POLITECNICO DI TORINO

Department of Management and Production Engineering
Master of Science in Engineering and Management

Application of unsupervised machine learning for resistance spot welding

Supervisor:
Prof. Giulia Bruno

Candidate:
Roxana Duma

Co-supervisor:
Gabriel Antal

A.A 2023/2024

Contents

List of figures

List of Tables

Abstract

The quality of resistance spot welding (RSW) is crucial in several sectors. For instance, one modern car contains up to 7000 spot welds. By examining electrode force and displacement signals, this work proposes an unsupervised machine learning method for the RSW process. The application of classical supervised algorithms requires labeled data which often are costly to gather. This solution solves the problem by potentially enabling automatic labeling. Therefore, the insights gained from the application of data clustering can be used then to improve the performance, robustness, efficiency, and interpretability of a subsequent supervised modeling strategy.

An experimental campaign of several spot welds was done to collect force-displacement data that were recorded by high-frequency monitoring devices, allowing for the extraction of crucial characteristics. These features were subjected to clustering methods, specifically K-means++, Hierarchical Clustering, and DBSCAN with the aim of distinguishing between weld affected by expulsion and not.

Keywords: Machine learning; Resistance spot welding; Unsupervised learning; Clustering; Expulsion; Quality.

Introduction

Resistance Spot Welding (RSW) is widely utilized in manufacturing, particularly in the automotive and aerospace industries [1]. This process is popular due to its simplicity but, traditionally, it requires meticulous maintenance and costly destructive testing to ensure quality welds. Industry 4.0 has prompted corporations to digitize and integrate both physical and digital resources. To achieve Industry 4.0, smart manufacturing requires integration across several areas, such as Artificial Intelligence (AI) and all process management phases. In this regard, the use of sensors in the manufacturing processes has considerably increased, for example in monitoring, quality appraisal and maintenance tasks. Those tasks use process signals to forecast the state of equipment components or process outcomes, allowing for suitable interventions. Sensor technology, the Internet of Things (IoT), and Big Data have enabled the change of the process. The retrieved raw data can be analyzed using AI to provide insightful results. Machine Learning (ML) and Deep Learning (DL) algorithms are widely used nowadays.

Since, in the literature, there are just a few works on unsupervised ML regarding RSW, the goal of this thesis is to employ data obtained from a series of laboratory experiments to assess the occurrence of the expulsion phenomenon and automatically divide the welding points into two distinct groups based on expulsion presence.

Expulsion, also known as spatter or flash, describes molten metal that flows from the weld spot during the RSW process. It is an undesired phenomenon because it may reduce weld quality and strength, creating porosities in the welded spot, and causing other damage to the welded pieces.

Spatter is a common indicator of challenges during the welding process, and it can be employed as a quality indicator for process monitoring.

In this work different clustering techniques have been implemented and compared, showing that unsupervised ML leads to several advantages in case of RSW, among which process monitoring and automatic labeling of data.

The present thesis work is structured into 6 chapters. The first chapter begins with a comprehensive theoretical foundation that establishes the groundwork in artificial intelligence, machine learning methods, and data preprocessing techniques. The second introduces the fundamentals of resistance spot welding. Following, in the third chapter a literature review and bibliographic research section examines the current state of the art, analyzing existing methodologies and identifying areas for contribution. The work then moves to the RSW case study described in chapter four, presenting the practical application and defining the research objectives. Chapter five handles details about the technical implementation, primarily focusing on Python and the other relevant tools used throughout the project. Chapter six concludes the thesis with the application of clustering techniques, presenting the experimental results and their analysis, demonstrating the practical impact of the research. Finally, conclusions and recommendations for future research are presented.

1. Artificial Intelligence

Artificial intelligence (AI) is the technology that makes it possible for computers and other devices to reproduce human autonomy, creativity, problem-solving, learning, and comprehension [2].

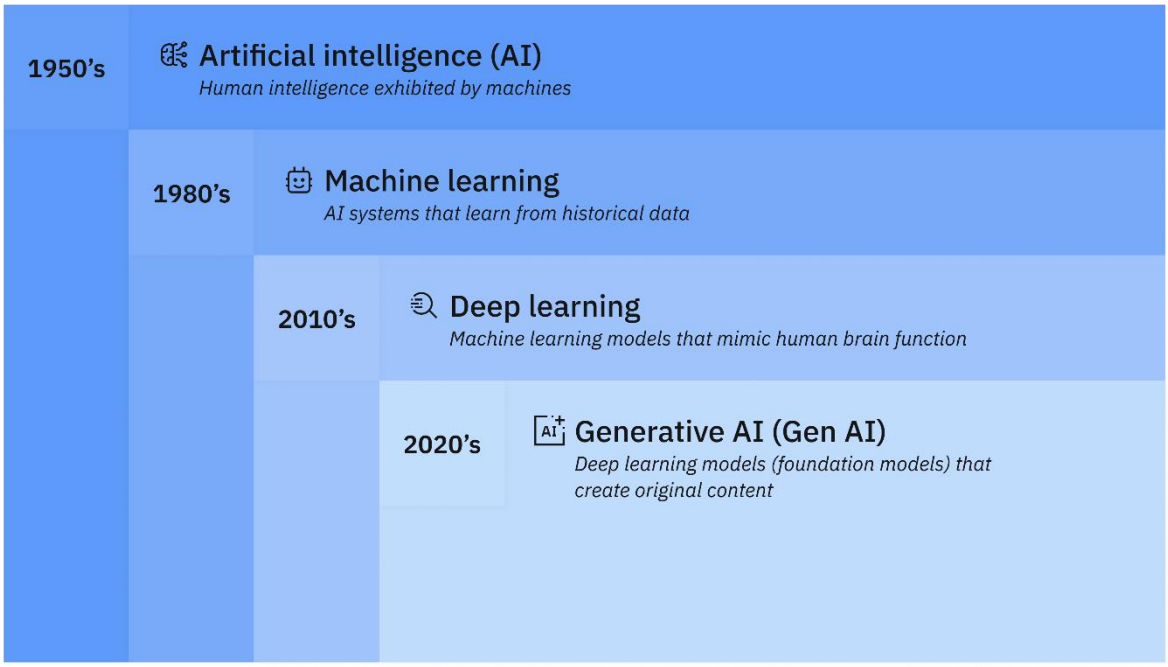

Figure 1 - Artificial Intelligence.

1.1 Machine Learning

Machine learning (ML) is a branch of artificial intelligence (AI) and computer science that focuses on the using of data and algorithms to enable AI to imitate the way that humans learn, gradually improving its accuracy. [3]

1.2 Deep Learning

Deep learning is a subset of machine learning that uses multilayered neural networks, called deep neural networks, that more closely simulate the complex decision-making power of the human brain [4].

1.3 Machine Learning Methods

- **Supervised machine learning (SL)** – The program "learns" from a training set that is part of the dataset. This set serves as "training examples" for the machine to achieve a specific outcome when new data is presented [2].
- **Unsupervised Learning (UL)** – there is not a training set provided. Instead, the data is clustered into different classes by the algorithm [2].

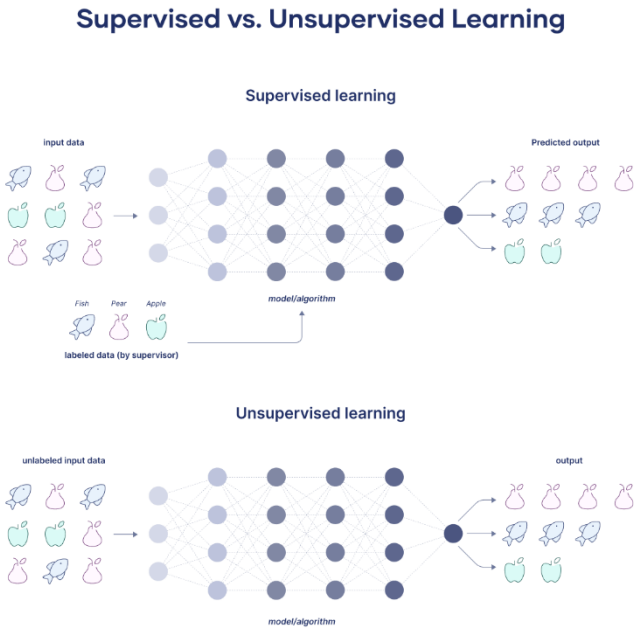

Figure 2 - Supervised vs. Unsupervised Learning.

Unsupervised learning models are used to perform three major tasks: clustering, association, and dimensionality reduction.

Given that the interest of this thesis is focused on unsupervised methods, the supervised ones are not explicitly treated.

1.3.1 Clustering

Clustering is a ML technique that organizes unlabeled data based on similarities or differences [5]. Clustering techniques are used to group raw, unclassified data objects based on information structures or patterns.
The main techniques foreseen for this work are Centroid-based, Hierarchical, and Density-based clustering [5].

Centroid-based clustering

Centroid-based clustering is a clustering technique that partitions or divides a dataset into similar groups based on the distance between their centroids. The centroid, or center, of each cluster is the mean or median of all the points in the cluster, depending on the data [5].

Hierarchical clustering

Hierarchical clustering, commonly known as hierarchical cluster analysis (HCA), is an unsupervised clustering approach that falls into two categories: agglomerative and divisive. Agglomerative clustering is described as a "bottoms-up approach”[6].
Its data points are first isolated as individual groupings, and then they are iteratively blended based on similarity until one cluster is formed.
Divisive clustering differs from agglomerative clustering in that it utilizes a "top-down" approach [6]. In this scenario, a single data cluster is partitioned according to the disparities between data points [6].

Density-based clustering

Density-based clustering detects areas with a high concentration of points separated by empty or sparse areas. Density-based clustering may find clusters of any shape. This is especially useful when clusters are not characterized by a specific location or dispersion. Density-based clustering can also discriminate between data points that belong to a cluster and those that should be classified as noise [5].

Density-based clustering is very effective when dealing with noisy or outlier datasets, or when the number of clusters that the data contains is not known prior.

1.3.2 Clustering algorithms

K-Means

The K-Means clustering algorithm is a popular centroid-based clustering technique. K-Means assumes that the center of each cluster defines the cluster using a distance metric, typically Euclidean distance, to the centroid [7]. To start the clustering, it is necessary to specify a number of predicted clusters, which constitutes the 'K' in K-Means, and the algorithm searches the data for reasonable groupings that match that number. The ideal K clusters in a given dataset are determined by iteratively minimizing the total distance between each point and the assigned cluster centroid.
To guarantee that the clustering results are significant and practical, determining the ideal number of clusters for K-Means clustering is essential. There are a number of methods for figuring out how many clusters are appropriate [7]. A few popular techniques:

- **Elbow Method**: It consists in plotting the explained variation as a function of the number of clusters and using the elbow of the curve to determine the number of clusters to use[7]. But the concept of the "elbow" is vague, and thus is known to be unreliable.
- **Silhouette (clustering)**: Silhouette analysis gives information about the distance between the generated clusters and it assesses the quality of clustering [7].
- **Gap Statistic**: Comparing the total intra-cluster variance for various choices of k with their predicted values under the null reference distribution of the data is known as the gap statistic. The value that produces the biggest gap statistic is known as the optimal k [7].

- **Davies-Bouldin index**: it is a metric used to quantify the degree of separation between clusters. A model with better separation is indicated by lower Davies-Bouldin index values [7].
- **Calinski-Harabasz index**: it assesses clusters according to their separation and compactness. Better-defined clusters are indicated by higher values of the index, which is computed as the ratio of between-cluster variance to within-cluster variance [7].
- **Rand index**: by considering both sets of items that are accurately allocated to the same or distinct clusters, the Rand index determines the percentage of agreement between the two clusters. Greater similarity and higher grouping quality are indicated by higher values. Introduced by Hubert and Arabie in 1985, the Adjusted Rand Index (ARI) corrects the Rand Index by accounting for the expected similarity of all pairings resulting from chance. This provides a more accurate estimate [7].

After the optimal number of clusters has been identified, a two-stage iterative procedure based on the expectation maximization machine learning method is included in the following phase. Each data point is assigned to its nearest centroid in the expected step according to distance (often Euclidean). In the maximizing step, the centroid, or cluster center, is reassigned after calculating the mean of all the points for each cluster. Until the centroid positions converge or the maximum number of iterations is reached, this process is repeated. K-Means is a hard clustering strategy, which means that each data point is assigned to a separate cluster with no likelihood of membership.

The algorithm works best when the clusters are around the same size and there are no noticeable outliers or fluctuations in density across the data [8].

K-Means frequently performs poorly when the data is highly dimensional or when clusters are significantly diverse in size or density. It is also particularly sensitive to outliers because it tries to build centroids based on the mean values of all values in the cluster, making it subject to overfitting to include such outliers [8].

K-means++, A K-Means algorithm that optimizes the choice of the first cluster centroid or centroids, was created by researchers Arthur and Vassilvitskii and enhances the final cluster assignment's quality [8].

Selecting one centroid from the dataset is the first step in the K-Means++ method's initialization process. It determines how far each data point is from the nearest cluster center for each succeeding centroid. The probability that a location is proportionately distant from the closest centroid previously selected is considered while choosing the next centroid [8]. Iterations of the process are carried out until the selected number of cluster centers has been initialized.

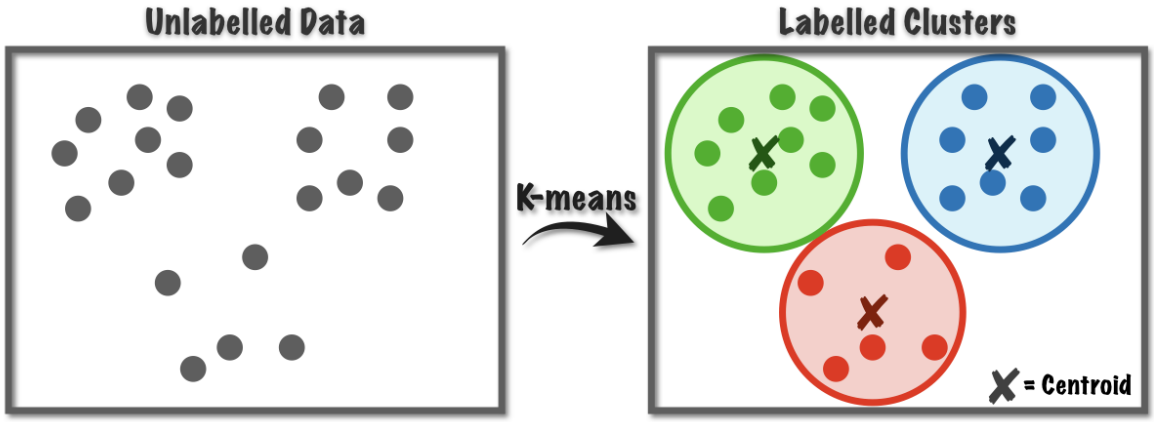

Figure 3 - K-Means.

DBSCAN

DBSCAN, which stands for Density-Based Spatial Clustering of Applications with Noise, is a potent clustering technique that clusters data points that are densely packed together [9]. DBSCAN is very helpful for exploratory data analysis since, unlike some other clustering methods, it doesn't need a predetermined number of clusters. Clusters are defined by the algorithm as dense areas divided by less dense areas. This method enables DBSCAN to detect outliers as noise and find clusters of any shape. DBSCAN is based on three main ideas [10]:

- **Core Points**: points that have a minimal number of other points (MinPts) within a given distance (ε or epsilon) [10].

- **Border Points**: points that are within a certain distance of a core point but do not have MinPts neighbors [10].
- **Noise Points**: points that are neither core nor border are known as noise points. They cannot be included since they are too far away from any cluster [10].

The two primary parameters used by DBSCAN are:

- **ε (epsilon)**: the greatest distance that two points must be apart in order to be regarded as neighbors [10].
- **MinPts**: The bare minimum of points needed to create a dense area [10].

Altering these settings may influence how the algorithm defines clusters, allowing it to adapt to various dataset types and clustering requirements.

Selection of ε (Epsilon)

The ε parameter specifies the maximum distance required between two places to be considered neighbors. To select the appropriate ε [9]:

- Apply domain knowledge: if a distance is relevant to the specific problem, utilize that as a starting point.
- K-Distance Graph: a more systematic approach. It consists in calculating the distance between each point and its k-th nearest neighbor (k = MinPts). The k-distances are plotted in increasing order, it can be noticed an "elbow" on the graph, when the curve begins to level off. The ε value at this elbow is typically a good choice [9].

Selection of MinPts

MinPts defines the minimal number of points needed to establish a dense zone [10].

- **General rule**: As a starting point, set MinPts = 2 * num_features, where num_features is the number of dimensions in the collection [10].
- **Noise consideration**: If the data contains noise or smaller clusters has to be discovered, reduce MinPts [10].

- **Dataset size**: For larger datasets increase MinPts to avoid forming too many tiny clusters [10].

Parameter selection can have a big impact on results. Experimenting with different settings and evaluating the generated clusters might help in determining the optimum match according to the dataset and problem.

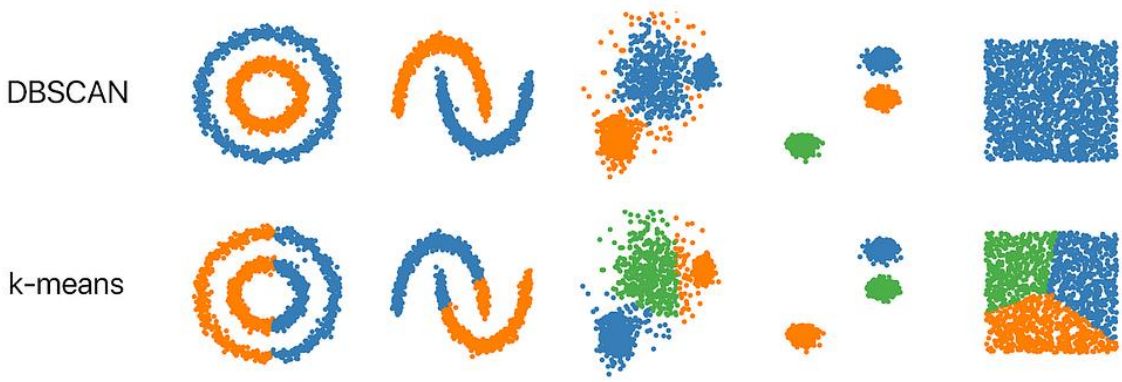

Figure 4 - DBSCAN vs K-Means.

Hierarchical Clustering

The goal of hierarchical clustering, also known as hierarchical cluster analysis or HCA, is to create a hierarchy of clusters. In general, there are two types of strategies [11]: agglomerative and divisive.

Agglomerative

This method is "bottom-up"; as one climbs the hierarchy, pairs of clusters are combined, with each observation beginning in its own cluster.

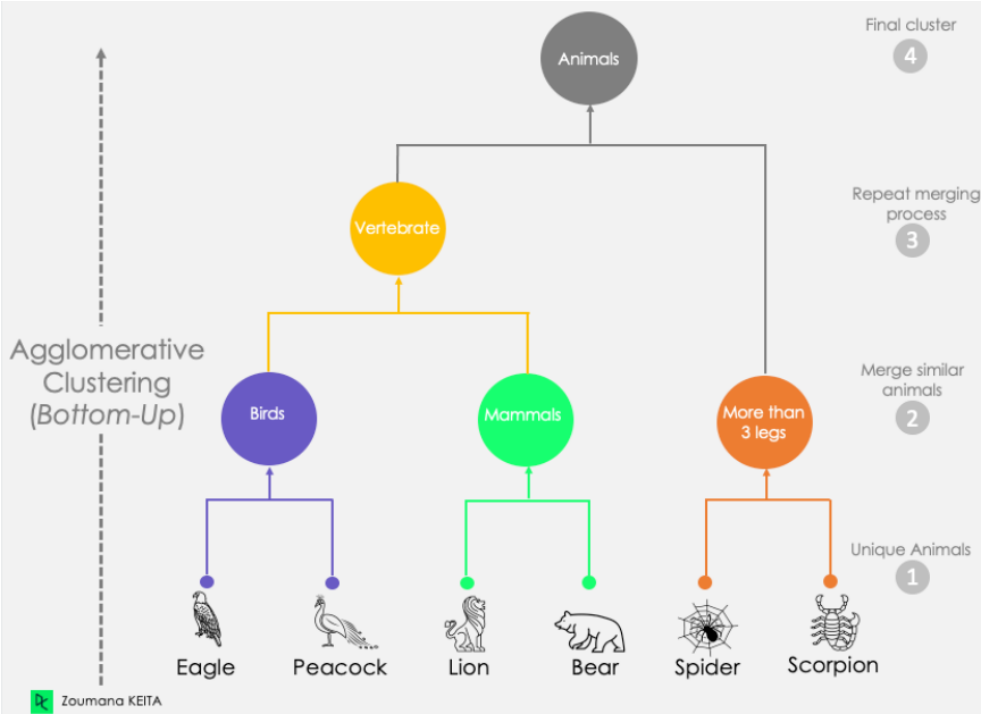

Figure 5 - Example of Agglomerative Clustering.

As shown in Figure 5, in this example the first step is considering each animal to be its unique cluster. Then three different clusters are generated based on their similarities (birds, mammals, and more than 3 legs). The two most similar clusters are combined to create the Vertebrate cluster. At last, the remaining two clusters are merged to create the final one, the Animals cluster [12].

Divisive

This method is "top-down": Splits are carried out iteratively as one descends the hierarchy, with all observations beginning in a single cluster.

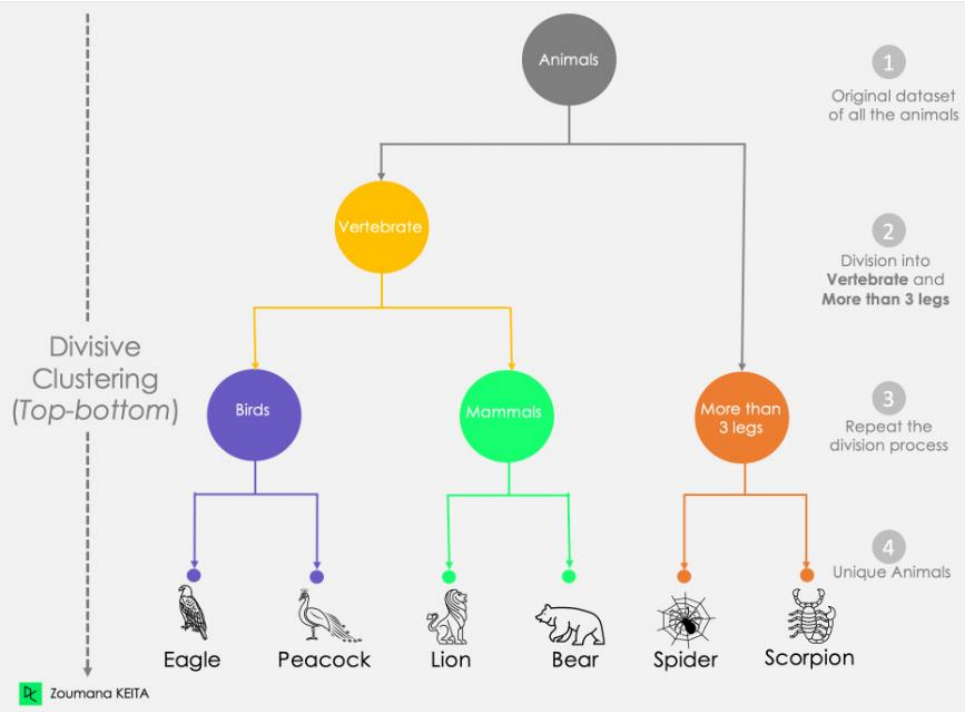

Figure 6 - Example of Divisive Clustering.

In this case, as shown in Figure 6, first the whole Animals dataset is considered as a whole. Then it is divided into two clusters. The division is applied iteratively until unique animals are created [12].

The hierarchical links between groups are shown by a dendrogram, a tree-like picture created via hierarchical clustering. The largest clusters, which contain all of the data points, are found at the top of the dendrogram, whereas individual data points are found at the bottom. The dendrogram can be cut at different heights to get a variable number of clusters. Iteratively combining or dividing clusters according to a metric of similarity or separation between data points produces the dendrogram. Until all data points are contained within a single cluster or until the predefined number of clusters is reached, clusters are continually split apart or combined.

To determine the optimal number of clusters, the dendrogram can be examinate and assess the height at which its branches form discrete clusters. The number of clusters can be ascertained by slicing the dendrogram at this height.

1.4 Data preprocessing

Data preprocessing is an important stage in Machine Learning that involves converting raw data to a clean and useful format [13]. This procedure assures that the data is appropriate for analysis and modeling, thereby boosting the performance and accuracy of ML models.

The main benefits of this procedure are:

- Improves data quality
- Handle missing data
- Normalize and scales data
- Eliminates duplicate records
- Handle outliers
- Enhance model performance

1.4.1 Feature engineering

Feature engineering is an ML method that converts raw data into a more useful set of inputs. Each input contains numerous qualities, known as features [14].

Providing relevant information to models improves their forecast accuracy and decision-making capabilities dramatically.

Feature engineering in ML and statistical modeling entails choosing, developing, converting, and extracting data features.

Key components include:

- creating features from existing data
- transforming and imputing missing or invalid features
- reducing data dimensionality using methods such as Principal Components Analysis (PCA)
- selecting the most relevant features for model training based on importance scores and correlation matrices

1.4.2 Feature selection

"Feature selection is the process of selecting the most useful features for building models in tasks like classification, regression or clustering" [15].

Unsupervised Feature Selection (UFS) methods do not require a supervised dataset [16]. According to Guyon et al. (2003) [17], Niijima and Okuno (2009) [18], and Devakumari and Thangavel (2010) [19], UFS approaches offer two significant benefits. They are unbiased and perform well when previous information is unavailable, and they can reduce the possibility of data overfitting when compared to supervised feature selection approaches, which may be unable to deal with a new type of dataset. Unsupervised Feature Selection methods can be divided into four main approaches (Alelyani et al. 2013 [20]; Dong and Liu 2018 [21]):

- **Filter method**: it analyzes features based on inherent aspects of the data, rather than utilizing a clustering algorithm to find relevant features. Filter methods are known for their fast and scalable performance.
- **Wrapper method**: this strategy focuses on identifying feature subsets that enhance the quality of the selected clustering algorithm. Wrapper approaches have large processing costs and are only compatible with specific clustering algorithms.
- **Hybrid methods**: Hybrid methods combine filter, and wrapper approaches to achieve a balance of efficiency (computational effort) and effectiveness (quality of the objective job utilizing selected features).
- **Statistical:** this type of feature selection entails employing statistical tests to assess the relationship between characteristics and the target variable (or between features themselves). Mutual information and correlation coefficients evaluate the relationship between variables, assisting in the identification of key qualities while excluding less important ones.

1.4.3 Dimensionality Reduction

While more data generally produces more accurate findings, it can have an impact on the performance of Machine Learning algorithms (for example, overfitting) and make it difficult to visualize datasets. Dimensionality reduction is a strategy used when a dataset contains too many features, or dimensions. It minimizes the amount of data inputs to a manageable quantity while retaining the dataset's integrity to the greatest extent possible. It is frequently utilized at the preprocessing data stage, and there are a few different dimensionality reduction algorithms that can be applied.

Principal component analysis (PCA)

Principal Component Analysis (PCA) is a dimensionality reduction technique that converts high-dimensional data into a new set of uncorrelated variables known as principal components. These components are arranged in order of the amount of variance they explain in the data, with the first component accounting for the largest variance. PCA works by identifying the directions in the data space that maximize the variance and projecting the data onto these directions. The technique necessitates data standardization first, and the number of components retained is typically determined by the cumulative explained variance ratio. PCA is very effective for feature extraction and data visualization, but because the generated features are combinations of original characteristics, they can be difficult to interpret.

1.5 Evaluation metrics

Unlike supervised learning, where there is actual data to evaluate the model's performance, clustering analysis lacks a strong evaluation measure that can be used to compare the results of various clustering algorithms.
Furthermore, because K-means requires K as an input and does not learn it from data, there is no correct answer to the number of clusters that should be used in any given situation. Domain knowledge and intuition can be useful at times, although they are rarely required. It can be tested how well the models work in the cluster-predict methodology using different K clusters because they are employed in downstream modeling.

The **Elbow** approach provides an estimate of a good k number of clusters based on the Sum of Squared Error (SSE) between data points and their associated cluster centroids. The "k" is picked at the point where SSE begins to flatten and create an elbow.

$$SSE = \sum_{i=1}^{K} \sum_{x \in C_i} dist^2(m_i, x)$$

where:

- x is a data point in a cluster C_i
- m is the representative point for cluster C_i

For each point, the error is the distance to the nearest cluster. Given two clusters, the one with the smallest error is chosen. One easy way to reduce SSE is to increase K, the number of clusters. A good clustering with smaller K can have a lower SSE than a poor clustering with higher K.

The **Silhouette** score indicates how similar a data point is to its own cluster when compared to other clusters. A higher Silhouette score value implies that the data point is better suited to its own cluster but poorly matched to other clusters.
The best score value is one, while the poorest is -1.

The silhouette coefficient for a sample is defined as:

$$\frac{(n - i)}{max(i, n)}$$

Where:

- n is the distance between each sample and the nearest cluster that the sample is not a part of.
- i is the mean distance within each cluster.

Since for this work there is prior knowledge about how the data points should be grouped, also the following performance metrics can be used for clustering evaluation. Specifically, the metrics examined are Adjusted Rand Index (ARI) and Purity.

Adjusted Rand Index (ARI)

The Rand Index calculates a similarity measure between two clustering by taking all pairs of samples and counting whether they are assigned to the same or different clusters in the expected and actual clustering[22]. The formula of the Rand Index is:

$$RI = \frac{\text{Number of Agreeing Pairs}}{\text{Number of Pairs}}$$

Figure 7 - Rand Index formula.

The raw RI score is then "adjusted for chance" to the ARI score using the following scheme[22]:

$$ARI = \frac{\text{RI - Expected RI}}{\text{Max(RI) - Expected RI}}$$

Figure 8 - The Adjusted Rand Index formula.

The Adjusted Rand Index spans from -1 to 1, with 0 representing random labelling, 1 representing identical clusters, and negative values representing very poor labelling.

Purity

Purity is a metric used to assess the quality of clustering findings, particularly when the ground truth labels for the data points are known. It measures the extent to which the clusters produced by a clustering algorithm match the true class labels of the data.

2. Resistance Spot Welding

Resistance Spot Welding (RSW) is a commonly utilized joining technology in many industries like, for example, the automotive one [1]. The process involves pressing two electrodes against the sheets to weld for a set period of time (i.e., squeeze time). Then, the electrodes and sheet stack are subjected to a high current flow for a predetermined time, known as the welding time. The Joule effect generates heat, which is used to melt the sheets locally and form the weld nugget.

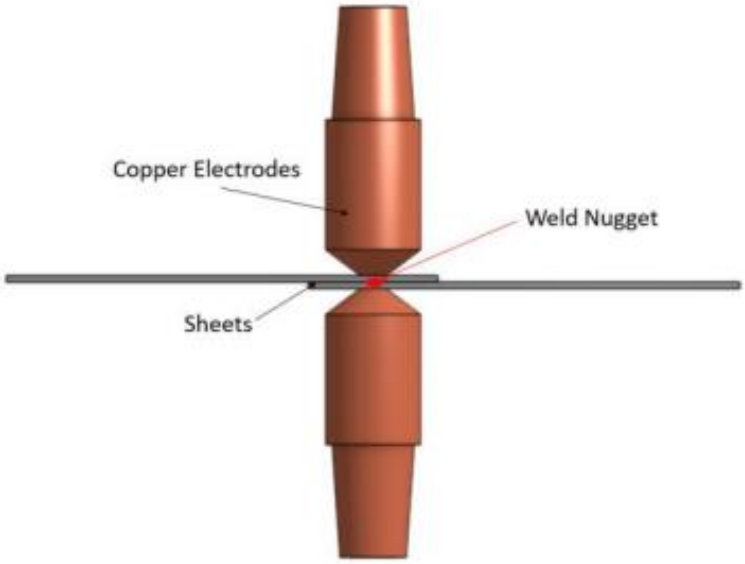

Figure 9 – RSW process schematization.

The process's features include ease of implementation, cheap cost and automation, making it suitable for a wide range of applications [23].

2.1.1 RSW Principles

The basic idea behind Resistance Spot Welding (RSW) is the combination of applied pressure with electrical resistance heating.
Joule's Law states that when an electric current passes through two or more overlapping metal sheets clamped between copper electrodes, the metal intrinsic electrical resistance produces heat.
The equation below defines how much energy is transferred to the welding system [24]:

$$Q = \int_{t1}^{t2} I^2(t)\ R(t)\ dt$$

Where:

- Q indicates heat energy generated throughout the process.
- I (t) is the welding current.
- R (t) is the sheet metals dynamic resistance.
- $t1$ and $t2$ represent the beginning and end of the welding session.

The interface between the sheets has the largest resistance, which results in localized heating there. A molten pool known as a weld nugget is produced by the generated heat, which usually reaches temperatures beyond the melting point of the material (about 1500°C for steel). Concurrently, the electrodes applied pressure aids in preserving contact and keeping the molten metal contained.
When the current is turned off, the nugget quickly hardens under constant pressure, establishing a robust metallurgical link between the sheets. The copper electrodes also act as heat sinks, keeping the weld localized and cool enough to make a robust junction.

2.1.2 Electrodes

Resistance Spot Welding (RSW) electrodes provide a variety of important tasks, which can be classified as mechanical, electrical, and thermal.
Mechanically, they exert and maintain pressure on the workpieces, guaranteeing appropriate contact between metal sheets and controlling the molten metal pool to prevent expulsion.
Electrically, the electrodes conduct and focus the current, resulting in localized heating at the desired welding surface.
Thermally, they operate as heat conductors, dispersing heat away from the surface and regulating the cooling pace of the weld nugget.
The electrodes are often made of copper alloys, which have great electrical and thermal conductivity as well as mechanical strength and wear resistance.
Regular maintenance is critical because electrodes deteriorate, distort, and become contaminated during the welding process, necessitating periodic inspections of electrode condition, cooling water flow, and proper alignment to ensure weld quality.

2.1.3 Welding Cycle

The Resistance Spot Welding (RSW) cycle is made up of multiple sequential time-based steps that govern the production of a quality weld nugget.
The cycle begins with the squeeze time, during which electrodes apply pressure to the workpieces to guarantee good contact and alignment.
This is followed by the weld time, during which electric current passes through the electrodes and workpieces, creating heat at their interface due to electrical resistance, forcing the metal to melt and form the welding nugget.
The current is subsequently shut off, resulting in the hold time (or forge time), during which pressure is maintained on the cooling weld to guarantee appropriate solidification and prevent faults such as porosity or voids.
Finally, the off time allows for electrode retraction and part repositioning before the next weld.

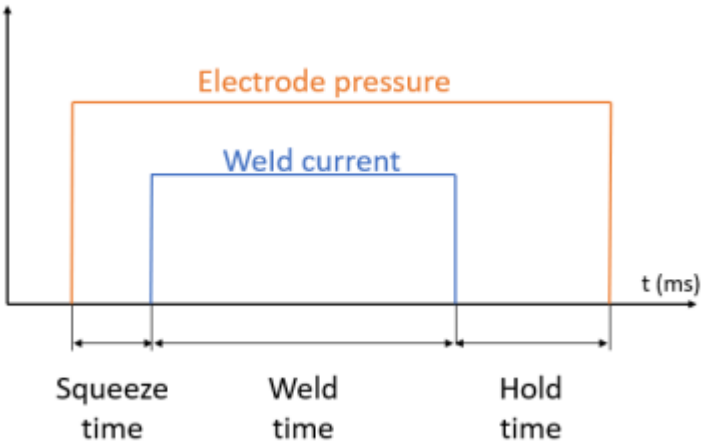

Figure 10 - Schematization of a spot weld cycle.

Some procedures may contain additional steps such as pre-weld and post-weld current pulses, or upslope and downslope durations for gradually increasing or lowering current to improve weld quality and reduce flaws.
The complete cycle often takes fractions of a second, making RSW ideal for mass production.

2.1.4 Weld Quality

Weld quality in Resistance Spot Welding (RSW) is defined by a mix of physical, mechanical, and metallurgical properties that ensure the joint structural integrity and performance. A high-quality spot weld has:

- an appropriate nugget size (usually 4-5 times the square root of the thinnest sheet thickness).
- suitable penetration depth.
- sufficient strength under both static and dynamic loading conditions.

Common weld faults include porosity, cracks, partial fusion, excessive indentation, and molten metal expulsion.
Environmental considerations such as zinc coating condition (in galvanized steels), surface cleanliness, and correct electrode maintenance have a substantial impact on the final weld quality. Regular quality control and process monitoring are required to ensure consistent weld quality in production environments.

Resistance Spot Welds can be evaluated using a variety of destructive and non-destructive testing methods, with a primary focus on nugget size, penetration, and strength properties.
Non-destructive testing methods include ultrasonic testing, which detects internal defects and measures nugget diameter; electrical resistance measurements, which monitor weld production in real time; and visual inspection for surface defects such as electrode depression, surface fractures, or expulsion [1].
Peel tests, chisel tests, and weld cross-sectioning for metallurgical inspection are instead the most popular destructive testing methods.

2.1.5 Expulsion

According to the ISO 17677-1 (2021) [25] definition expulsion, also known as a splash, spatter, or flash, is the phenomenon of molten metal particles being ejected at the electrode's point of contact with the welded material or at the sheets' faying surfaces. In particular, the expulsion between the faying surfaces of the workpieces is highly undesirable because it has a considerable impact on weld quality due to liquid metal loss. As a result, flaws like porosities within the nugget and significant indentation may arise from splash, causing the weldment's mechanical strength to be significantly reduced. Instead, expulsion at the electrode-material contact point can compromise surface quality and also electrode life, but only on the sheet surface; weld strength is unaffected.
An improper welding schedule is the most frequent reason for weld expulsion [26]. In industry, welding parameters are sometimes set close to the limits of flash, like high welding current, to achieve larger weld nugget sizes and meet quality criteria. The weld nugget is very important in influencing the general strength and mechanical qualities of the weld. Weldings are typically performed under splash conditions due to the variability of expulsion boundaries for characteristics, such as electrode wear. Reducing expulsion in RSW can help minimize nonconforming and low-quality welds. Monitoring and controlling expulsions are critical for adjusting welding conditions and reducing defects.

The principal causes of this phenomenon include:

- high welding current.
- insufficient electrode force.
- poor part fit.
- old electrodes.
- inappropriate timing parameters.

In order to lower the frequency of expulsion, it is crucial to be able to monitor it and adjust welding conditions or take other corrective measures. Existing publications indicate that the study of expulsion has received attention in two areas.
Studying its mechanism is the first step. For expulsion prediction and root cause analysis, a few mathematical models and criteria were put forth [27], [28].
The other focuses on evaluating and detecting ejection. It has been demonstrated that the majority of commonly used welding process signals, such as:

- **electrical signals**: including dynamic resistance, power input, and secondary voltage.
- **mechanical signals**: such as vibration, force, and displacement.
- **acoustic signals**: such as ultrasound and acoustic emission.

3. State of the art

The research procedure utilized to gather information for this thesis is described below. In order to create some personal knowledge on the topics, the essential definitions (e.g., resistance spot welding and machine learning technologies) were retrieved on the web, seeking for the most credible, informative, and trusted websites. The study was conducted in a top-down method, beginning with high-level keywords and progressing to more specific inquiries using the Scopus instrument.

Initially, the research aimed to provide a comprehensive overview of works on RSW, Expulsion, Machine Learning, and Clustering. Keywords were searched within the 'Article title, Abstract, and Keywords' of research articles, and then merged for more targeted results.

The following section describes some of the steps used during the research process:

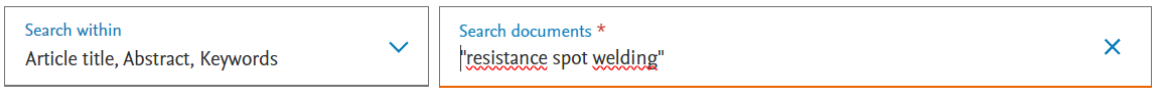

Figure 11 - Resistance spot welding search, Scopus.

A search so general led to 3,896 documents being found. An increasing number of articles have been developed in the last 20 years, reflecting the increasing interest in the subject. In addition, most of the documents found are from China, US and Japan. Talking about the application field, the main concerning areas are Engineering, followed by Material Science.

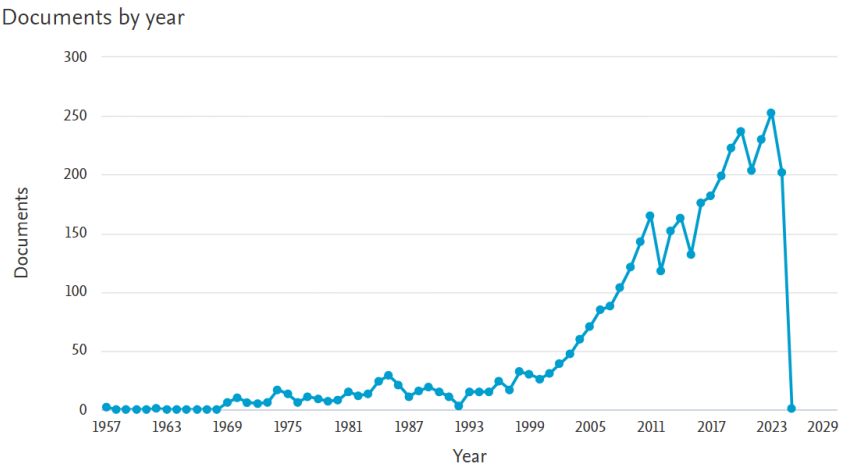

Figure 12 - Resistance Spot Welding articles per year.

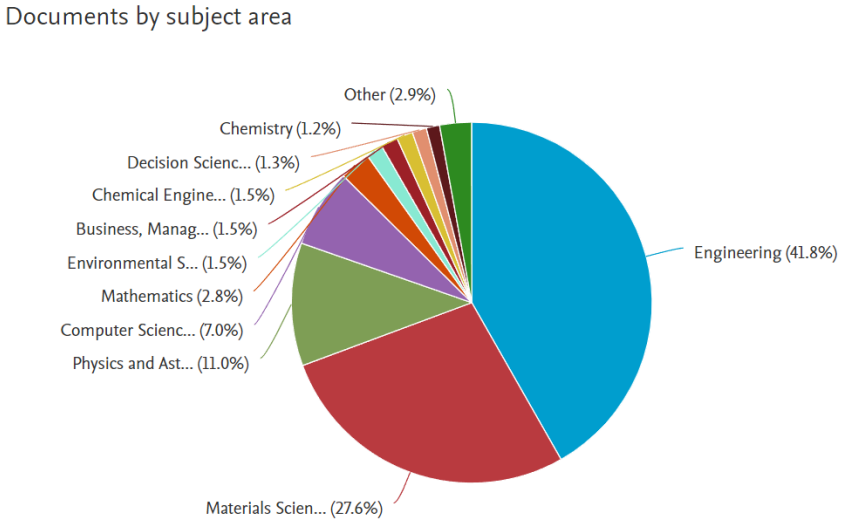

Figure 13 - Resistance Spot Welding articles by subject area.

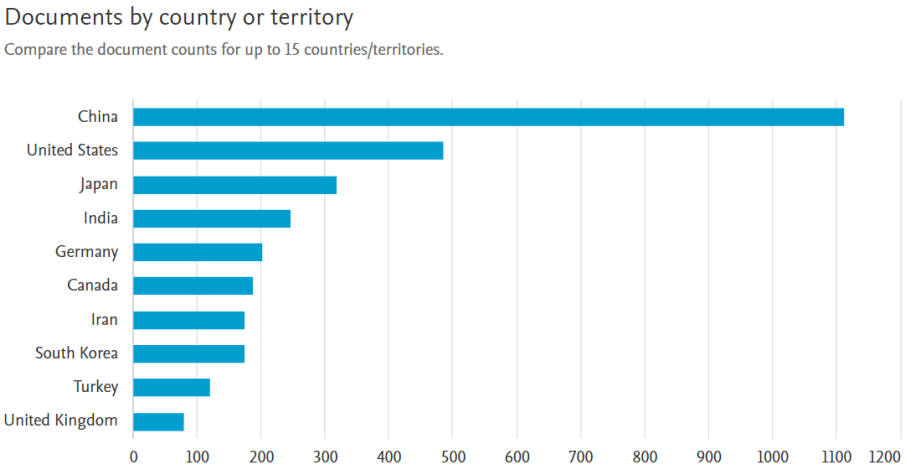

Figure 14 - Resistance Spot Welding articles per country.

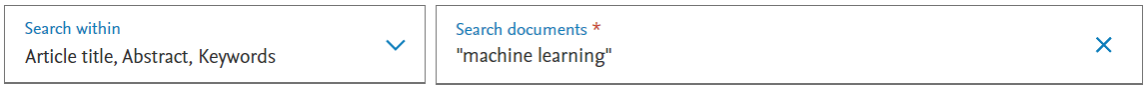

Figure 15- Machine Learning search, Scopus.

The second general search of "machine learning" shown in Figure 15 led to 706,447 documents being found. The number of documents per year saw an increase starting from 2003. Most of the documents found are from US, China and India. Talking about the application field, the main concerning areas are Computer Science and Engineering.

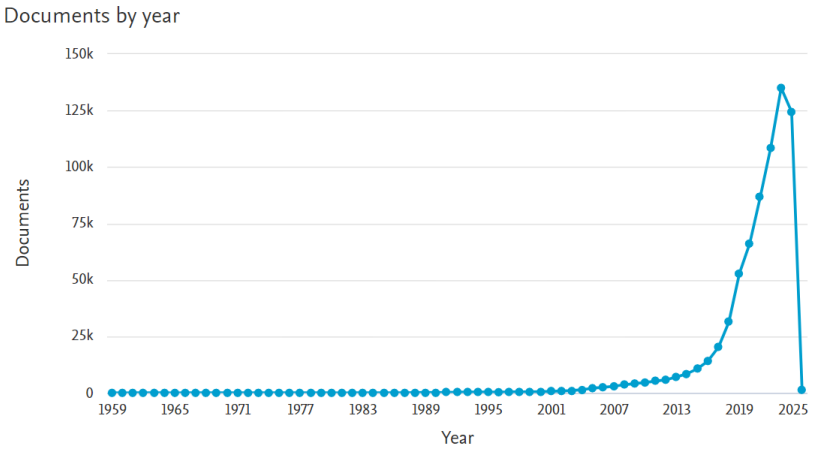

Figure 16 - Machine Learning articles per year.

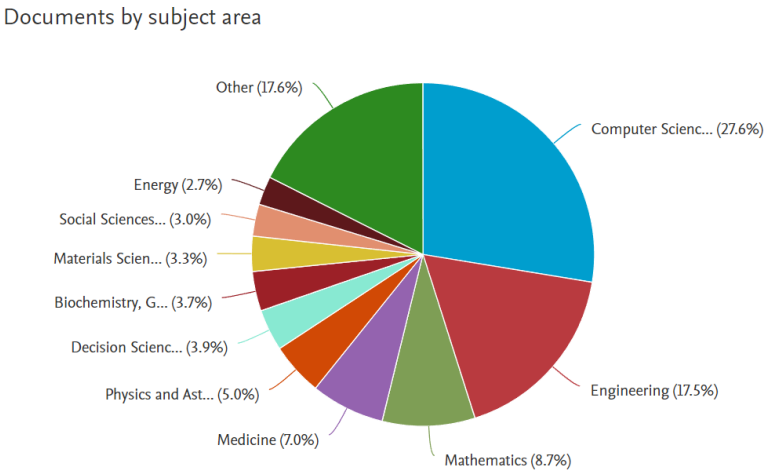

Figure 17 - Machine Learning articles by subject area.

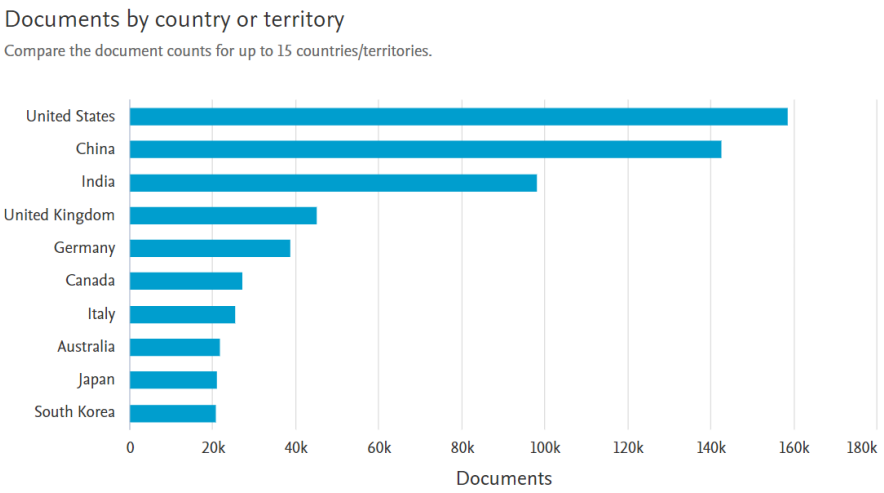

Figure 18 - Machine Learning articles per country.

3.1 Bibliographic research

Three specific queries have been inserted on Scopus in order to retrieve information regarding RSW and machine learning applications for quality assurance. The most relevant articles have been selected and resumed in the following tables. Documents from the last 2-3 years have been taken into consideration about the topics described afterwards.

One of the interesting aspects is that there are fewer results for clustering, since the majority of the papers focus on supervised algorithms. This result justifies the need for deeper studies regarding the unsupervised approaches and, therefore, this work.

Tables 1,2,3 show the titles, keywords and short descriptions for each article considered.

First query:

- Query: (“resistance spot welding” or “rsw”) AND “quality” AND “monitoring”
- 190 documents found.

The documents in Table 1 have been chosen as the most relevant, focusing on the use of machine learning and time series signals (e.g., electrode force, electrode displacement and dynamic resistance). These publications cover the most popular ways for quality control monitoring through ML in the resistance spot welding process. The works date back to the last two years.

Title	Keywords	Short description	Ref.
Dynamic resistance signal–based wear monitoring of resistance spot welding electrodes	Artificial neural networks; Dynamic resistance; Electrode wear; Resistance spot welding	The authors focused on predicting electrode wear in resistance spot welding for automotive manufacturing using dynamic resistance signals. They analyzed the relationship between dynamic resistance patterns and electrode tip geometry changes during consecutive welding processes. After establishing a clear correlation between resistance curves and tip geometries, they developed a machine learning model to predict electrode wear.	[29]
Quality monitoring for a resistance spot weld process of galvanized dual-phase steel based on the electrode displacement	Data analysis; Electrode displacement; Process monitoring; Quality assurance; Resistance spot welding	The authors developed a new methodology to monitor resistance spot welding quality using electrode displacement measurements on a galvanized dual-phase steel. Their approach builds on previous research and consists of three main steps: detecting manufacturing issues, evaluating weld expulsion, and monitoring nugget diameter. They used electrode velocity and indentation displacement measurements during both welding and hold times as key parameters for process evaluation.	[30]
Comparison and explanation of data-driven modeling for weld quality prediction in resistance spot welding	Neural networks; Process monitoring; Quality prediction; Resistance spot welding	The study focused on enhancing resistance spot welding (RSW) quality through process monitoring using neural networks. The researchers compared two modeling approaches: a feature-based Multilayer Perceptron (MLP) and a raw sensing-based convolutional neural network (CNN), while also evaluating the impact of electrical and mechanical sensing data on model performance. Both models successfully predicted weld quality metrics and detected expulsion using current and resistance signals, with improved performance when force and displacement measurements were included.	[31]
An Approach to Inline Monitoring of the Electrode	Data mining; electrode wear state	The research focused on developing an inline monitoring system for electrode wear in Resistance Spot Welding (RSW) for automotive manufacturing.	[32]

State in Resistance Spot Welding	monitoring; feature construction; industrial data analytics; resistance spot welding	Rather than using traditional predetermined intervals for electrode tip dressing, they developed a data-driven approach using dynamic electrical resistance measurements between dressing cycles to determine optimal dressing timing.	
Online evaluation of resistance spot welding quality and defect classification	dynamic resistance; medium-frequency direct current; mild steel; quality evaluation; resistance spot welding	The research developed an online monitoring system for resistance spot welding (RSW) quality evaluation and defect classification in mild steel welding. The researchers designed a finite impulse response low-pass filter using Blackman window function to process dynamic resistance signals, and studied how welding parameters affected nugget diameter and shear strength to determine optimal settings. They focused on detecting two main defect types - expulsion and incomplete fusion - by extracting time-domain features from dynamic resistance curves.	[33]
A machine learning approach for efficient and robust resistance spot welding monitoring	Deep learning; Machine learning; Nugget diameter; Resistance spot welding	The research developed an improved method for predicting weld nugget diameter in resistance spot welding using a combination of unsupervised deep learning and Gaussian process regression. The researchers used autoencoders to extract features from dynamic resistance curves, creating a low-dimensional representation of the process information. These features were then linked to nugget diameter predictions using Gaussian process regression. The new approach demonstrated higher prediction accuracy compared to traditional geometric attribute methods while maintaining low implementation costs.	[34]
Quality Monitoring of Resistance Spot Welding Based on a Digital Twin	digital twin; quality monitoring; spot welding; wavelet analysis	The research focused on applying digital twin technology to monitor resistance spot welding of 2219/5A06 aluminum plates of varying thicknesses. The researchers developed a digital twin system to bridge the physical and virtual aspects of the welding process, enabling virtual-real interaction and process optimization. To enhance information exchange within the digital twin system, they established a data acquisition system and proposed a real-time data processing method using wavelet threshold analysis. This approach aimed to overcome the limitations of physical models in monitoring the highly nonlinear coupled welding process quality.	[35]
Machine learning with domain knowledge for	Condition monitoring; Feature engineering;	The research developed machine learning pipelines for quality monitoring in Resistance Spot Welding using real production data instead of laboratory settings. The researchers created 12 different ML	[36]

predictive quality monitoring in resistance spot welding	Industry 4.0; Machine learning; Predictive maintenance; Quality monitoring; Resistance spot welding	pipelines combining four feature engineering settings with three ML methods (linear regression, multi-layer perception, and support vector regression). Their approach uniquely treated welding as a continuous process rather than independent events, enabling quality prediction of upcoming welds. The method incorporated engineering knowledge in both feature design and result interpretation	

Table 1- Research paper analyzed for Query 1.

The papers focus on monitoring welding quality through the prediction of different targets. All approaches mainly use supervised classification and regression approaches. For examples, in [29], [32] the electrode wear problem is treated while in [30], [31], [34], [35], the target variable is the welding nugget, being one of the most important quality features when dealing with RSW. Other works focuses on expulsion and incomplete fusion [30], [33]. Finally, one work selected predicts through different ML with feature extraction an aggregate welding quality parameter [36].

Second query:

- Query: (“resistance spot welding” or “rsw”) AND “machine learning”
- 48 documents

The publications from Table 2, on the other hand, illustrate how machine learning models may help not only in monitoring, but also for process optimization and quality assessment.

Title	Keywords	Short description	Ref.
Improving RSW nugget diameter prediction method: unleashing the power of multi-fidelity neural networks and transfer learning	Multi-fidelity neural networks; Nugget diameter prediction; Resistance spot welding (RSW); Transfer learning	The research developed a novel machine learning approach for predicting nugget diameter in RSW by combining low-fidelity simulation data with high-fidelity experimental data through transfer learning. They first trained a model using data from finite element simulations and design of experiments, then fine-tuned it with actual experimental data. This dual-fidelity approach demonstrated improved prediction accuracy while reducing the need for extensive experimental trials, offering a cost-effective method for predicting critical RSW process parameters.	[37]
Machine learning tool for the prediction of electrode wear effect on the quality of resistance spot welds	Artificial intelligence; Electrode degradation; Electrode wear; Machine learning; Predictive maintenance; Resistance spot welding	The study develops a machine learning tool to assess electrode wear's impact on resistance spot welding quality, using experimental data from electrode displacement and force sensors. Neural network analysis achieved 90% accuracy in predicting weld quality based on electrode wear conditions.	[1]
Prediction of Nugget Diameter and Analysis of Process Parameters of RSW with Machine Learning Based on Feature Fusion	Bayesian algorithm; feature fusion; welding quality prediction	The research focuses on predicting welding quality in automotive body-in-white (BIW) production, emphasizing the importance of considering both welding process and material parameters for safety assurance. The study employs principal component analysis (PCA) for dimensionality reduction of material parameters and uses greedy feature selection to identify key characteristics. A comprehensive prediction model is developed that integrates both process parameters and material characteristics, improving upon conventional methods that only consider process parameters.	[38]
Implementation of Machine Learning Algorithms for Weld Quality Prediction and Optimization in Resistance Spot Welding	adaptive neuro-fuzzy inference system, artificial neural network, dual-phase steel, genetic algorithm, multi-objective optimization, resistance spot welding	The paper focuses on improving resistance spot welding (RSW) quality in automotive manufacturing, specifically examining 1.40-mm-thick DP780 steel sheets. The research investigates three critical welding parameters: welding current, welding time, and electrode force, evaluating their effects on nugget diameter, peel strength, tensile shear strength, and mean dynamic contact resistance.	[39]

A study on the machine learning method for estimating resistance spot welding button diameter using power curve and steel type information	Artificial neural network; Button diameter; Correlation analysis; Multiple linear regression analysis; Monitoring data; Resistance spot welding	The research addresses the high inspection costs in resistance spot welding quality evaluation in the automotive industry by developing a real-time prediction method. The study focuses on predicting weld button diameter using power data collected during welding and implementing a deep neural network model. The artificial neural network model demonstrated high accuracy with a coefficient of determination of 0.99 and a root mean square error of just 0.06 mm. The approach offers a cost-effective alternative to traditional inspection methods in automotive manufacturing. The results indicate that power monitoring combined with deep learning can effectively predict weld quality in real-time.	[40]
Applications of ultrasonic testing and machine learning methods to predict the static & fatigue behavior of spot-welded joints	RSW joint of multiple sheets; Ultrasonic test ;Image processing ;Static strength; Fatigue behavior; Artificial neural network; Genetic algorithm	This novel develops an integrated search system to assess the relationship between Ultrasonic Testing (UT) results and the strength of spot-welded joints in automotive applications. The study examines three-sheet resistance spot welds made from different low carbon steels, using pulse-echo data extracted through Image Processing Technique (IPT) combined with mechanical testing. An Artificial Neural Network (ANN) model, optimized by Genetic Algorithm (GA), was developed to predict static strength and fatigue life of three-sheet RSW joints based on UT results	[41]
Infrared (IR) quality assessment of robotized resistance spot welding based on machine learning	Infrared camera; Machine learning; Quality assessment; Resistance spot welding	The research introduces an automated quality assessment approach for resistance spot welding using machine learning analysis of infrared (IR) camera data integrated into a robotized welding system. The study evaluated different process parameters and quality criteria through two experimental approaches, finding that model prediction accuracy depends on the proximity of process parameter points. Analysis revealed that maximum IR intensity and temporal features of IR cooldown profiles provided the best class separation for quality prediction. The research demonstrates the advantages of IR monitoring for weld quality assessment, but notes that in rapid industrial welding scenarios, limited temporal IR data might affect prediction accuracy. The findings suggest that model selection is crucial when dealing with time-constrained industrial applications.	[42]

Table 2 - Research papers analyzed for Query 2.

As for the first query, targets are mainly the same (e.g., nugget, electrode wear) but some of the selected works use different data, for example ultrasonic [41] and infrared [42]. Other papers propose innovative ML techniques, such as transfer learning (TL) [37] and adaptive neuro-fuzzy inference [39].

Third query:

- Query: (“resistance spot welding” or “rsw”) AND “clustering”
- 11 documents

The results of the last query are fewer than the previous ones, primarily due to the fact that most researchers focus on the application of supervised machine learning. The two most relevant works have been summarized in Table 3.

Title	Keywords	Short description	Ref.
A parallel strategy for predicting the quality of welded joints in automotive bodies based on machine learning	Back propagation neural network (BPN); K-means clustering algorithm; Parallel strategies; Principal component analysis (PCA); Quality prediction; Resistance spot welding	The research addresses the challenge of real-time quality assessment in resistance spot welding by developing a parallel strategy using machine learning for different data subsets. The methodology employs PCA dimensionality reduction followed by k-means clustering to classify sub-datasets, with the elbow method determining optimal cluster numbers. Machine learning models are then applied in parallel to predict weld quality based on each sub-dataset's unique distribution characteristics. The approach outperforms traditional BP neural networks in predicting weld joint quality across all types, particularly with complex data distributions.	[43]
Functional clustering methods for resistance spot welding process data in the automotive industry	dynamic resistance curve; functional clustering; functional data analysis; Industry 4.0; resistance spot welding	The study explores the use of clustering methods for analyzing dynamic resistance curves (DRCs) in RSW, offering an alternative to costly offline testing in automotive manufacturing. The research demonstrates how functional clustering methods can effectively analyze DRCs without requiring specific feature extraction, providing a more efficient quality assessment approach. Data collected from Centro Ricerche Fiat's lab tests	[44]

		show that DRC clusters strongly correlate with electrode wear status, which impacts weld quality. This approach aligns with Industry 4.0's digital transformation, enabling online quality assessment through process parameter monitoring. The methodology offers a practical, cost-effective solution for real-time weld quality evaluation in large-scale production environments.	

Table 3 - Research papers analyzed for Query 3.

Figure 19 - Keywords cloud related to papers inserted in previous tables.

From Figure 19 it can be observed that there is a strong interest in the application of ML for the RSW process. Despite the keyword “machine learning” is one of the most frequent in the selected papers, works related to unsupervised applications in RSW are just a few, as it can be seen in Table 3 resulted from the third query.

4. Case study

This chapter describes the procedures followed, starting from the experimental campaign. After the data acquisition phase, feature engineering is performed before applying the clustering algorithms.

4.1 Experimental method

4.1.1 Welding machine, materials, and electrodes

The experimental welding operations were carried out using a medium-frequency direct current RSW machine (Fig. 20) in "current constant" mode and a TE700 (Tecna) control unit. This study focuses on two embedded sensors: (i) a piezoelectric surface strain sensor mod. 9232A (Kistler Italia) that records electrode force during welding, and (ii) a magnetostrictive linear position sensor mod. Temposonics R-series with a 2 μm resolution that measures electrode displacement. Both signals were registered at a 40 kHz sample rate using a National Instruments NI-9862 CAN interface module and handled with LabVIEW software.

Figure 20 - RSW machine in J-Tech laboratory.

The material used is GI50/50-U zinc-coated DP590 steel, which is commonly used in the automobile industry.

The welds were done using two Cu-Cr-Zr electrodes with a truncated cone shape and a nominal contact diameter of 4.5 mm, as specified by ISO 14373:2015 [45]. The top electrode moved to clamp the sheet stack, but the bottom electrode stayed fixed. A water flow rate of 4 l/min was used to cool both electrodes, as recommended by ISO 14373:2015 [45].

The thickness of the single sheets is 0.8. According to ISO 14273:2016 [46], the individual test piece dimensions are (45x105) mm and the weld overlap is 35 mm, as shown in Figure 21.

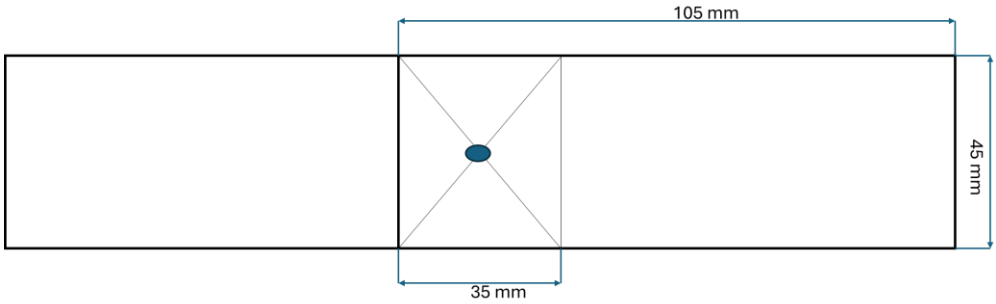

Figure 21 - Geometry for the shear tension tests.

4.1.2 Experimental setup and procedures

The partial weldability lobe at constant welding time was evaluated using ISO 14327:2004 [47] and ISO 18278-1:2022 [48] standards. The lobe's lower limit corresponds to a weld size of $3{,}5\sqrt{t}$ (t = material thickness), while the upper limit applies to welding conditions with expulsion. Welding time parameters were set in accordance with ISO 14373:2015 [45] and are shown in Table 4.

Up slope time (ms)	Current time (ms)	Down slope (ms)
25	150	25

Table 4 - Welding time parameters.

According to the ISO 17677-1:2021 [25] nomenclature, the weld time is calculated by adding upslope, downslope, and current. According to this, the weld time is 200

milliseconds. Focusing on the upper limit, 39 spot welds were performed utilizing 13 different combinations of welding current (I = 6.5 kA - 7 kA - 7.5 kA - 8 kA) and welding pressure (P = 0.8 bar – 0.9 - 1 bar - 1.1 bar). According to the standard ISO 14373:2015 [45], the initial points were made with 7 kA and 1 bar. To translate the pressure into force, the welding machine's technical parameters, which state that the highest applied pressure is 6.5 bar and equals to 12.42 kN, can be used. For example, using the percentage, one bar represents approximately 1.91 kN.
The presence of ejection was visually checked. Figure 22 shows a specimen, after a shear tension test, affected by expulsion that can be observed as the black area.

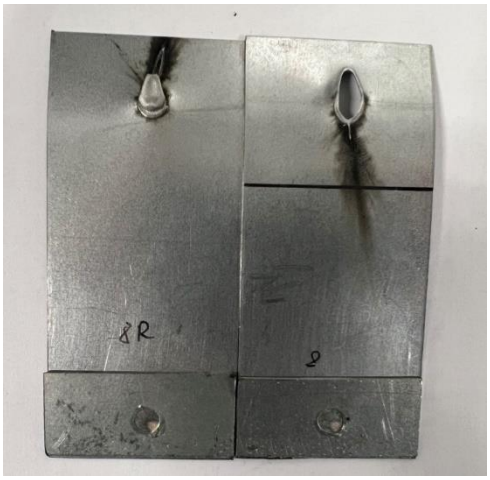

Figure 22- Specimen affected by expulsion.

The following table resumes the process parameters and the presence of expulsion for each specimen considered for this study.

ID	Current (kA)	Pressure (bar)	Expulsion
1	7	1	No
2			No
3			No
4	7,5	1	Yes
5			No
6			Yes
7	8	1	Yes
8			Yes
9			Yes
10	6,5	1	No
11			No
12			No
13	6,5	0,8	Yes
14			No
15			No
16	7	0,8	Yes
17			Yes
18			Yes
19	6,5	1,1	No
20			No
21			No
22	7	1,1	No
23			No
24			No
25	7,5	1,1	Yes
26			No
27			No
28	6,5	0,9	No
29			No
30			No
31	7	0,9	No
32			No
33			Yes
34	7,5	0,9	Yes
35			Yes
36			Yes
37	8	1,1	Yes
38			Yes
39			Yes

Table 5 - Process parameters.

4.2 Signals extraction

Several parameters can be evaluated during the RSW process to determine weld quality. Online signal monitoring is crucial for Industry 4.0, reducing the need for destructing tests and saving costs on quality testing samples.
Welding current or voltage are commonly utilized process indications due to their close correlation with the Joule effect.
Explanatory features (predictor variables) can be extracted from process signals to map electrode wear metrics (target variables) [49]. The RSW machine uses sensors to capture various signals. The most commonly utilized wear monitoring parameters include:

- electrode displacement
- force
- secondary current
- voltage
- resistance

The electrode displacement was monitored using a Temposonics R-series magnetostrictive linear position sensor with a precision of 2 μm and a frequency of 40 kHz. The signal was captured using the NI-9862 CAN interface module and controlled with LabVIEW software. A DAT format file was generated for each weld, containing one row every $2.5 * 10^{-5}$ second. The files provide information on the previously described signals, as well as details about the process and machine status.
With the aim of having clearer curves, the mean was calculated every 40 points. This means that there is one point every 0,001 s, as it was possible due to the sampling frequency (40 kHz). The values were stored in a vector for each welding.

4.2.1 Electrode Force

Electrode force is one of the most crucial parameters in Resistance Spot Welding (RSW), influencing weld formation and quality through a variety of mechanisms. The force, usually provided via pneumatic or hydraulic systems, must be sufficient to ensure intimate contact between the workpieces and sustain that contact throughout the welding cycle. Adequate electrode force aids in the removal of surface imperfections, the breakdown of surface impurities, and the containment of molten metal during nugget formation to prevent expulsion. Too little force can cause high contact resistance, leading to surface heating, expulsion, and electrode sticking, whereas too much force can induce superfluous indentation, limit heat generation due to low contact resistance, and produce undersized welds.

The ideal electrode force varies according to material type, thickness, surface condition, and welding current, ranging from a few hundred to several thousand Newtons. Electrode force also influences dynamic resistance during welding because it impacts the contact area and resistance at both the electrode-workpiece interface and the faying surface where the weld forms, which is critical for consistent weld quality.

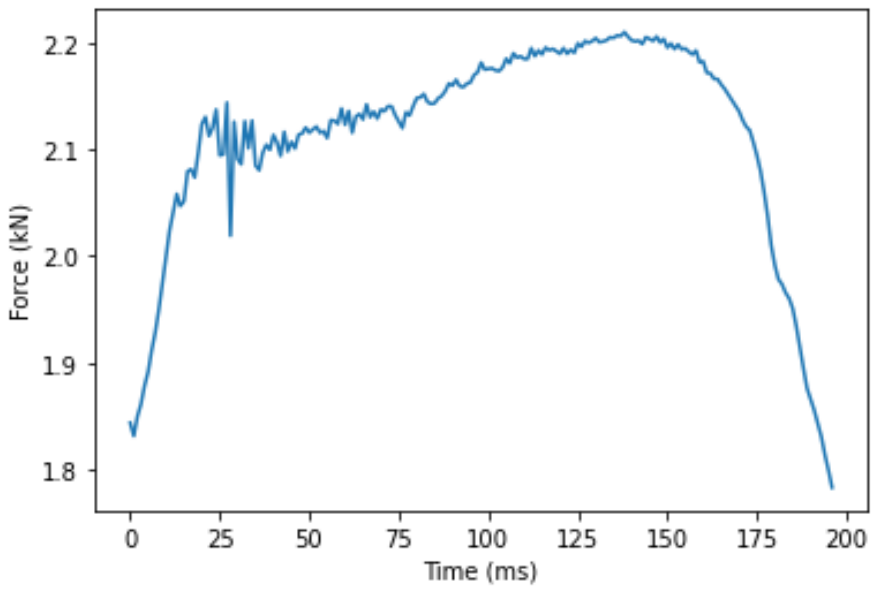

Figure 23 - Example of an electrode force acquired during the experimental campaign.

In Figure 23, an example illustrating the trend of the force recorded by the sensors used in the laboratory for the case study that will be presented in the next chapter. When welding current is applied, it rapidly increases and reaches a peak before stopping. The initial increase is due to thermal expansion of the weld nugget generated by the Joule action. As the weld zone softens, heat increases and electrode force decreases, resulting in lower resistance.

4.2.2 Electrode Displacement

Electrode displacement in Resistance Spot Welding (RSW) refers to the dynamic movement of electrodes throughout the welding process, which is an important indicator of weld formation and quality. The displacement curve typically exhibits four distinct phases as Zhang et al. [24] show in his work: initial compression as the electrodes clamp the sheets, thermal expansion as the materials heat up, sudden depression as the nugget forms and the material softens, and final solidification contraction. This movement pattern, which is frequently measured in micrometers, can be tracked in real time to provide useful information on welding progress and potential issues. For example, excessive displacement could imply ejection or material softening, whereas inadequate movement could indicate poor nugget formation. Figure 24 shows an example of electrode displacement recorded during the experimental campaign.

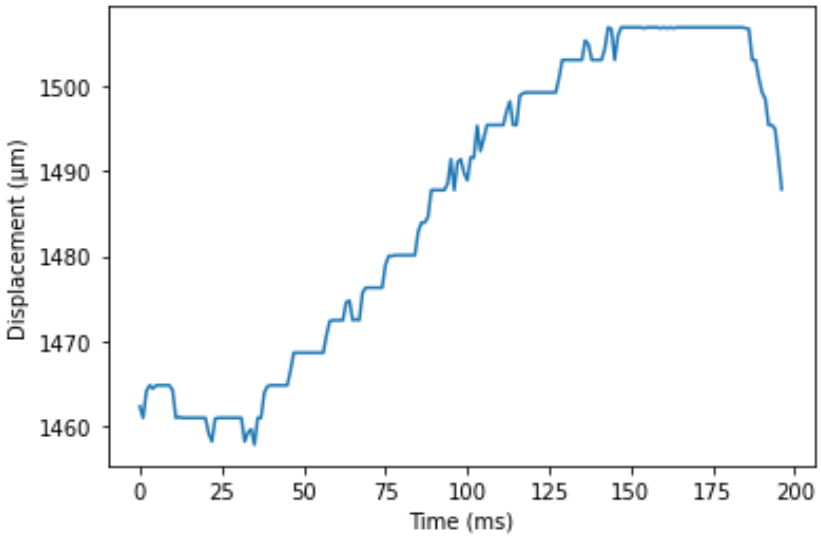

Figure 24 - Example of an electrode displacement acquired during the experimental campaign.

4.3 Feature engineering

A vector is created to collect data recorded from the electrode force sensor.

$$F^i = \left[f_1^i, \dots, f_j^i, \dots, f_N^i\right] \quad (kN)$$

The following features have been extracted for the Force signal:

- **f_max**: maximum force value.
- **f_max_time**: time in correspondence of the maximum force.

$$f^i_{\text{max\_time}} = \text{index}(f^i_{max}) \cdot 0.001 \ \ (\text{s})$$

- **f_avg_inc_befmax**: difference between the maximum force and the first value divided by the corresponding time interval.

$$f^i_{\text{avg\_inc\_befmax}} = \frac{f^i_{\text{max}} - f^i_1}{index(f^i_{max}) * 0.001} \quad (\frac{\text{kN}}{\text{s}})$$

- **f_avg_dec_aftermax**: difference between the maximum force and the last value over the time interval.

$$f^i_{\text{avg\_dec\_aftermax}} = \frac{f^i_{\text{max}} - f^i_N}{\left[index(f^i_N) - index(f^i_{max})\right] \cdot 0.001} \quad (\frac{\text{kN}}{\text{s}})$$

- **f_stds**: standard deviation of the force values.

$$f^i_{stds} = \sqrt{\frac{1}{N-1}\sum_{j=1}^{N}\left(f^i_j - f^i_{mean}\right)^2} \quad (kN)$$

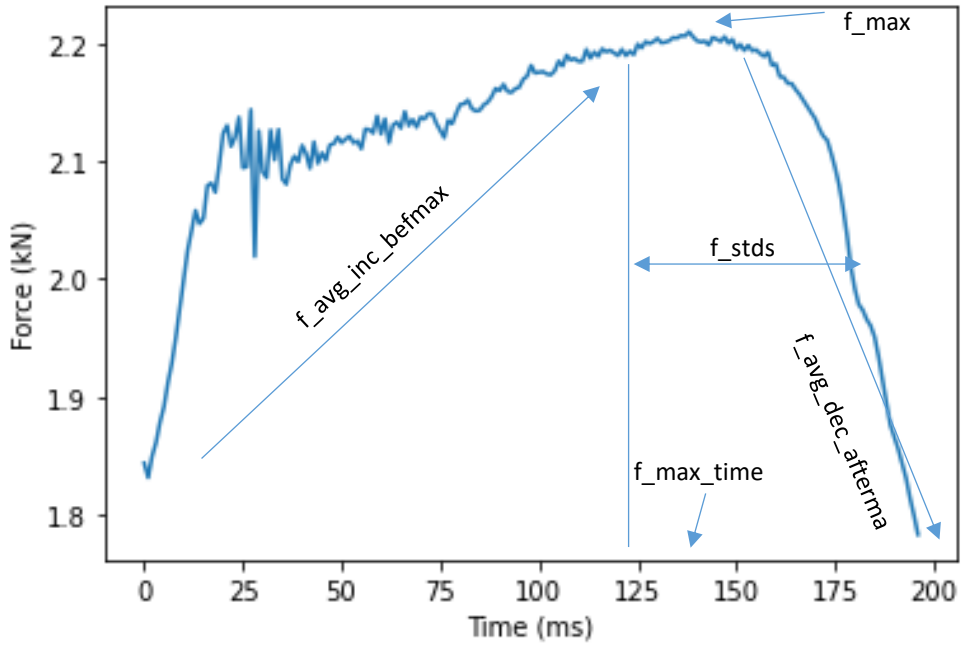

Figure 25 - Features extracted for the Force signal.

Accordingly, a vector is created to collect data recorded from the electrode displacement sensor and the following features have been computed for the Displacement signal:

$$D^i = \left[d_1^i, \dots, d_j^i, \dots, d_N^i\right] \quad (\mu m)$$

- **d_max**: maximum displacement value.

$$d_{max}^i = \max\left(D^i\right) \quad (\mu m)$$

- **d_max_time**: time in which the maximum displacement occurs.

$$d_{\max\_time}^i = \text{index}(d_{max}^i) \cdot 0.001 \quad (s)$$

- **d_avg_inc_befmax**: difference between the maximum displacement and the first value divided by the corresponding time interval.

$$d_{avg\_inc\_befmax}^i = \frac{d_{max}^i - d_1^i}{index\left(d_{max}^i\right) * 0.001} \quad \left(\frac{\mu m}{s}\right)$$

- **d_avg_dec_aftermax**: difference between the maximum displacement and the last value over the time interval.

$$d_{\text{slope\_aftermax}}^i = \frac{d_{\max\_last}^i}{\left[index\left(d_N^i\right) - index\left(d_{max}^i\right)\right] \cdot 0.001} \quad \left(\frac{\mu m}{s}\right)$$

- **d_stds**: standard deviation of the displacement values (ddof=1).

$$d_{stds}^i = \sqrt{\frac{1}{N-1} \sum_{j=1}^{N} \left(d_j^i - d_{mean}^i\right)^2} \quad (\mu m)$$

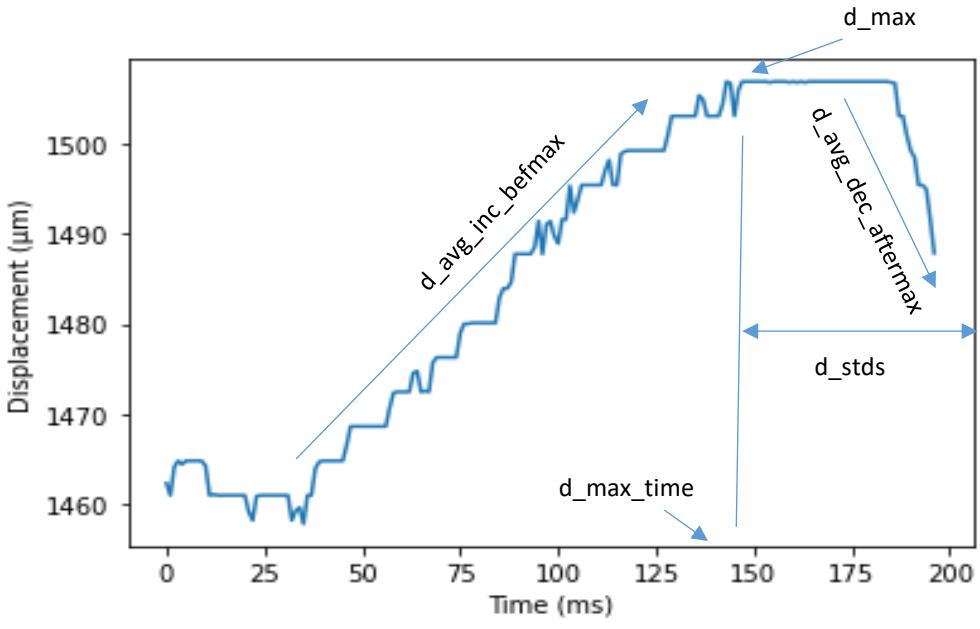

Figure 26 - Feature extracted for the Displacement signal.

5. Tools

In this chapter the programming language, the software program, and the libraries utilized will be discussed.

5.1 Python

Python is a high-level, multi-purpose programming language. Its design philosophy prioritizes code readability through the use of extensive indentation. Python has continuously ranked as one of the most popular programming languages, and it is widely used in the machine learning community.

5.2 Libraries

In Python, a library consists of modules, functions, and objects that offer predefined tools for specific tasks. Libraries improve software development by providing access to complicated functionalities without the need to design them from the start.

Pandas is a robust data manipulation library that uses DataFrames and Series to handle structured data, making it ideal for analyzing tabular data such as Excel sheets or CSV files.

NumPy focuses on numerical computing, providing efficient arrays and mathematical functions that provide the basis for many scientific computing activities.

Matplotlib is Python's major plotting toolkit, allowing you to generate a variety of charts, graphs, and visualizations with extensive customization possibilities.

The **Os** library is part of Python's standard library and provides functions for interacting with the operating system. It allows to work with files, directories, and system paths in a platform-independent way.

Scikit-learn is a comprehensive machine learning framework that provides tools for data preprocessing, model selection, and implementing various algorithms such as classification.

5.3 Anaconda and Jupyter Notebook

Anaconda Navigator is a graphical user interface (GUI) that allows to interact with packages and environments without having to input conda commands into a terminal window.

Jupyter Notebook is an open-source web application for creating and sharing documents containing live code, equations, visualizations, and narrative text.

It runs in every web browser and supports code in 40+ programming languages, though it's most commonly used with Python.

Jupyter Notebook comes included with Anaconda, making them a powerful combination for data science work.

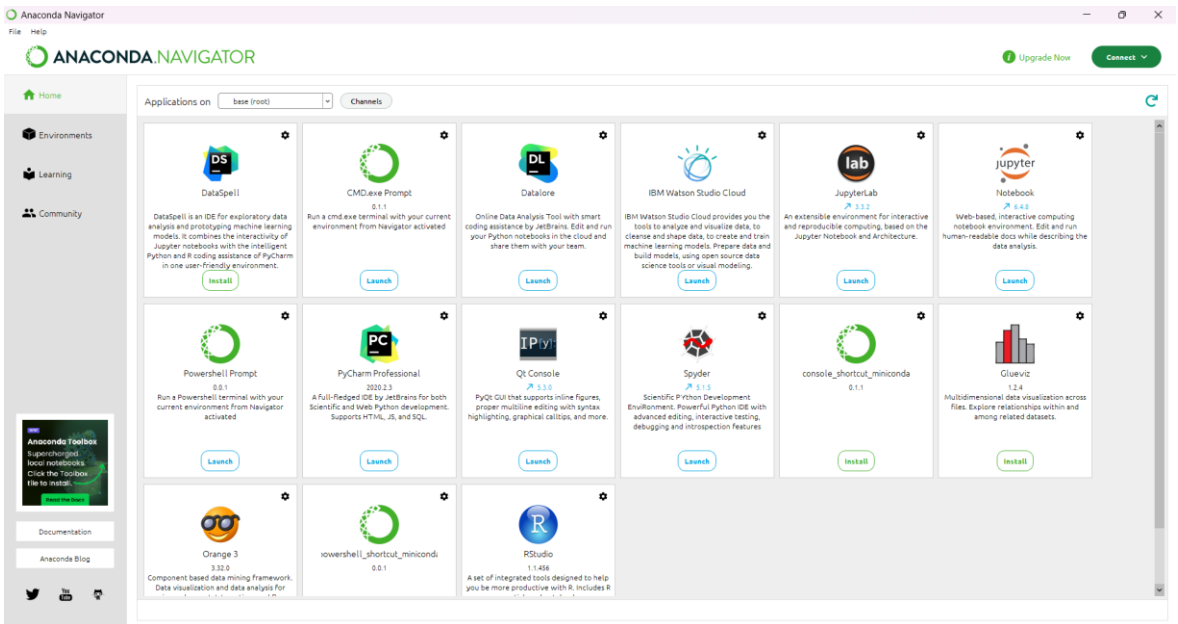

Figure 27 - Anaconda Navigator.

6. Clustering application

The features that were extracted from both force and displacement signals, better explained in Chapter 4.3, served as input variables for the clustering algorithm, with the aim to automatically distinguish between two distinct groups in the dataset. The first group corresponds to the samples without expulsions while the second one contains the welding points affected by the splash phenomenon.

6.1 Result without Feature Selection

Once the features for both Force and Displacement signals are extracted, the input data is used as the starting point for K-Means++, DBSCAN, and Hierarchical Clustering algorithms. First of all, the data is standardized, this is an important step because otherwise features with larger scales would dominate the model's learning. Standardization in fact helps in faster convergence during optimization, reduces the chance of getting stuck in local optima and makes training more stable.

6.1.1 K-Means++

The first algorithm to be implemented is K-Means++.
In order to determine the optimal number of clusters (K) for K-Means, the Elbow Method technique is implemented. The Elbow Method is a visual way for determining the optimal 'K' (number of clusters) in K-means clustering.
It works by computing the Within-Cluster Sum of Squares (WCSS), which is the sum of the squared distances between data points and the cluster center.
However, there is a point where raising K no longer results in a significant fall in WCSS, and the pace of decline decreases. This point is commonly referred to as the elbow.

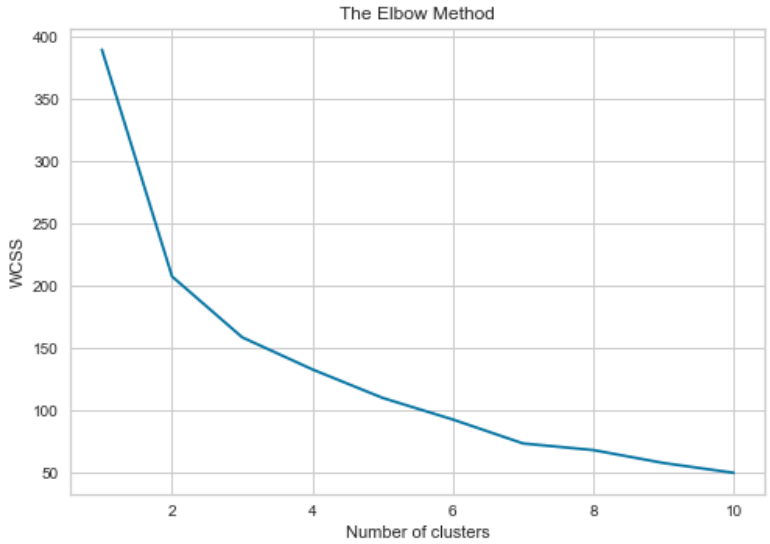

Figure 28 - The Elbow Method.

The code plots the Within-Cluster Sum of Squares (WCSS) against different values of K. WCSS measures how compact the clusters are (lower means better clustering). The "elbow" in the resulting plot suggests the optimal K, which in our case is set to 2. After determining the optimal K value, it is possible to proceed with the algorithm's implementation.

```
kmeans = KMeans(n_clusters = 2, init = "k-means++", random_state = 28)
y_kmeans = kmeans.fit_predict(X)
```

Figure 29 - K-Means++ application.

First, 2 clusters are created, “K-means++” is used as initialization for better centroids. Two operations are then combined:

- Fit: in order to find the cluster center.
- Predict: to assign each data point to nearest cluster.

The outcome is an array of cluster labels (0 or 1) for each data point.

```
y_kmeans
```

```
array([0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1])
```

Figure 30 – K-Means++ outcome.

6.1.2 Hierarchical Clustering

Hierarchical Clustering is applied as the second algorithm. First, the clustering is performed using the Ward method, as shown in the code below.

```python
def perform_hierarchical_clustering(df, n_clusters=2):
    """
    Perform hierarchical clustering and calculate silhouette scores
    """
    # Standardize the features
    scaler = StandardScaler()
    data_scaled = scaler.fit_transform(df)

    # Create linkage matrix for dendrogram
    linkage_matrix = linkage(data_scaled, method='ward')

    # Perform hierarchical clustering
    clustering = AgglomerativeClustering(n_clusters=n_clusters)
    clusters = clustering.fit_predict(data_scaled)

    # Calculate silhouette scores
    silhouette_avg = silhouette_score(data_scaled, clusters)
    silhouette_vals = silhouette_samples(data_scaled, clusters)

    # Add cluster assignments to original dataframe
    df_with_clusters = df.copy()
    df_with_clusters['Cluster'] = clusters
    df_with_clusters['Silhouette_Score'] = silhouette_vals

    return clusters, linkage_matrix, df_with_clusters, silhouette_avg, silhouette_vals
```

Figure 31 - Part of the code implemented for applying the Hierarchical Clustering in Python.

Then the dendrogram is plotted in order to better analyze the results.

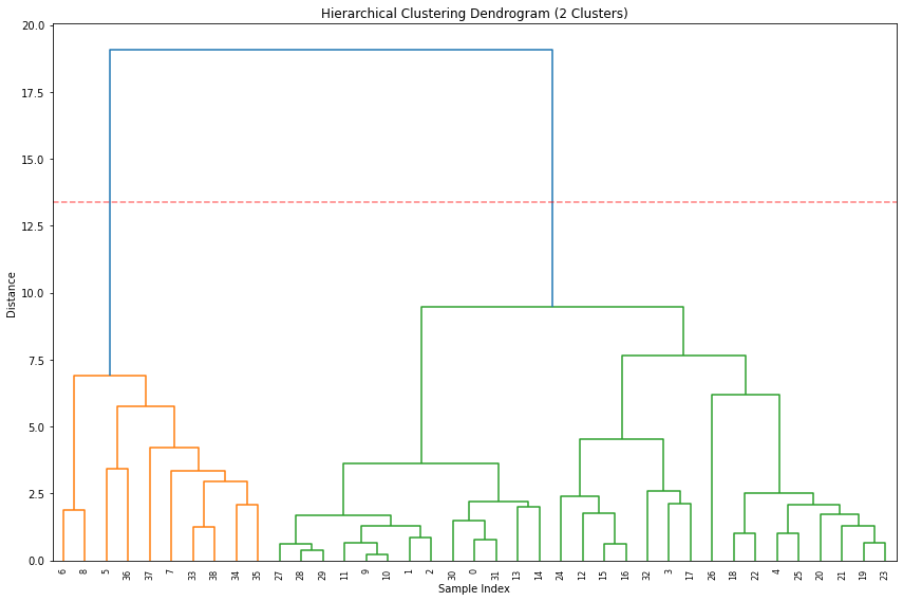

Figure 32 - Hierarchical Clustering Dendrogram.

The dendrogram clearly shows two main clusters (indicated by the orange and green branches). The red dashed horizontal line is the cut-off line that determines them. The height of the vertical lines represents the dissimilarity between clusters.

The left cluster (orange) is composed of fewer samples with lower height connections. Instead, the other cluster (green) contains more samples with connections at different heights.
The outcome, also in this case, is an array for clusters label for each data point.

```
clusters
```

```
array([0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1], dtype=int64)
```

Figure 33 - Hierarchical clustering results.

6.1.3 DBSCAN

The third algorithm implemented is DBSCAN.
At first, the k-distance graph is implemented, which is a technique used to determine the optimal Eps parameter for DBSCAN clustering.

```python
from sklearn.neighbors import NearestNeighbors

def plot_k_distance_graph(data, k):
    # Calculate distances to k-nearest neighbors for each point
    neigh = NearestNeighbors(n_neighbors=k)
    neigh.fit(data)
    distances, _ = neigh.kneighbors(data)

    # Sort distances in ascending order
    distances = np.sort(distances[:, k-1])

    # Plot k-distance graph
    plt.plot(range(len(distances)), distances)
    plt.xlabel('Points')
    plt.ylabel(f'Distance to {k}th nearest neighbor')
    plt.title('k-distance graph')
    plt.show()

plot_k_distance_graph(X, k=2)  # for 2D data
```

Figure 34 - K-distance graph code.

In particular, it helps visualize the distance distribution between points and it shows where density changes significantly in the dataset.

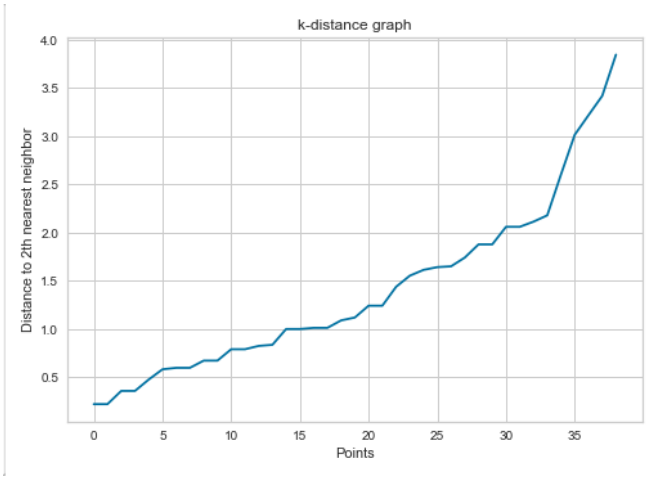

Figure 35 - K-distance graph.

The x-axis displays points arranged according to their distance from their kth nearest neighbor. The y-axis depicts these distances. The "elbow" point can be identified where the curve changes sharply. This elbow point is frequently an excellent option for the Eps parameter in DBSCAN.

```
from sklearn.cluster import DBSCAN

# Create and fit the DBSCAN model
dbscan = DBSCAN(eps=1.8, min_samples=5)
clusters = dbscan.fit_predict(X)

# Plot the results
plt.scatter(X[:, 3], X[:, 2], c=clusters, cmap='viridis')
plt.title('DBSCAN Clustering')
plt.xlabel('d_stds')
plt.ylabel('d_avg_dec_aftermax')
plt.colorbar(label='Cluster')
plt.show()
plt.savefig("figura_prova.jpg", bbox_inches='tight', dpi=300)

# Print number of clusters and noise points
n_clusters = len(set(clusters)) - (1 if -1 in clusters else 0)
n_noise = list(clusters).count(-1)
print(f'Number of clusters: {n_clusters}')
print(f'Number of noise points: {n_noise}')
```

Figure 36 - DBSCAN application.

The algorithm was then implemented with the following code.
Different values of eps and combinations of features were tested to reach the best configuration. In the picture below are represented the standard deviation and the average decrease after max of the displacement.

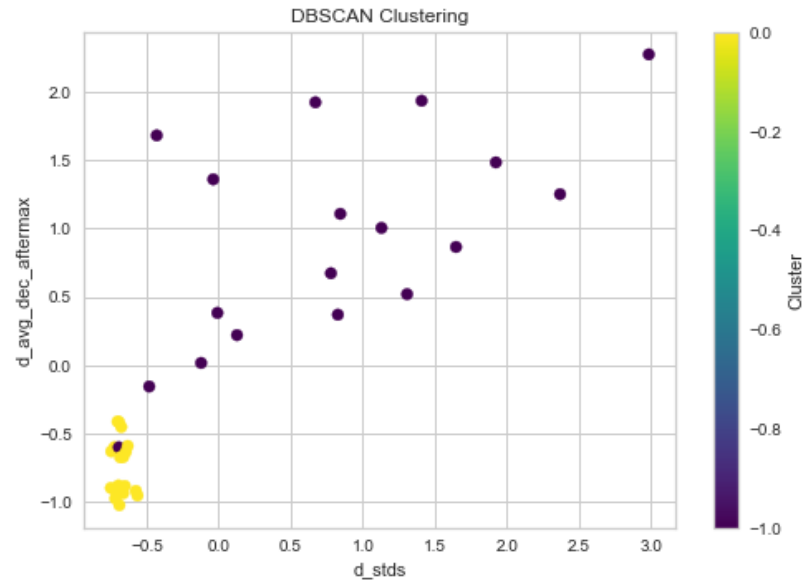

Figure 37 - DBSCAN Clustering in 2D.

From Figure 37 it can be noticed that 1 cluster was identified, containing the yellow points, which correspond to the "good points" that didn't register the expulsion phenomenon. All the others instead are categorized as noise.

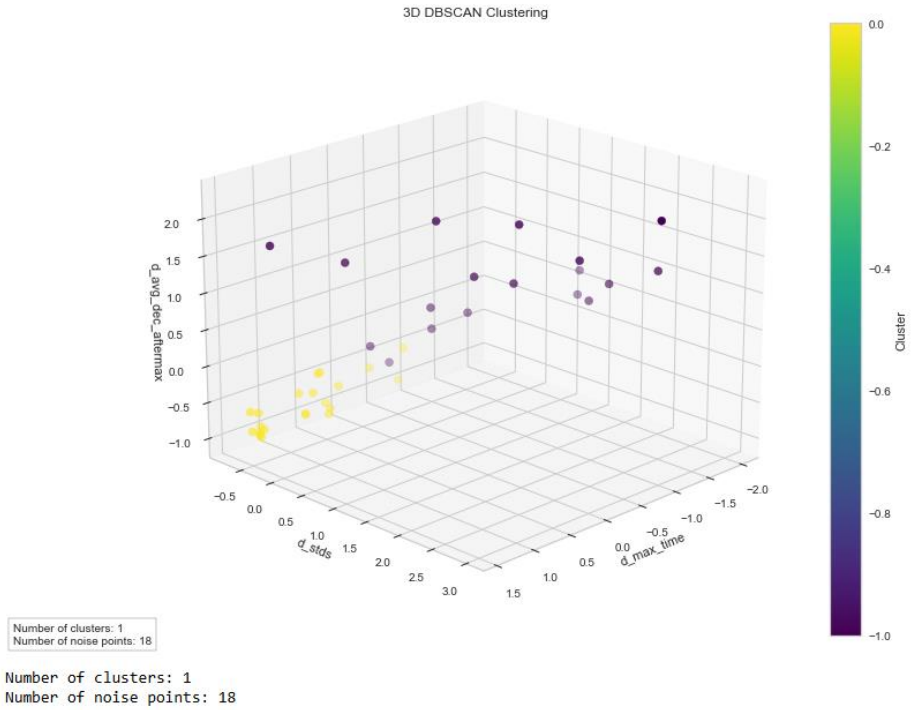

Figure 38 - DBSCAN Clustering in 3D.

Looking at Figure 37, it seems that there is a noise point in the middle of the yellow ones (i.e., good points, no expulsion). This happens due to the 2D representation but, if another dimension is added (Figure 38), it is clear that the misleading interpretation is not correct. Therefore, all the features have to be taken into account to explain the above consideration.

6.1.4 Performances

The Silhouette Analysis is performed to interpret and validate data consistency within clusters. The approach offers a concise graphical depiction of how effectively each object has been classified.
Since there is prior knowledge about how the data points should be grouped, some supervised performance metrics can be used for clustering evaluation. Specifically, the other metrics examined are Adjusted Rand Index (ARI) and Purity.

K-Means++

Silhouette:

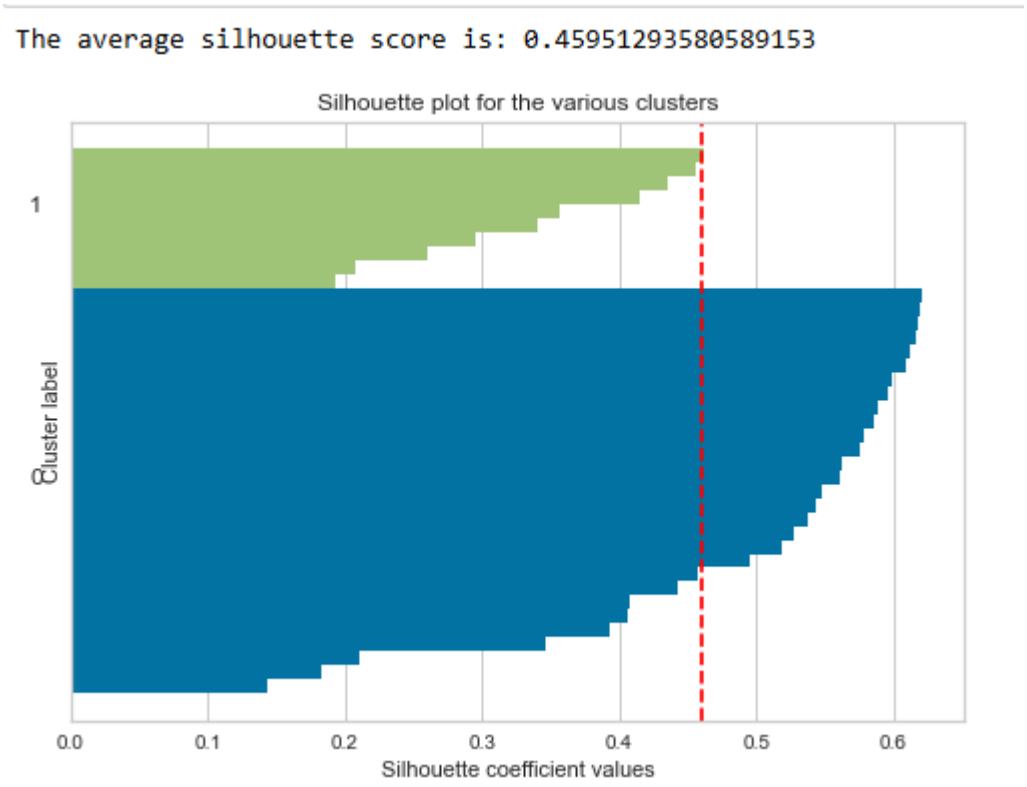

Figure 39 - Silhouette analysis for K-Means++ results.

The red dashed line indicates the overall average silhouette score which is 0.46. This result suggests reasonably good cluster separation, though there's room for improvement. Both clusters show positive silhouette coefficients (all values > 0). No negative values mean no points are likely misclassified.

Adjusted Rand Index (ARI)

The ARI result for K-Means++ is:

```
Adjusted Rand Index: 0.397
```

Figure 40 - ARI result for K-Means++.

This implies that K-Means++ is capturing some structure but missing significant patterns, and that it has moderate agreement with ground truth.

Purity

```
K-Means++ Purity: 0.821
```

Figure 41 - Purity result for K-Means++.

This implies that is establishing "pure" clusters, where the points primarily belong to the same true class.

Hierarchical Clustering

Silhouette

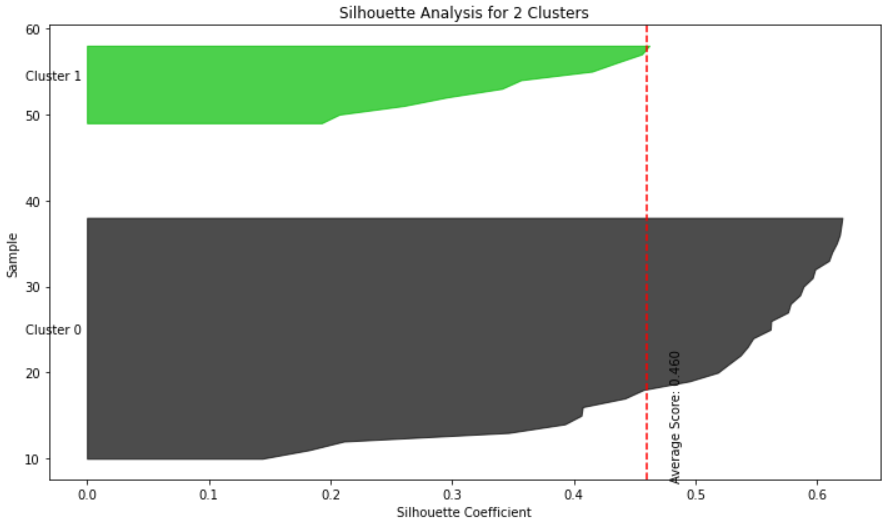

Figure 42 - Silhouette result for Hierarchical Clustering results.

Cluster Assignments and Silhouette Scores:
==========
Average Silhouette Score: 0.460

Mean Silhouette Score by Cluster:
==========
Cluster 0: 0.500
Cluster 1: 0.342

The result of the Silhouette Analysis is 0.46 is reasonable, but it can suggest there might be some overlap between clusters.

Adjusted Rand Index (ARI)

```
Adjusted Rand Index: 0.397
```

Figure 43 - ARI result for Hierarchical Clustering.

This implies inconsistency with the actual class structure. It is most likely dividing what should to be single clusters, perhaps forming more clusters than are required.

Purity

```
Hierarchical Purity: 0.821
```

Figure 44 - Purity result for Hierarchical Clustering.

Approximately 82% of points are assigned to the appropriate class members. Clusters are comparatively "clean" on their own. Little, uniform clusters are forming.

DBSCAN

Silhouette

```
Silhouette Score: 0.346
```

Figure 45 - Silhouette score for DBSCAN.

This result implies moderate cluster cohesiveness, some overlap between the clusters and the fact that some points may be quite near to surrounding clusters.

Adjusted Rand Index (ARI)

```
Adjusted Rand Index: 0.897
```

Figure 46 - ARI result for DBSCAN.

This shows very strong accord with ground truth. DBSCAN accurately captures actual structure.

Purity

```
DBSCAN Purity: 0.974
```

Figure 47 - Purity result for DBSCAN.

This suggests that it's creating pure clusters.

Analysis	***Silhouette***	***ARI***	***Purity***
K-Means++	*0.459*	*0.397*	*0.821*
Hierarchical	*0.460*	*0.397*	*0.821*
DBSCAN	*0.346*	*0.897*	*0.974*

Table 6 - Performance results.

Comparing the results of the performance for the algorithms it can been seen that for K-Means++ the high purity (0.821) but low ARI (0.397) suggests that it is creating "pure" clusters but it's likely over-segmenting the data.
The same can be observed for Hierarchical clustering, where clusters are internally "pure", but the overall structure doesn't match the true data partitioning well.
For DBSCAN instead ARI and Purity metrics are consistently high. This suggests robust clustering, even if the Silhouette score is moderate. This can be due to possible reasons, like non-spherical cluster shape or varying density, but it is not necessarily a bad result.

6.2 Results with Feature Selection

6.2.1 PCA

PCA analysis has been run through a Python script. Figure 48 shows the explained variance by principal components.

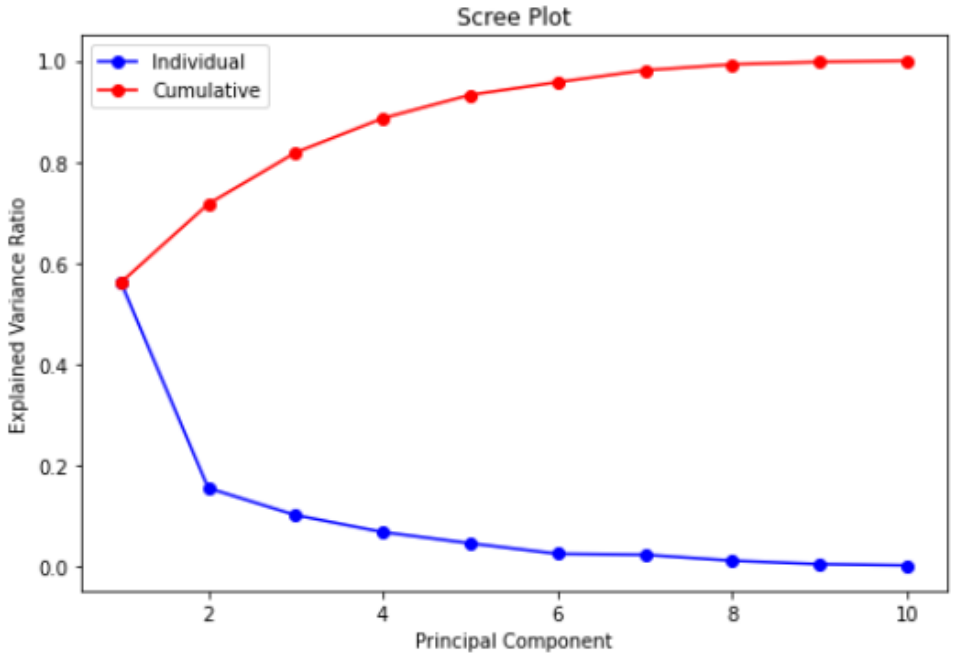

Figure 48 - PCA results.

Analyzing the feature loadings for the first 5 components (Figure 49), it can be observed that:

1. PC1:
   - Strongest positive loadings: d_stds (0.402), f_stds (0.386), f_avg_inc_befmax (0.346)
   - Strongest negative loadings: d_max_time (-0.347), d_max (-0.278)
   - This suggests PC1 primarily captures variation in standard deviations and increasing trends before maximums.
2. PC2:
   - Strongest negative loading: f_max_time (-0.772)
   - Notable negative loading: f_avg_dec_aftermax (-0.470)
   - PC2 seems to be heavily influenced by timing features, particularly for f-related measurements.

3. PC3:
    - Strongest positive: d_avg_inc_befmax (0.574)
    - Strongest negative: f_max (-0.560)
    - This component contrasts increasing trends before maximum in d-measurements with maximum values in f-measurements.
4. PC4:
    - Strongest positive: d_max (0.700)
    - Notable positive: f_max (0.461)
    - This component appears to capture maximum values across both d and f measurements.
5. PC5:
    - Strongest positive: d_avg_dec_aftermax (0.633)
    - Strongest negative: d_avg_inc_befmax (-0.552), f_avg_inc_befmax (-0.470)
    - This component contrasts decreasing trends after maximum with increasing trends before maximum.

Feature loadings:

	PC1	PC2	PC3	PC4	PC5
d_max_time	-0.347064	-0.270754	0.171313	0.292985	-0.002709
d_max	-0.278381	-0.204303	0.317637	0.700260	-0.073699
d_avg_dec_aftermax	0.342709	-0.069775	0.302989	0.125998	0.633340
d_stds	0.401915	-0.117004	0.079651	0.155630	0.030357
d_avg_inc_befmax	0.291882	-0.037745	0.573538	-0.153595	-0.551512
f_max_time	-0.012820	-0.772477	-0.069001	-0.206279	-0.135162
f_avg_dec_aftermax	0.294538	-0.469595	-0.202646	-0.106230	0.001945
f_stds	0.386471	0.013401	0.266050	0.076874	0.216901
f_max	0.286656	-0.062333	-0.559794	0.460647	-0.046989
f_avg_inc_befmax	0.345576	0.208961	-0.112636	0.296931	-0.469924

Figure 49 - PCA feature loadings

The first five components explain 93,25% of variance. Therefore, the most important features for each considered component are:

- PC1 (56.2%): d_stds, f_stds, f_avg_inc_befmax, d_max_time, d_max
- PC2 (15.5%): f_max_time, f_avg_dec_aftermax

- PC3 (10.2%): d_avg_inc_befmax, f_max
- PC4 (6.8%): d_max, f_max
- PC5 (4.6%): d_avg_dec_aftermax, d_avg_inc_befmax, f_avg_inc_befmax

In conclusion, from the above PCA analysis, the most important features are *d_stds, f_stds, f_avg_inc_befmax, d_max_time, d_max, f_max_time, f_avg_dec_aftermax, f_max, d_avg_dec_aftermax, d_avg_inc_befmax*, meaning that in order to explain more thant 90% of variance, all ten features are necessary.

6.2.2 Mutual information

Mutual information is determined between two variables and represents the decrease in uncertainty for one variable given a known value for the other.

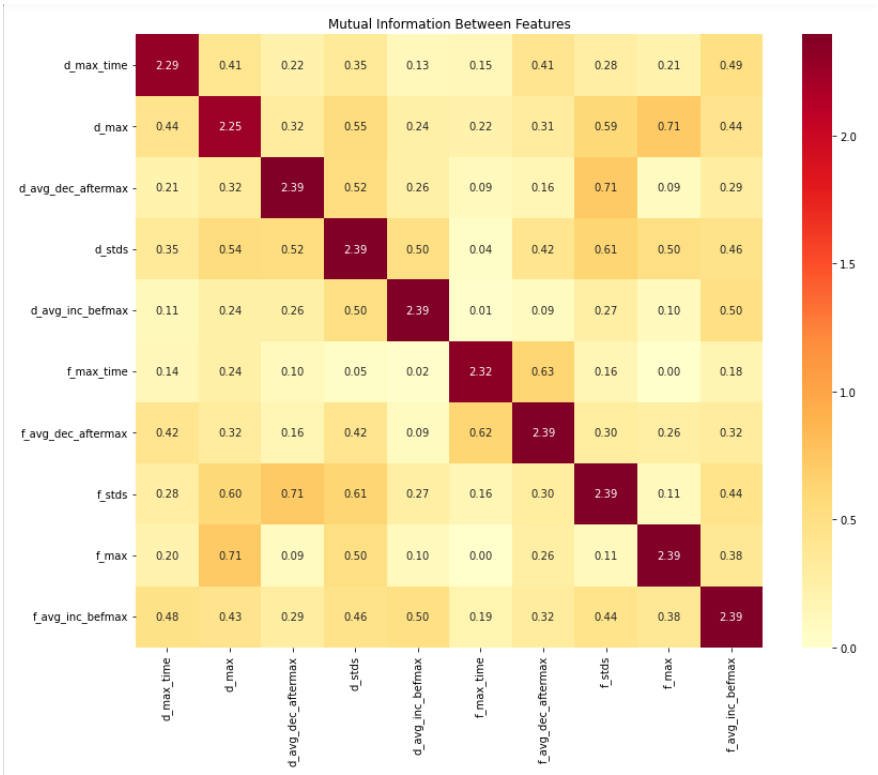

Figure 50 - Mutual information between features.

```
Highly related feature pairs (MI ≥ 0.7):

d_max - f_max: 0.715
d_avg_dec_aftermax - f_stds: 0.712

Average MI for each feature:
d_stds                0.634707
d_max                 0.607409
f_avg_inc_befmax      0.588784
f_stds                0.586125
f_avg_dec_aftermax    0.529157
d_avg_dec_aftermax    0.506594
d_max_time            0.492377
f_max                 0.476537
d_avg_inc_befmax      0.449533
f_max_time            0.380512
```

Figure 51 - Mutual information results.

Therefore, according to mutual information feature selection, it is possible to eliminate *d_max* and *f_stds* since they have a pair with high mutual information and, at the same time, their average MI is higher than the one of their pairs.
In conclusion, from the above MI analysis, the most important features are *d_stds, f_avg_inc_befmax, d_max_time, f_max_time, f_avg_dec_aftermax, f_max, d_avg_dec_aftermax, d_avg_inc_befmax.*

6.2.3 Performances

The performances have been analyzed after the FS and the results are the same as before features selection. This could imply that for the initial features this step did not provide any additional benefit. The algorithm was able to discover the underlying patterns and group the data points equally well with or without it.

6.3 Discussion

Table 7 summarizes all three clustering methods results, with and without the feature selection. It can be seen that the application of the feature selection, in this case, did not change the results. First of all, it has to be mentioned that the real class (i.e., expulsion/no expulsion) is known a priori from the visual inspection performed during the experimental campaign. Second, the final association between, for example, non-expulsion and class 0, is based on a majority criteria: if the majority of non-expulsion points are assigned to the class 0, this class will correspond to the non-expulsion class. The two classes identified by the above methods have been associated with the real class based on the previous considerations. Knowing that, the “accuracy” (i.e., percentage of correct real class assignment) of each algorithm can be measured.

The results reveal that DBSCAN exceeds K-Means++ and Hierarchical Clustering on this dataset, with an error rate of only 2.56%, compared to the 17.95% of the others. This shows that the data has a distinct density-based structure, which DBSCAN can effectively capture, whilst the other two algorithms struggle to appropriately group the data points. DBSCAN's exceptional performance can be due to its capacity to find clusters of any shape and size. K-Means++ and Hierarchical Clustering, on the other hand, make assumptions about the shape and size of clusters that may not be appropriate for this dataset. It is important to highlight that the ideal clustering algorithm is determined by the specific properties of the data, the clusters' underlying structure, and the desired clustering objectives.

Sample	Expulsion	K-Means++	K-Means++ with FS	Hierarchical	Hierarchical with FS	DBSCAN	DBSCAN with FS
1	No	0	0	0	0	0	0
2	No	0	0	0	0	0	0
3	No	0	0	0	0	0	0
4	Yes	0	0	0	0	-1	-1
5	No	0	0	0	0	0	0
6	Yes	1	1	1	1	-1	-1
7	Yes	1	1	1	1	-1	-1
8	Yes	1	1	1	1	-1	-1
9	Yes	1	1	1	1	-1	-1
10	No	0	0	0	0	0	0
11	No	0	0	0	0	0	0
12	No	0	0	0	0	0	0
13	Yes	0	0	0	0	-1	-1
14	No	0	0	0	0	0	0
15	No	0	0	0	0	0	0
16	Yes	0	0	0	0	-1	-1
17	Yes	0	0	0	0	-1	-1
18	Yes	0	0	0	0	-1	-1
19	No	0	0	0	0	0	0
20	No	0	0	0	0	0	0
21	No	0	0	0	0	0	0
22	No	0	0	0	0	0	0
23	No	0	0	0	0	0	0
24	No	0	0	0	0	0	0
25	Yes	0	0	0	0	-1	-1
26	No	0	0	0	0	0	0
27	No	0	0	0	0	-1	-1
28	No	0	0	0	0	0	0
29	No	0	0	0	0	0	0
30	No	0	0	0	0	0	0
31	No	0	0	0	0	0	0
32	No	0	0	0	0	0	0
33	Yes	0	0	0	0	-1	-1
34	Yes	1	1	1	1	-1	-1
35	Yes	1	1	1	1	-1	-1
36	Yes	1	1	1	1	-1	-1
37	Yes	1	1	1	1	-1	-1
38	Yes	1	1	1	1	-1	-1
39	Yes	1	1	1	1	-1	-1

Table 7 - Clustering results. In red are highlighted the wrong class assignments for each sample.

Moving forward to the practical insights of the results, in Figure 52, all 39 “Force vs Time” curves of the dataset have been plotted.

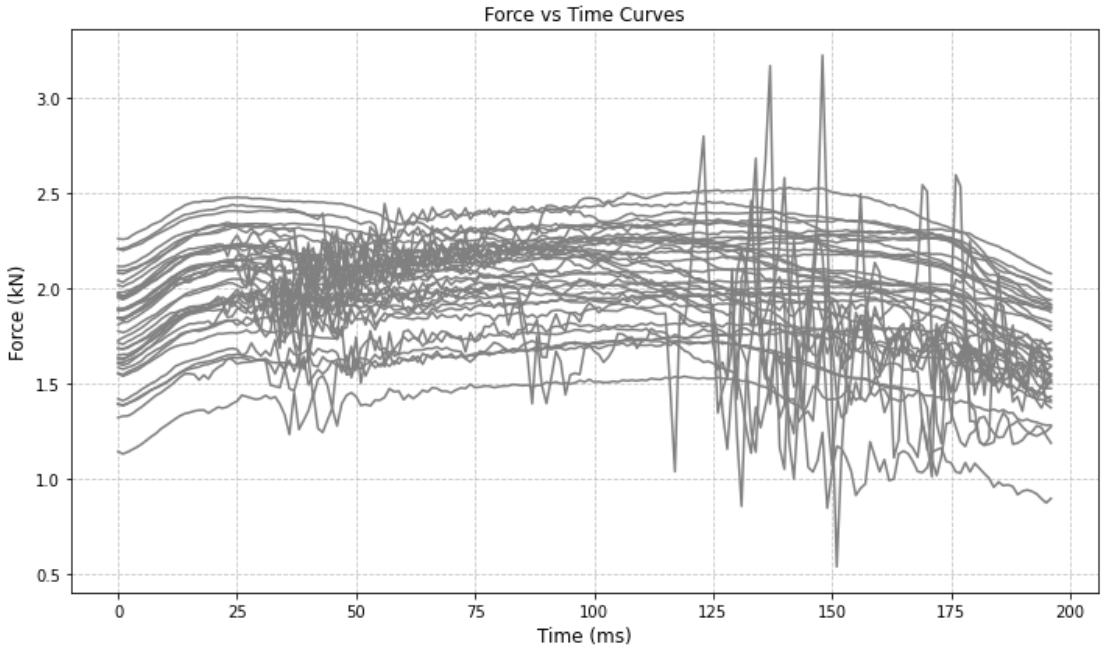

Figure 52 – Force vs Time Curves.

Looking at Figure 52 no immediate clear division is possible. A solution to extract meaningful information from the data and potentially uncover any hidden patterns is given by the methods shown above. Applying the best algorithm in terms of accuracy (i.e., DBSCAN), Figure 53 shows the same curves colored according to its classification.

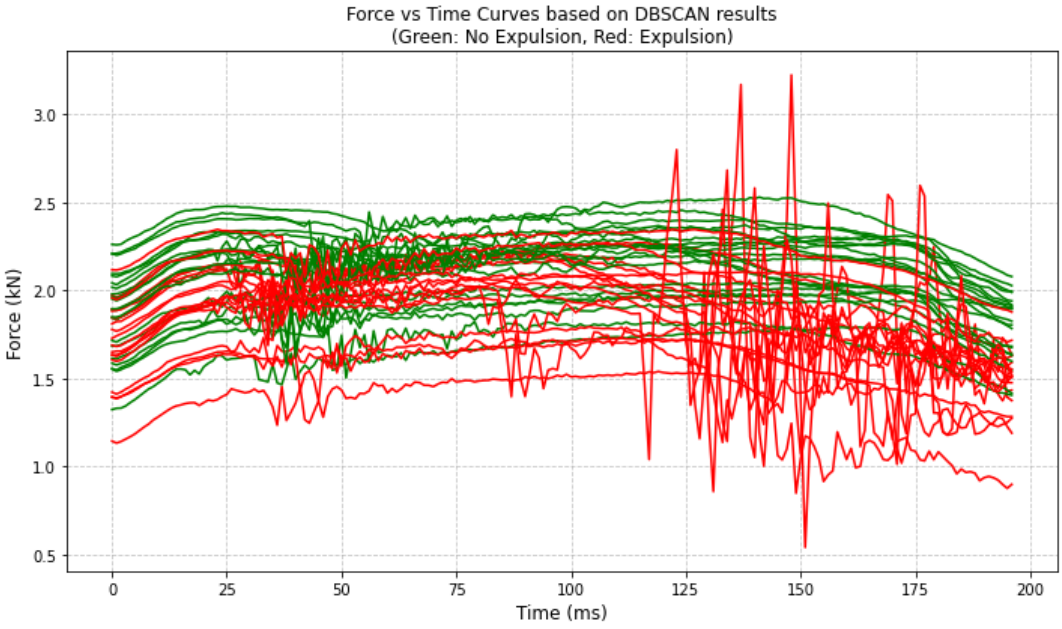

Figure 53 - Force vs Time Curves based on DBSCAN results.

Figure 53 illustrates the differences between the no-expulsion and expulsion events, and the DBSCAN classification appears to have effectively separated the two distinct force response patterns: the green curves represent cases without expulsion (class 0), and the red curves represent situations with expulsion (noise, “class” -1). The green curves (no expulsion) are characterized by gradual force increases and decreases. The peak forces are generally lower compared to the expulsion cases. The red curves (expulsion) instead show sharp spikes and rapid force fluctuations. Similar considerations can be made for displacement curves. The Plotting of the “Displacement vs Force Curves” can be seen in Figure 54.

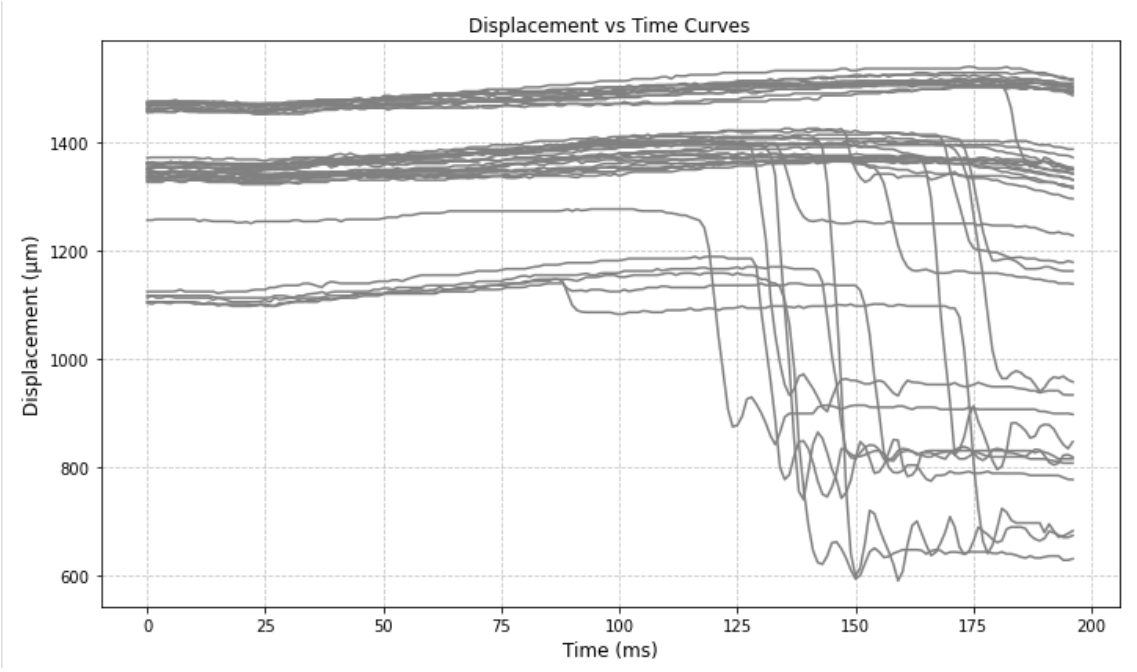

Figure 54 - Displacement vs Time curves.

Significant oscillations, sharp peaks, and quick shifts during the observed period can be noticed but the overall nature of the curve patterns makes the identification of specific patterns challenging.

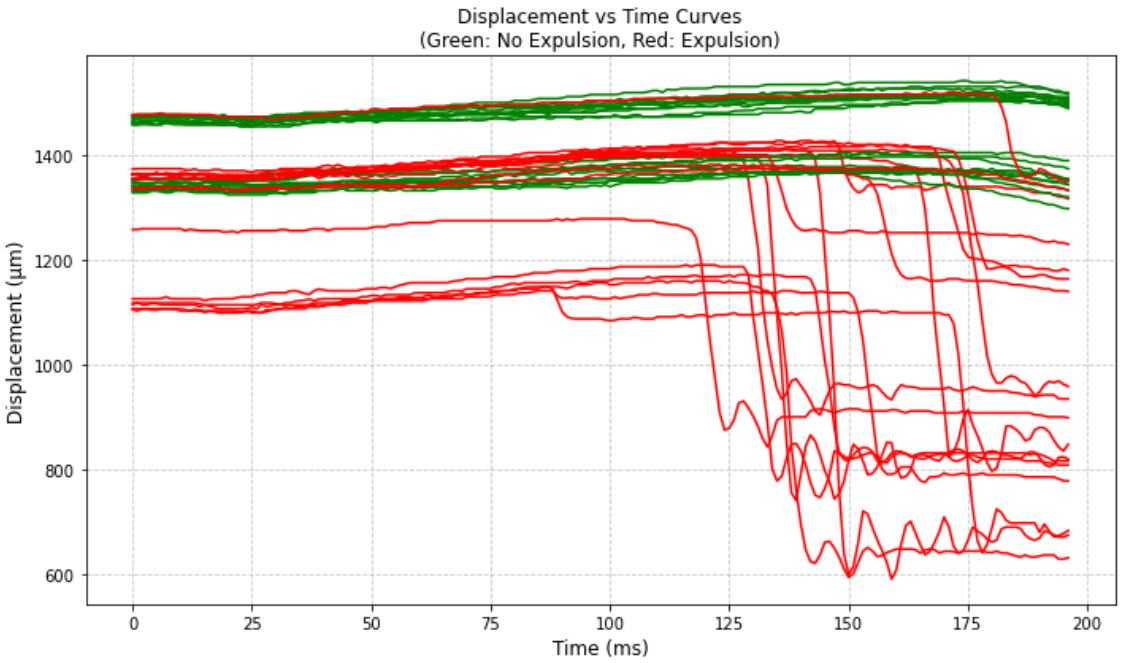

Figure 55 - Displacement vs Time Curves.

Figure 55, which incorporates the DBSCAN classification results, clearly separates these curves into two distinct curves. Like for the Force graph, the green curves represent cases with no expulsion, while the red curves indicate cases with expulsion. It is worth noting that the expulsion displacement curves show a rapid drop in the second part of the welding time (i.e., 100-200 ms), corresponding to an electrode instant and negative shift when expulsion happens.

Conclusions

This thesis proposes a machine learning method leading to several advantages for the RSW process. The work introduces an alternative use of unsupervised techniques since, as better explained in the Discussion section (6.3), the real classes (i.e., expulsion/no expulsion) are known a priori from the experimental campaign. Regarding the applied methods and their accuracy without feature selection, K-Means++ and Hierarchical methods have a percentage error of almost 18% but DBSCAN reduces the error to 3%. In general, K-means++ and Hierarchical clustering have shown several drawbacks, both being based on distance metrics. DBSCAN major characteristics provide considerable advantages for the dataset. As a density-based clustering technique, it effectively identifies the non-expulsion points and considers all the others as noise.

The three algorithms have been used after feature selection that has been performed using the PCA and mutual information techniques. In total, eight out of ten features have been identified but the results have not changed.

As a first result, since it is one of the most expensive phases, both in terms of time and costs, the method can be used for data labeling, useful for further supervised machine learning applications (e.g., classification). Another important result is that the proposed procedure can be employed for extracting new features from displacement and force curves since it highlights the unique patterns of each class. Thirdly, this application can enable process monitoring by automatically spotting out the non-conforming welding points. Fourthly, clustering can be employed for post process quality control, avoiding costly destructive testing methods.

Future implementation could focus on the creation of new features in order to reach 100% accuracy. Furthermore, new experiments could be implemented to test the proposed methodology on a larger scale. Lastly, since in this work expulsion was induced by varying process parameters, another future work could focus on replicating the approach in presence of other defects (e.g., initial gap and edge welding) that commonly cause expulsion in production lines.

Bibliography

[1] L. Panza, G. Bruno, G. Antal, M. De Maddis, e P. Russo Spena, «Machine learning tool for the prediction of electrode wear effect on the quality of resistance spot welds», *Int. J. Interact. Des. Manuf. IJIDeM*, vol. 18, fasc. 7, pp. 4629–4646, set. 2024, doi: 10.1007/s12008-023-01733-7.

[2] N. Gupta e R. Mangla, *Artificial intelligence basics: a self-teaching introduction*. Dulles, Va: Mercury Learning & Information, 2020.

[3] T. M. Mitchell, *Machine Learning*. in McGraw-Hill series in computer science. New York: McGraw-Hill, 1997.

[4] O. Campesato, *Artificial Intelligence, Machine Learning, and Deep Learning*. Dulles, VA: Mercury Learning and Information, 2020. doi: 10.1515/9781683924654.

[5] «Wikipedia - Clustering». [Online]. Disponibile su: https://en.wikipedia.org/wiki/

[6] Ayantika Sarkar e Pradnya Kashikar, «Literature review of implementation of machine learning algorithms for improving the network security». [Online]. Disponibile su: https://www.researchgate.net/publication/378941021_Literature_review_of_implementation_of_machine_learning_algorithms_for_improving_the_network_security

[7] «Wikipedia - K-Means Clustering». [Online]. Disponibile su: https://en.wikipedia.org/wiki/K-means_clustering

[8] «IBM - K-Means Clustering». [Online]. Disponibile su: https://www.ibm.com/topics/k-means-clustering

[9] «Wikipedia - DBSCAN». [Online]. Disponibile su: https://en.wikipedia.org/wiki/DBSCAN

[10] «DataCamp - DBSCAN». [Online]. Disponibile su: https://www.datacamp.com/tutorial/dbscan-clustering-algorithm?dc_referrer=https%3A%2F%2Fwww.google.com%2F

[11] «Wikipedia - Hierarchical Clustering». [Online]. Disponibile su: https://en.wikipedia.org/wiki/Hierarchical_clustering

[12] «DataCamp - Hierarchical Clustering». [Online]. Disponibile su: https://www.datacamp.com/tutorial/introduction-hierarchical-clustering-python

[13] J. Huang, Y.-F. Li, e M. Xie, «An empirical analysis of data preprocessing for machine learning-based software cost estimation», *Inf. Softw. Technol.*, vol. 67, pp. 108–127, nov. 2015, doi: 10.1016/j.infsof.2015.07.004.

[14] «Wikipedia - Feature engineering». [Online]. Disponibile su: https://en.wikipedia.org/wiki/Feature_engineering

[15] S. Solorio-Fernández, J. A. Carrasco-Ochoa, e J. Fco. Martínez-Trinidad, «A review of unsupervised feature selection methods», *Artif. Intell. Rev.*, vol. 53, fasc. 2, pp. 907–948, feb. 2020, doi: 10.1007/s10462-019-09682-y.

[16] S. Solorio-Fernández, J. Ariel Carrasco-Ochoa, e J. Fco. Martínez-Trinidad, «A systematic evaluation of filter Unsupervised Feature Selection methods», *Expert Syst. Appl.*, vol. 162, p. 113745, dic. 2020, doi: 10.1016/j.eswa.2020.113745.

[17] Isabelle Guyon e Andre Elisseeff, «An Introduction to Variable and Feature Selection». [Online]. Disponibile su: https://www.jmlr.org/papers/volume3/guyon03a/guyon03a.pdf

[18] S. Niijima e Y. Okuno, «Laplacian Linear Discriminant Analysis Approach to Unsupervised Feature Selection», *IEEE/ACM Trans. Comput. Biol. Bioinform.*, vol. 6, fasc. 4, pp. 605–614, ott. 2009, doi: 10.1109/TCBB.2007.70257.

[19] D. Devakumari e K. Thangavel, «Unsupervised Modified Adaptive Floating Search Feature Selection», in *Advances in Computing and Communications*, vol. 191, A. Abraham, J. Lloret Mauri, J. F. Buford, J. Suzuki, e S. M. Thampi, A c. di, in

Communications in Computer and Information Science, vol. 191. , Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 358–365. doi: 10.1007/978-3-642-22714-1_37.
[20] S. Alelyani, J. Tang, e H. Liu, «Feature Selection for Clustering: A Review», in *Data Clustering*, 1ª ed., C. C. Aggarwal e C. K. Reddy, A c. di, Chapman and Hall/CRC, 2018, pp. 29–60. doi: 10.1201/9781315373515-2.
[21] H. Liu, *Feature Engineering for Machine Learning and Data Analytics*, 1ª ed. CRC Press, 2018. doi: 10.1201/9781315181080.
[22] «towardsdatascience - Performance metrics in ML». [Online]. Disponibile su: https://towardsdatascience.com/performance-metrics-in-machine-learning-part-3-clustering-d69550662dc6
[23] L. Panza, M. D. Maddis, e P. R. Spena, «Use of electrode displacement signals for electrode degradation assessment in resistance spot welding», *J. Manuf. Process.*, vol. 76, pp. 93–105, apr. 2022, doi: 10.1016/j.jmapro.2022.01.060.
[24] H. Zhang e J. Senkara, *Resistance welding: fundamentals and applications*. Boca Raton, Fla.: CRC/Taylor & Francis, 2006.
[25] «BS EN ISO 17677-1-2021 - Resistance welding. Vocabulary. Part 1, Spot, projection and seam welding».
[26] Y.-J. Xia, Z.-W. Su, Y.-B. Li, L. Zhou, e Y. Shen, «Online quantitative evaluation of expulsion in resistance spot welding», *J. Manuf. Process.*, vol. 46, pp. 34–43, ott. 2019, doi: 10.1016/j.jmapro.2019.08.004.
[27] H. Zhang, S. J. Hu, J. Senkara, e S. Cheng, «A Statistical Analysis of Expulsion Limits in Resistance Spot Welding», *J. Manuf. Sci. Eng.*, vol. 122, fasc. 3, pp. 501–510, ago. 2000, doi: 10.1115/1.1285873.
[28] Senkara J., Zhang H., e Hu S.J., «Expulsion prediction in resistance spot welding».
[29] D. Zhao *et al.*, «Dynamic resistance signal–based wear monitoring of resistance spot welding electrodes», *Int. J. Adv. Manuf. Technol.*, vol. 133, fasc. 7–8, pp. 3267–3281, ago. 2024, doi: 10.1007/s00170-024-13993-y.
[30] M. Ullrich, M. Wohner, e S. Jüttner, «Quality monitoring for a resistance spot weld process of galvanized dual-phase steel based on the electrode displacement», *Weld. World*, vol. 68, fasc. 7, pp. 1791–1800, lug. 2024, doi: 10.1007/s40194-024-01720-w.
[31] M. Russell *et al.*, «Comparison and explanation of data-driven modeling for weld quality prediction in resistance spot welding», *J. Intell. Manuf.*, vol. 35, fasc. 3, pp. 1305–1319, mar. 2024, doi: 10.1007/s10845-023-02108-1.
[32] Graduate School of Excellence Advanced Manufacturing Engineering, University of Stuttgart, Stuttgart, Germany e S. Durnagöz, «An Approach to Inline Monitoring of the Electrode State in Resistance Spot Welding», *Int. J. Electr. Electron. Eng. Telecommun.*, vol. 13, fasc. 3, pp. 245–251, 2024, doi: 10.18178/ijeetc.13.3.245-251.
[33] W. Yang, P. P Gao, e X. Gao, «Online evaluation of resistance spot welding quality and defect classification», *Meas. Sci. Technol.*, vol. 34, fasc. 9, p. 095016, set. 2023, doi: 10.1088/1361-6501/acce58.
[34] L. Bogaerts, A. Dejans, M. G. R. Faes, e D. Moens, «A machine learning approach for efficient and robust resistance spot welding monitoring», *Weld. World*, vol. 67, fasc. 8, pp. 1923–1935, ago. 2023, doi: 10.1007/s40194-023-01519-1.
[35] J. Dong, J. Hu, e Z. Luo, «Quality Monitoring of Resistance Spot Welding Based on a Digital Twin», *Metals*, vol. 13, fasc. 4, p. 697, apr. 2023, doi: 10.3390/met13040697.
[36] B. Zhou, T. Pychynski, M. Reischl, E. Kharlamov, e R. Mikut, «Machine learning with domain knowledge for predictive quality monitoring in resistance spot welding», *J. Intell. Manuf.*, vol. 33, fasc. 4, pp. 1139–1163, apr. 2022, doi: 10.1007/s10845-021-01892-y.

[37] Z.-J. Yue *et al.*, «Improving RSW nugget diameter prediction method: unleashing the power of multi-fidelity neural networks and transfer learning», *Adv. Manuf.*, vol. 12, fasc. 3, pp. 409–427, set. 2024, doi: 10.1007/s40436-024-00503-2.
[38] Q. Zhu, H. Shen, X. Zhu, e Y. Wang, «Prediction of Nugget Diameter and Analysis of Process Parameters of RSW with Machine Learning Based on Feature Fusion», *Electronics*, vol. 13, fasc. 13, p. 2484, giu. 2024, doi: 10.3390/electronics13132484.
[39] N. N. Johnson *et al.*, «Implementation of Machine Learning Algorithms for Weld Quality Prediction and Optimization in Resistance Spot Welding», *J. Mater. Eng. Perform.*, vol. 33, fasc. 13, pp. 6561–6585, lug. 2024, doi: 10.1007/s11665-023-08503-2.
[40] J. Kim, Y. Park, e N. Ku, «A study on the machine learning method for estimating resistance spot welding button diameter using power curve and steel type information», *J. Mech. Sci. Technol.*, vol. 37, fasc. 7, pp. 3711–3719, lug. 2023, doi: 10.1007/s12206-023-0636-x.
[41] N. Amiri, G. H. Farrahi, K. R. Kashyzadeh, e M. Chizari, «Applications of ultrasonic testing and machine learning methods to predict the static & fatigue behavior of spot-welded joints», *J. Manuf. Process.*, vol. 52, pp. 26–34, apr. 2020, doi: 10.1016/j.jmapro.2020.01.047.
[42] P. Stavropoulos, K. Sabatakakis, A. Papacharalampopoulos, e D. Mourtzis, «Infrared (IR) quality assessment of robotized resistance spot welding based on machine learning», *Int. J. Adv. Manuf. Technol.*, vol. 119, fasc. 3–4, pp. 1785–1806, mar. 2022, doi: 10.1007/s00170-021-08320-8.
[43] G. Chen, B. Sheng, R. Luo, e P. Jia, «A parallel strategy for predicting the quality of welded joints in automotive bodies based on machine learning», *J. Manuf. Syst.*, vol. 62, pp. 636–649, gen. 2022, doi: 10.1016/j.jmsy.2022.01.011.
[44] C. Capezza, F. Centofanti, A. Lepore, e B. Palumbo, «Functional clustering methods for resistance spot welding process data in the automotive industry», *Appl. Stoch. Models Bus. Ind.*, vol. 37, fasc. 5, pp. 908–925, set. 2021, doi: 10.1002/asmb.2648.
[45] «BS EN ISO 14373-2015-Resistance welding. Procedure for spot welding of uncoated and coated low carbon steels».
[46] «BS EN ISO 14273-2016-Resistance welding. Destructive testing of welds. Specimen dimensions and procedure for tensile shear testing resistance spot and embossed projection welds».
[47] «BS EN ISO 14327-2004-Resistance welding. Procedures for determining the weldability lobe for resistance spot, projection and seam welding».
[48] «BS EN ISO 18278-1-2022-Resistance welding. Weldability. Part 1, General requirements for the evaluation of weldability for resistance spot, seam and projection welding of metallic materials».
[49] L. Panza, G. Bruno, M. De Maddis, F. Lombardi, P. R. Spena, e E. Traini, «Data-Driven Framework for Electrode Wear Prediction in Resistance Spot Welding», in *Product Lifecycle Management. Green and Blue Technologies to Support Smart and Sustainable Organizations*, vol. 639, O. Canciglieri Junior, F. Noël, L. Rivest, e A. Bouras, A c. di, in IFIP Advances in Information and Communication Technology, vol. 639. , Cham: Springer International Publishing, 2022, pp. 239–252. doi: 10.1007/978-3-030-94335-6_17.