this port.

After completing the installation, manually add each unit to be monitored in the DCE-Client application, following chapter "6.5.2 Configure Plant" of the manual. To add each device, it is necessary to know its IP address and assign it a unique name with which each one will be identified.

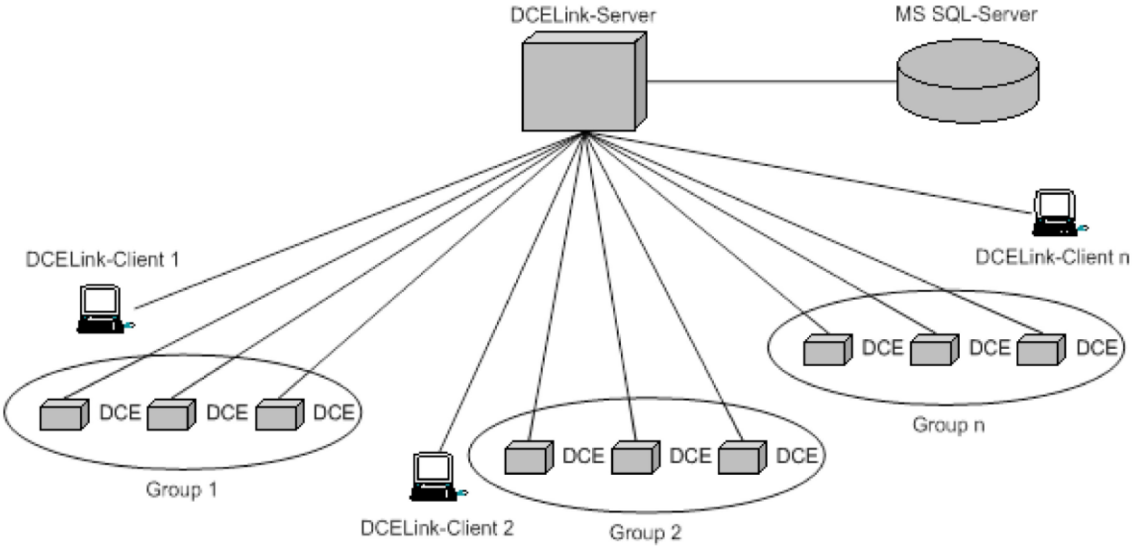
When configuration is complete, the main screen displays all devices that have been added as shown in the figure below.

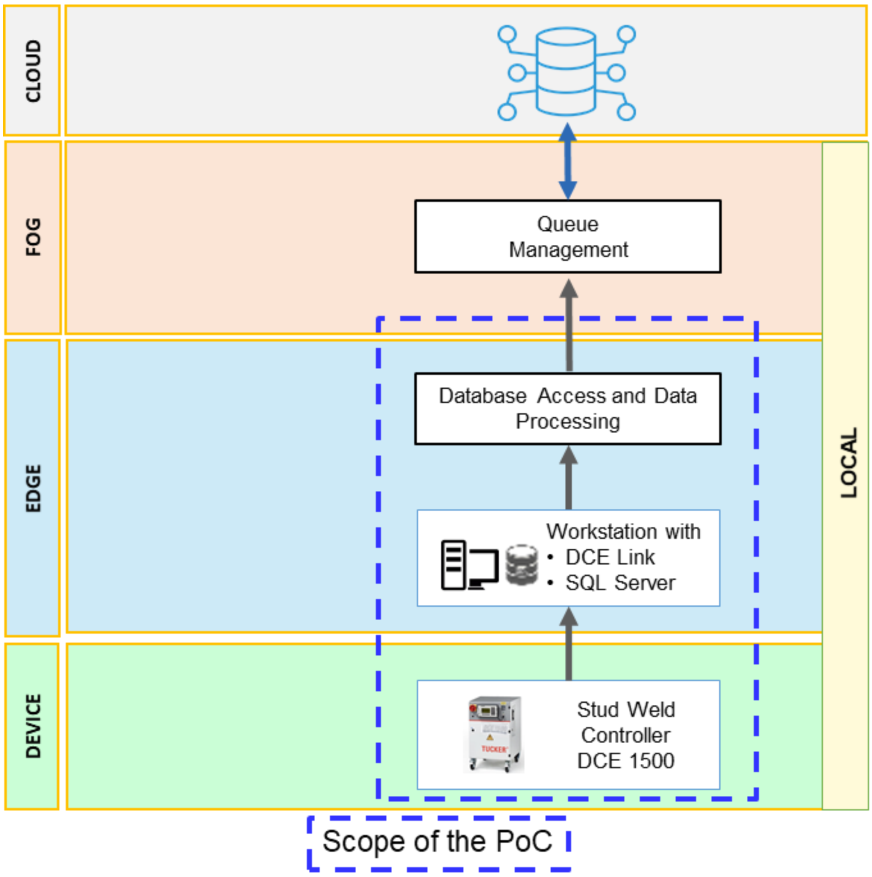


2.3.4 Proof of Concept

it was carried out in the Zaragoza plant, in one of the computers located in the laboratory. The three modules that make up the system were installed there. The installed DCELink software version is 1.82.

Although the manual proposes an architecture in which each module is installed on different computers, for proof of concept was made the installation of all on the same computer. Next, you can see the architecture proposed by the manual and the architecture that was finally implemented.





2.3.5 Device Data Analysis

The SQL Server database for DCE 1500 and DCE 1800 is the same.

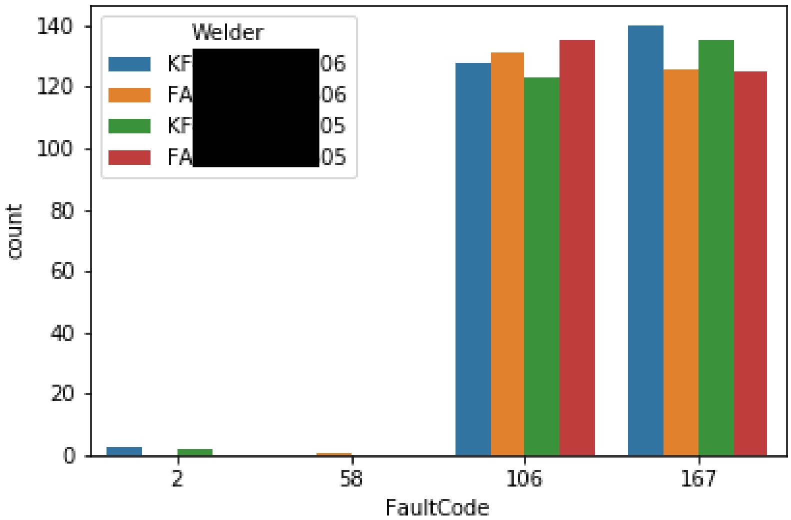
Accessing the database through the SQL Server 2014 Management Stu-dio program, the tables that are stored are analysed. The date of the first capture is co-rresponde with 2018-12-03 17:31:53.000.

It contains 32 tables from which the most important are:

- Faults
  It contains a list of the weld faults. FaultCode referes to the type of fault. 25 fields with datetime, integer and object data formats

- Welds
  It contains all parameters that control the stud welding process. 147 fields with boolean, datetime, float, integer data formats
  A great number of fields from this database are threshold values, static.

Among the most important fields in the Faults table is the so-called FaultCode where the code associated with the error that has occurred at a given instant is stored. In the manual called "Programming Manual Control and Power Unit TMP" you will find the description associated with each fault number. In the figure you can see a count of the errors that have occurred in the period that has been collected, differentiating which controller they come from.



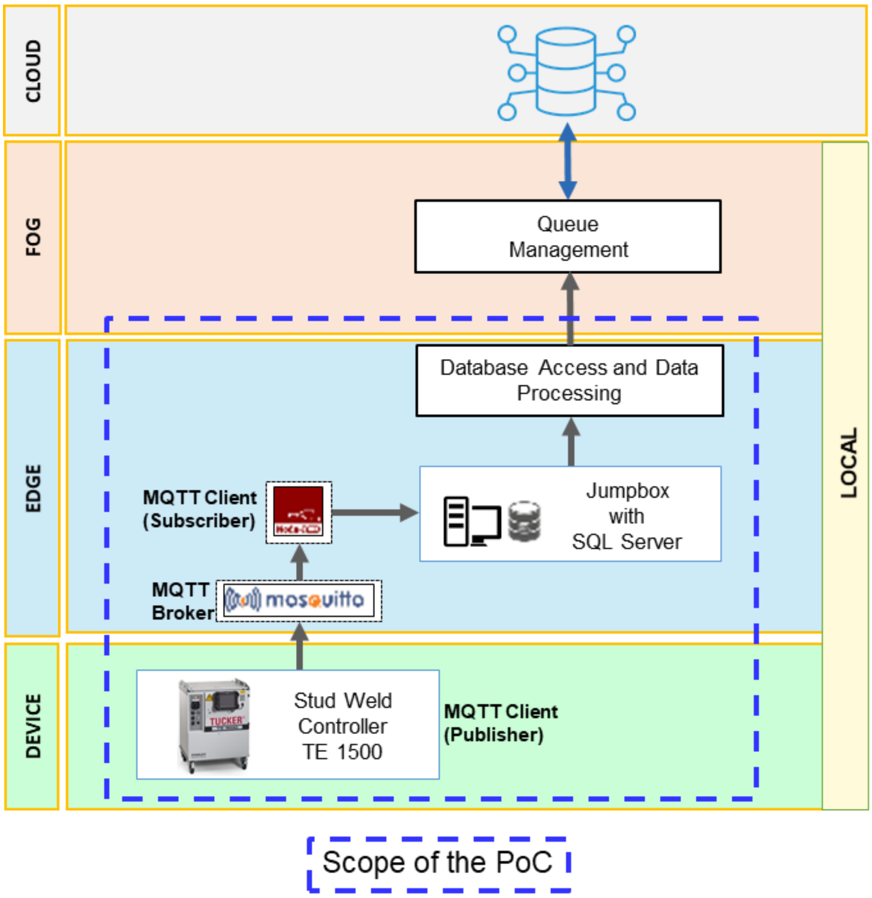
In the Welds table all parameters referring to the bolt welding process are stored. It consists of 147 fields of different types, among which there are 45 values that change, so it can be associated with values of each instant of welding and 102 that are always constant, which refer mainly to maximum and minimum values allowed or predetermined values of beginning and end, among many others.

2.4 TE Controllers

The Tucker TE family is installed only in three OV sites.

2.4.1 Connectivity possibilities

TE family includes the possibility to communicate by means of the MQTT (Message Queuing Telemetry Transport) protocol. Therefore, no other vendor software (as DCE Link) is required.

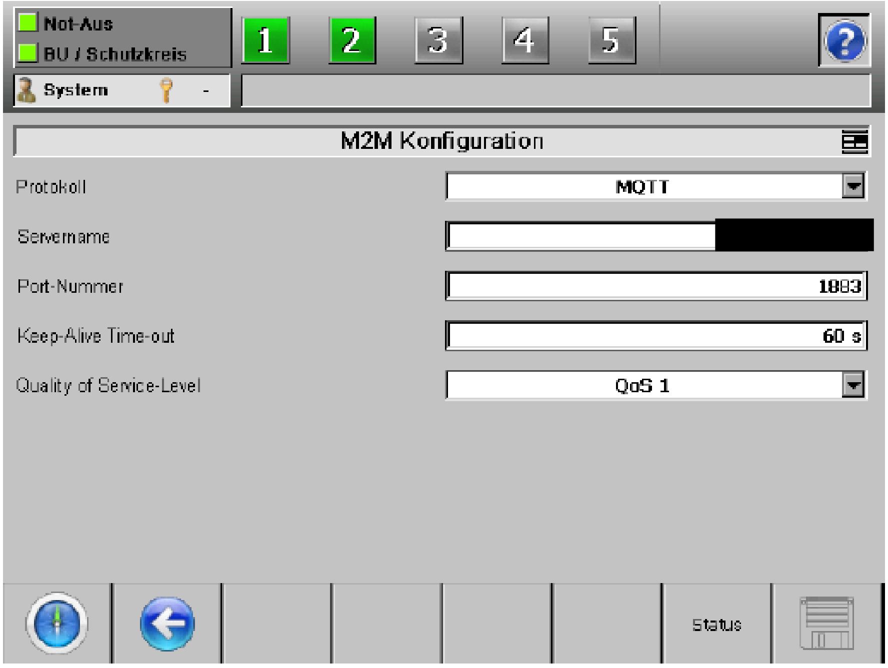


2.4.2 Software installation for TE controllers

As this is a more modern device, it is possible to establish a connection using the MQTT protocol.

The installation was performed at Ellesmere Port from a "Jumpbox" computer, which has connectivity to the TE controller.

Once the firmware has been updated, it is possible to select the MQTT configuration option on the device, as shown in the following figure.



Following the 'MQTT interface for TE Controller' manual, a device was configured. Below is an example of the steps required to configure the device.

2 Configuration
2.3 Configuration of the CPM Ethernet Adapter

- IP address:
- Netmask
- Gateway
- DNS
- Host Name
- Domain Name
- DHCP
- Service activation for onboard adapter

2.4 Definition of a device name

- **Device name:**

2.5 Configuration MQTT interface

- **Protocol:** MQTT
- **Servername:**
- **Port Number:** 1883
- **Keep-Alive Time-out:** 60
- **Quality of service-Level:** QoS 1

3 Available MQTT topics

- TEDevice/TE1500/DEvice

En el jumpbox de Ellesmere Port, con sistema operativo Windows, se instaló Mosquitto(1) y Node-RED(2) para poder realizar la conexión vía MQTT.

(1) The MQTT broker standardized by Groupe PSA is Rabbit MQ, which inevitably requires authentication (username and password) from customers. In the case of the Tucker TE welding controller, there is no menu to formalize such authentication, so it was decided to turn to Mosquitto, another MQTT broker that does not require it.

(2) Node-RED is a flow-based development tool for visual programming developed originally by IBM for wiring together hardware devices, APIs and online services as part of the Internet of

Things. On our test Node-RED is an MQTT client that subscribes to the topics published by the Tucker TE device

Using Node-Red, we subscribe to the topic the messages arrive at and store them in a CSV file.

Both the different topics available and the description of the measurements that are collected are in the manual 'MQTT interface for TE Controller'.

The list of accessible topics follows this structure:

- Topic Level TEDevice:
  This topic level is always "TEDevice" and makes it possible to filter all TE stud weld controllers with the help of wildcards.
- Topic Level [TEDeviceType]:
  The topic level "[TEDeviceType]" contains the TE device type and is named after the device type:
  - TE1500
  - TE1500AC
  - ...
- Topic Level [DeviceID]:
  The [DeviceID] is used to uniquely identify the device in the topic structure and corresponds to the device name.

Invalid characters (, / ',, #' and, + ') are replaced by' _ '.

To ensure that each device has a unique device name and therefore a unique device ID.

The following sections describe which MQTT topics are provided by the TE controller.

- MQTT status
  - The topic "TEDevice / [TEDeviceType] / [DeviceID] / MQTTStatus" transfers the connection status of the welding machine to the MQTT broker. Data may only be evaluated
  - If it is "online". If the MQTT status is "offline", the data is not valid and must not be evaluated.
- Measured data Parameters in the XML measurement data
  - The available measurement data parameters are in the MQTT topic
  - "TEDevice / [TEDeviceType] / [DeviceID] / LastWeld / MeasurementData" of the corresponding device is output as key value pairs in XML format.

Although no table with real data has been obtained, as it was not possible to establish a connection with the controller, a detailed list of the fields collected is included in Annex A. The data is saved with an XML structure, according to the manual.

2.4.3 Device Data Analysis

The MQTT Service from Tucker TE1500 provides the following topics (sets of information)

- TEDevice / [TEDeviceType] / [DeviceID] / MQTTStatus transfers the connection status of the welding machine to the MQTT broker. Data may only be evaluated
- TEDevice / [TEDeviceType] / [DeviceID] / LastWeld / MeasurementData of the corresponding device is output as key value pairs in XML format. There are 99 measurement data parameters available. Following is the beginning of the table.

Key	Description
DeviceName	Device name
DeviceType	Device type - TE1500 - TE1800
DateTime	Date and time of the welding yyyy-MM-dd HH:mm:ss
Date	Date of the welding yyyy-MM-dd
Time	Time of the welding HH:mm:ss
CarbodyID	Car body identification
StudID	Stud ID (Weld program)
...	....

3 Sealing and gluing (SCA)

3.1 Introduction to the technology

Sealing is a method that allows the closure of peripherals by means of a mas-tica substance preventing leaks from occurring. It is also responsible for joining two or more materials that may or may not have the same characteristics through an adhesive substance.

3.2 Device inventory

3.2.1 Connectivity possibilities

The models of the devices with which we are going to work are the SYS6000, which is presented with and without the vision system called Quiss. Regardless of the presence of the Quiss system, there are two different ways to access the data generated by the controllers:

- Using a software supplied by Atlas Copco (SCA) called PDM.
- Bakups

Since it was not known with certainty when it would be possible to install the software, it was decided to start by collecting the data through backups and, where possible, to establish the connection to the welding controller through the software.

3.3 Device access method

There are two ways of retrieving data from SYS 6000:

- MANUAL
  A system backup can be extracted to a USB stick. Contents of the backup can be configured. Data may include parameters, faults, application logs, etc. This method is neither adequate for a real-time monitoring nor for a constant data streaming for BI & DA purposes

- AUTOMATIC
  By means of PDM, a data logging system from Atlas Copco*
  This is the most suitable method for both a real time monitoring and a data streaming for BI & DA purposes.

3.3.1 Software PDM

PDM new application that has not yet been commercially deployed for SCA SYS 6000. PDM consists of two elements:

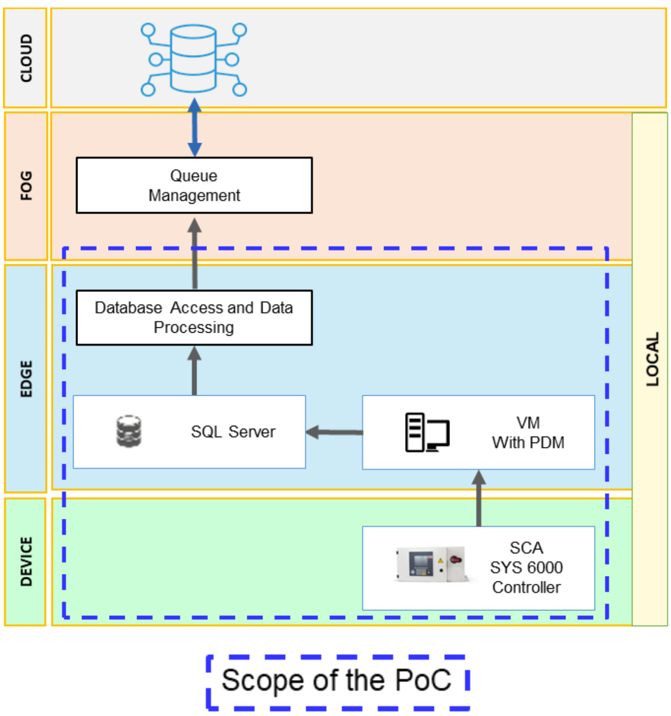
- APPLICATION
  It requires a Windows Server machine

- DATABASE
  Must be SQL Server. It may reside on the same machine as the application, or on a separated one

The software was developed by Atlas Copco (the current owner of SCA Schu-cker) and allows connection to SCA controllers and extraction of process data. It was necessary for a technician from the company to install the PDM.

The requirements are described in the manual "PDM_SpecSheet_v.2" but it is important to emphasize that it is possible to install the database on one machine and the PDM server on another. In our case, the database was installed on a laboratory computer and the server on a virtual machine running Windows Server 2014.

La arquitectura que se propone, tras conocer los requerimientos necesarios, es la siguiente:



3.4 Proof of Concept

In a first approach, an attempt was made to find out how viable it was to make a connection via the remote desktop to the computer integrated with the devices that have a vision system to be able to make backups. Three ways of doing this were considered.

The first one was to try to access remotely the computer that has the device incorporated, but it was discarded due to the fact that those responsible for maintenance feared that there would be some conflict between our connection and the one that is controlling the machine. The second option was to check if the software offered the possibility of automating the back-ups. Both options were discarded because they were not viable, so we opted for the third option, which is to make backups manually.

The data was obtained from 16 different computers.

The data capture period is limited by the maximum number of rows that can be stored in each table and is 5000.

3.5 Device Data Analysis

SCA SYS 6000 keeps record of the parameters and a log of all dispensing cycles and faults.

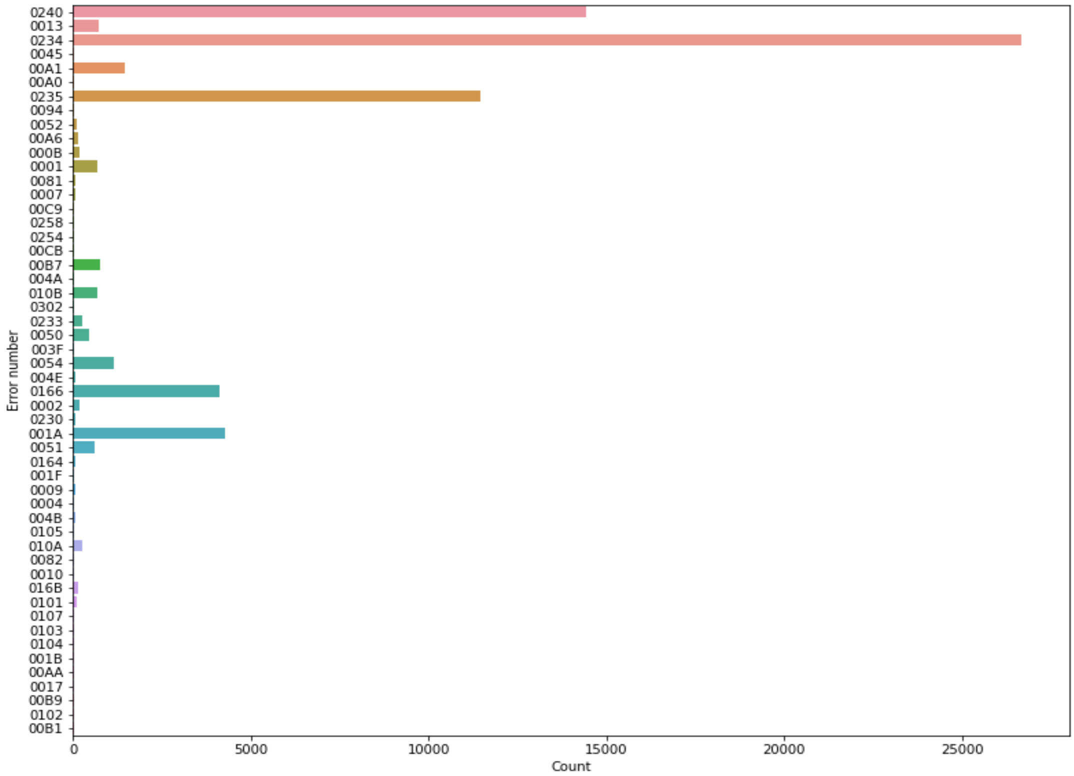
The dataset is practically the same either if it retrieved through a manual backup or through PDM.

The backup contains the following files within the Statistic folder:

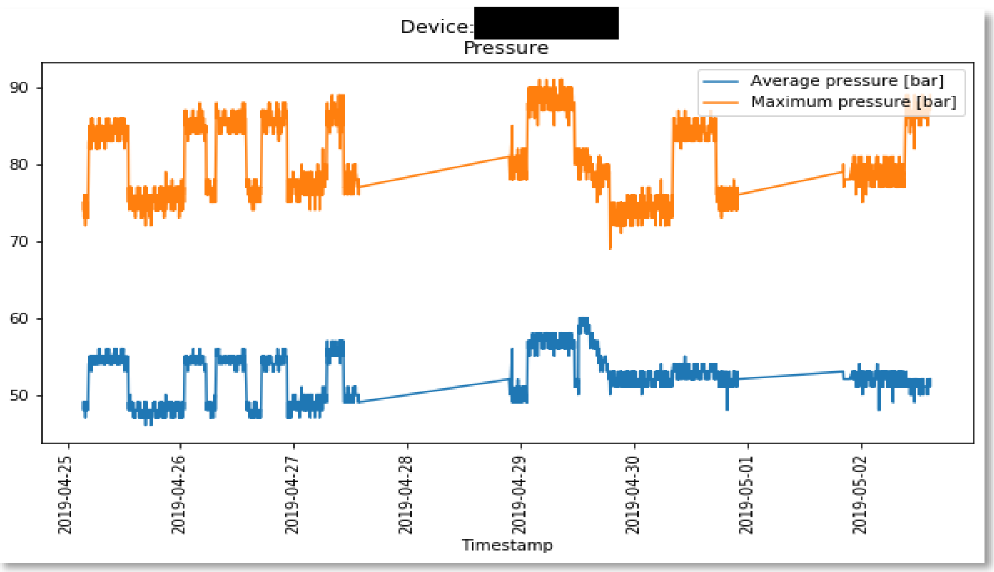
- **StatisticApplication**
  Contains 29 fields with detailed information of each dispensing cycle: timestamp, program, result, pressure, volume, temperature, etc…
- **StatisticCounter**
  Contains one record with 29 fields that summarizes the counts
- **StatisticError**
  Provides a detailed errors and faults log with 13 fields
- **StatisticLog**
  Provides a detailed log of main events log. It contains 5 fields
- **StatisticParams**
  Provides a log of all changes in the process parameters. It contains 8 fields
- **StatisticApplicationBeads**
  Provides a log of all beads that have been applied. It contains up to 260 fields.
- **StatisticSystemDiagnose**
  Provides a description of all diagnostics. It contains 16 fields
- **StatisticBarrelChange**
  Provides a log of all barrel changes. It contains 11 fields

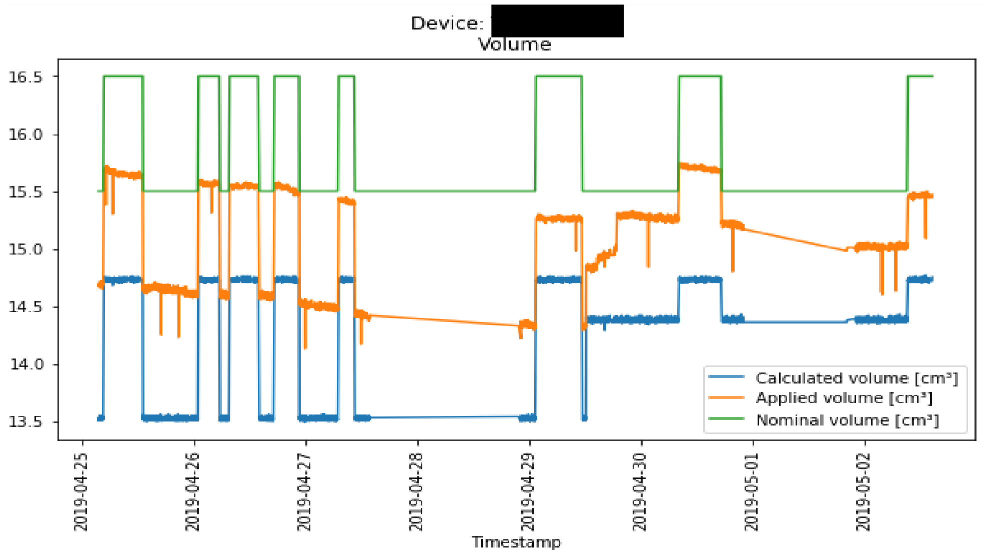
Data was collected from twelve different controllers, which store in different folders data referring to the system configuration, the different parameters for each car model and several files in CSV format where the statistics are found.

There are two files that make reference to the failures, a file with XML format in which all the possible errors are described and a hexadecimal number is associated to them; and a table that consists of nine columns where the failures that have occurred during the capture period are shown. In the following figure a summary is made, and it can be appreciated that the greater quantity of failures is produced due to failures detected by the vision system (0240, 0234, 0234).



In a different table the data concerning the application of the mastic is stored. This table consists of 23 fields showing parameters associated with the volume and pressure of the application, among others. In the figures you can see the curves of the collected values.




Device: [redacted]
Volume
16.5
16.0
15.5
15.0
14.5
14.0
13.5
Calculated volume [cm³]
Applied volume [cm³]
Nominal volume [cm³]
2019-04-25
2019-04-26
2019-04-27
2019-04-28
2019-04-29
2019-04-30
2019-05-01
2019-05-02
Timestamp