electronics

Article

Fault Prediction in Resistance Spot Welding: A Comparison of Machine Learning Approaches

Gabriele Ciravegna *, Franco Galante *, Danilo Giordano, Tania Cerquitelli and Marco Mellia

Politecnico di Torino, Department of Control and Computer Engineering, Corso Duca degli Abruzzi, 24, 10129 Torino, Italy; danilo.giordano@polito.it (D.G.); tania.cerquitelli@polito.it (T.C.); marco.mellia@polito.it (M.M.)

* Correspondence: gabriele.ciravegna@polito.it (G.C.); franco.galante@polito.it (F.G.)

Abstract: Resistance spot welding is widely adopted in manufacturing and is characterized by high reliability and simple automation in the production line. The detection of defective welds is a difficult task that requires either destructive or expensive and slow non-destructive testing (e.g., ultrasound). The robots performing the welding automatically collect contextual and process-specific data. In this paper, we test whether these data can be used to predict defective welds. To do so, we use a dataset collected in a real industrial plant that describes welding-related data labeled with ultrasonic quality checks. We use these data to develop several pipelines based on shallow and deep learning machine learning algorithms and test the performance of these pipelines in predicting defective welds. Our results show that, despite the development of different pipelines and complex models, the machine-learning-based defect detection algorithms achieve limited performance. Using a qualitative analysis of model predictions, we show that correct predictions are often a consequence of inherent biases and intrinsic limitations in the data. We therefore conclude that the automatically collected data have limitations that hamper fault detection in a running production plant.

Keywords: resistance spot welding; machine learning; fault prediction

check for updates

Citation: Ciravegna, G.; Galante, F.; Giordano, D.; Cerquitelli, T.; Mellia, M. Fault Prediction in Resistance Spot Welding: A Comparison of Machine Learning Approaches. *Electronics* **2024**, *13*, 3693. https://doi.org/10.3390/electronics13183693

Academic Editor: Chunping Li

Received: 6 August 2024
Revised: 5 September 2024
Accepted: 14 September 2024
Published: 18 September 2024

1. Introduction

The advent of Industry 4.0 has opened up unprecedented opportunities facilitating communication and collaboration between machines and processes and enabling intelligent decision making through the use of *smart* industrial technologies. The increasing availability of large amounts of data collected in industrial plants is paving the way for new opportunities such as the use of artificial intelligence and machine learning solutions to improve the efficiency of processes. At the same time, Industry 4.0 brings with it various challenges that need to be solved. Machines equipped with various sensors generally provide extensive but heterogeneous data. In addition, data collected during manufacturing processes can be much noisier than data collected in a controlled (laboratory) environment. For this reason, the development of robust analysis pipelines and the evaluation of the information contained in the collected data are of crucial importance and interest from both a research and application perspective.

Resistance spot welding (RSW) is a popular welding process in the manufacturing industries, used to join sheet metals from different parts of bodies [1]. The main strengths of this technique are its speed, ease of automation, and low energy consumption, which makes it productive and cost-efficient. The RSW process is based on the heat generated by the electrical resistance when a high current flows through two electrodes that come into contact with the sheet metals, melting them and creating a weld nugget. The process typically consists of two phases: during the first phase, a force is applied to the sheets; in the second phase, an electric current flows through the electrodes to melt the materials forming the joint. The latter can be further subdivided into multiple phases, such as a warm-up phase, in which the electrode heats up, and a stationary phase, in which the actual welding takes place. The whole process lasts a few hundred milliseconds.

Automatic defect detection in RSW is a major challenge, mainly due to the high reliability of the process, which results in a low number of defects, and the variability of the process conditions. Machine learning (ML) pipelines can automatize this process. However, it is difficult to create a dataset with a high number of defective data points for the proper training of machine learning models. Another important challenge is the evaluation of weld quality, a fundamental step also to produce labeled datasets. It requires either destructive testing (peel or shear testing), which is impractical in a production environment, or costly, slow manual testing, which can also lead to inaccurate evaluations. In manual testing, *ultrasound testing* is the de facto industry standard for inspection. Specifically, this inspection is carried out in the form of either an A-scan or a C-scan [2], with the latter providing a comprehensive two-dimensional image of the fused area. These techniques require a high degree of skill by the operator and the inevitable removal of the manufactured body from the production line for inspection. This is clearly not suitable for mass production and forces companies to perform random quality checks on a small subset of welded bodies. This results in a scarcity of labeled datasets from running production plants. The literature is mainly concerned with controlled test environments. In fact, virtually all approaches reviewed in this article (see Section 2) consider welds obtained in a laboratory environment, which provide only limited insight into the problem under investigation and are hardly implementable in-plant. The only exceptions, taking into consideration real-world data, either do not consider the problem of fault detection [3,4] or approximate it using a proprietary quality measure (Q-Value) [5] or employ a different set of features, like images [6].

To summarize, the contributions of this work are as follows:

- We propose a publicly available dataset (anonymized for confidentiality reasons) collected and labeled by a leading automotive company in a running production plant to address the lack of labeled welding data. The dataset contains a variety of information that is automatically captured by the welding robots. This includes contextual information (such as electrode wear) and time series data on current, voltage, and force exerted during the welding process.
- We propose several pipelines for analyzing the collected data, employing different sets of features and combinations of them.
- We compare several machine learning (ML) and deep learning (DL) models trained along the different pipelines to investigate the feasibility of an automatic real-time fault detection system with the data automatically collected.
- We show that although this task appears feasible when weld quality is approximated by proxy variables (e.g., nugget diameter) or in controlled test environments, detecting welding faults in a production-like environment is still challenging and requires further research.

This article is structured as follows: We discuss the relevant literature and present the ML and DL models that form the foundation of this work in Section 2. Section 3 is devoted to the description of the datasets and presents the basic ML pipelines used in the experiments. Section 4 presents the experimental setup and the results for the different models performing a comparative analysis over three pipelines and two versions of the dataset. In Section 5, we critically discuss the results to address whether the automatically collected information is sufficient for the detection of welding defects, while Section 6 concludes the article and presents some research directions for future work.

2. Related Work

In this section, we review the ML models that have been employed in the context of welding data analysis. We see how these models have been employed to solve different tasks within this context and which performance they reported.

2.1. Machine Learning Models

In the analysis of welding process data, various machine learning models have been utilized to handle both static and time series data. The following is a non-exhaustive selection of the most common models used in this domain.

K-Nearest Neighbors (K-NNs). The k-nearest neighbors (K-NNs) algorithm is a simple yet effective model used in machine learning for both classification and regression tasks, including the analysis of welding process data. K-NN operates by identifying the k most similar training examples in the feature space, e.g., by computing their Euclidean distance. Then, K-NN makes predictions based on a majority (or average) voting schema in which the predicted label is the most common (or average) among the k neighbors. When applied to time series data, K-NN can be adapted to consider temporal dependencies by incorporating time series distance metrics such as Dynamic Time Warping (DTW).

Decision Tree and Random Forest. Decision trees make decisions based on a series of binary rules, effectively dividing the dataset into branches that lead to potential outcomes. They are particularly useful for static data from welding processes due to their interpretability and ease of use. Random forests, an ensemble of decision trees, improve predictive performance by reducing overfitting through averaging multiple trees, thus providing more robust predictions for complex welding data.

Time Series Tree-Based Models. When working over time series data, decision trees require a preprocessing step extracting descriptive features of the temporal evolution. This preprocessing allows tree-based models to capture both local and global patterns in the data, making it suitable for analyzing temporal sequences. As an example, Time Series Forest (TSF) is an ensemble method specifically designed for time series classification [7], which extracts interval-based features from time series data and utilizes a collection of decision trees, each one trained over a specific interval. The final prediction is then averaged over the prediction of each decision tree. Canonical Interval Forest (CIF) extends the concept of TSF [8]. It generates a diverse set of features from different intervals of the time series data, which are then used to train an ensemble of decision trees. A further improvement has been proposed in [9] in which the author proposed the Diverse Representation Canonical Interval Forest (DrCIF). This model further enhances CIF by incorporating a variety of representations of the time series data, leading to improved classification accuracy.

Convolutional Neural Networks. One-Dimensional Convolutional Neural Networks (1D CNNs) have shown great promise in time series analysis [10]. These networks leverage the ability of convolutional layers to capture local dependencies within the raw data, making them highly suitable for complex time series. In simple versions, CNN-1D consists of a few convolutional layers only, followed by pooling layers and fully connected layers. Inspired by similar models developed for 2D image classification, several extensions have been proposed. ResNet [11] is a complex architecture that includes skip connections, allowing the network to learn residual mappings to improve gradient flow during training and avoid the vanishing gradient problem. Inception Time [12] introduces inception modules, convolutional layers employed in parallel convolutional kernels of varying sizes, allowing the model to capture patterns at different scales simultaneously.

2.2. Welding Data Analysis

Several works in the literature deal with the prediction of weld quality by means of machine learning techniques. Classical models have limited application since the input data are inadequate to describe the transient conditions of the weld zone [13].

As the predictive objective of the machine learning models, however, there is no universal agreement on the definition of *quality* in the literature. Some papers consider the diameter of the nugget [3,4,14], while others the shear strength [15–17], i.e., the amount of shear stress the material can withstand. Some works consider both such metrics [18] or other manufacturer-specific measures, such as the Q-value in [5]. Other approaches, such as [6,19], address a problem closer to ours, namely, the classification of good and defective welds, but consider fundamentally different data sources: X-ray images in [19] and weld images captured by a camera on an assembly line in [6].

As for the models considered in the literature, they range from classical regression techniques [14,18], to standard machine learning techniques such as classification and regression trees [4] and random forests [20], and to deep learning models (e.g., Multi-Layer Perceptron (MLP) [17,21], CNN [6], or auto-encoders [19]).

Some studies consider time series as input for their models. Zhang et al. [15] consider the time evolution of electrode displacement (i.e., the distance between electrodes during the welding process) to predict quality, measured as tensile shear strength. They then use Chernoff surfaces to present their estimate, providing an immediate visual output. Summerville et al. [14] also look at time series, but this time, they take the voltage and current during the welding process. The aim of the work is to evaluate the nugget size based on the dynamic resistance, i.e., the ratio between the voltage and the current (available in-line from the welder). They use principal component analysis (PCA) (and correlations to prune the components) to determine the input of a multilinear regressor. The authors compare a C-scan test (which works well on aligned samples) with their approach, which appears to be more accurate and achieves an MSE of less than 0.5 mm. The ground-truth nugget size was measured using a destructive chisel test. In the context of Ultrasonic Testing (UT), Amiri et al. [17] investigated the relation between the UT testing results and the strength of RSW welds (under different loading conditions), leveraging the *pulse-echo data*. The authors trained a neural network using the number of echoes and the difference between consecutive echoes (first and second, second and third, and so on). They then used a genetic algorithm to optimize the network's structure. Sun et al. [16] use the current, voltage, and force of the electrode fed to a single-layer feed-forward neural network optimized with a particle swarm optimization algorithm to evaluate the quality of welds in terms of shear strength, achieving high accuracy. They take into account only a few features associated with the collected time series, such as the power or the effective value of the current.

Another approach considered in the literature uses images of welds to perform either classification or regression tasks. For example, Ye et al. [22] enhance and segment images of welds and extract features that are fed into a small neural network that performs binary fault classification. In [23], the authors discuss the limitations of simple image segmentation approaches and apply semantic segmentation instead. In a recent paper [6], the authors propose a novel architecture based on MobileNetV3 and multi-scale feature fusion (to combine low-level features and high-level semantic features). Hou et al. [19] use X-ray images of the welds instead of weld images to train (stacked) sparse auto-encoders in an unsupervised manner. Then they drop the decoding layers and add a softmax classifier to perform the classification task, achieving high accuracy. While interesting, this research work relies on visual data, which may be difficult to obtain in a production environment.

Gavidel et al. [3] have an approach that is closer to ours in some respects. They compare several techniques (support vector machines, decision tree, random forest, K-NN, etc.), but consider the estimation of the nugget diameter, for which they propose a Deep Neural Network. They have 913 welds from an American manufacturer with information on material, thickness, coating, force, current, and welding time. Among the approaches they considered, the Deep Neural Network (DNN) achieves the smallest mean absolute error. They also calculate the importance of the features and then determine (by majority voting) that the top 3 most significant features are welding current, welding force, and details on the coating of the material. However, they did not have access to electrode morphology features, tip wear, or information on expulsion. Interestingly, K-NN is the model with the second-best performance. Another approach that draws on data, this time from an automotive equipment manufacturer, is [4], which builds a knowledge database with decision rules extracted from classification and regression trees trained on their welding dataset. Their goal is to predict the diameter width and extract new knowledge about the welding process.

Finally, Zhou et al. [5] provide a complete overview of the literature (see Table 1 in [5]) and tackle the task of quality monitoring, for which they use the Q-Value, a score developed by Bosch Rexroth that is calculated using process curves and other process information incorporating domain knowledge. The authors take the novel perspective of considering the cyclical nature of the welding process rather than looking at each weld in isolation. With this goal in mind, they attempt to predict the welding quality at a future point in time by using the information from previous times and knowing the features of the next welding operation. They define features based on the process and time series information and then extract a subset of it. They consider four settings with increasingly engineered features (setting 0 consists only of raw features). As for the models, they use linear regression, multi-layer perception, and support vector regression.

From the analysis of the literature, we identify a significant research gap, as a comprehensive comparison of different ML techniques is currently lacking, especially with respect to their application to production-like data. Most works consider either a single technique [4,6,7,14–17,19] or a small set of techniques [5,20], with the exception of [3], which compares different techniques. We consider existing techniques and apply them to a real-world scenario with data from a running production plant. In contrast with other works (e.g., Ref. [3], which also performs an algorithm comparison with production-like data), we use a larger amount of information that includes both static process-specific information (similar to [3]) and time series of the welding-related variables (voltage, current, and force). In this way, we can discuss in more detail the sources of information that are most promising for welding fault prediction. We also consider ensemble methods combining the best-performing approaches from our comparison, an approach that has not yet been explored in the current literature.

3. Materials and Methods

The primary objective of this work is to develop different ML pipelines that use data collected automatically from the welding machines, such as context information and time series of process variables, to predict defective welds. We use the data to train and evaluate a number of ML models, and we eventually combine the best models to exploit the strengths of each approach.

In the following, we first describe the RSW dataset under consideration (Section 3.1). Since there is a lack of publicly available spot weld datasets in the literature, we also publish an anonymized version of the dataset at https://github.com/smartdatapolito/resistance_spot_welding_dataset (accessed on 4 August 2024). Second, we describe the ML pipelines used, as described in Figure 1. We consider classifiers working with static features that we apply to both contextual information and engineered statistical features extracted from the raw time series (Section 3.2), classifiers working directly with the raw temporal data (Section 3.3), and ensemble models comprising the best models working with different modalities and combining the predictions of all of them (Section 3.4).

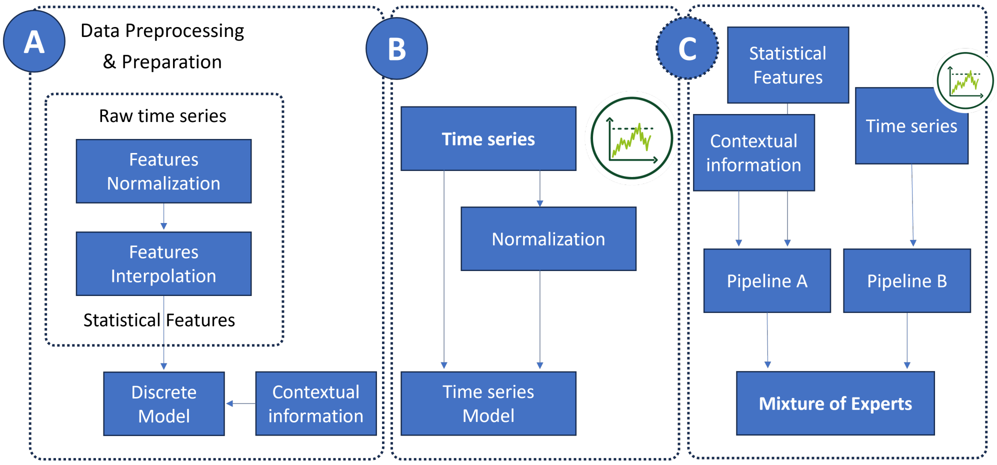


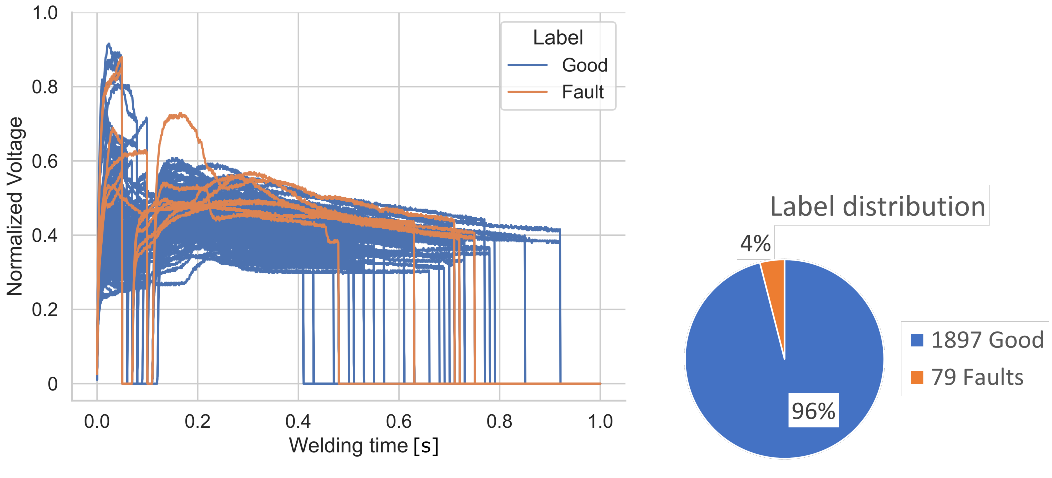
Figure 1. Different types of ML pipelines employed. Pipeline (**A**), a classifier working on static data that analyzes contextual information, discretized statistics of the time series, or a combination of them. Pipeline (**B**), a time series model that directly observes the raw time series or after a normalization step. Pipeline (**C**), an ensemble model combining the predictions of different classifiers possibly working over different features and employing different ML pipelines.

3.1. Dataset

In this work, we use a dataset composed of 1976 samples describing as many real-world resistance spot weldings collected in an automotive production facility for over 3 years (2022–2024).

The samples originate from 158 different welding spots (different welding locations over the vehicle body). Among the 1976 weldings, 1011 involve two layers of metal, while the remaining welds were performed over three layers, resulting in a total of 16 different material combinations, e.g., a weld of steel (FE) and chromium (CR). The 1976 samples are described by two sets of features: discrete *contextual information* describing the welding characteristics and *time series* corresponding to the temporal evolution of the variables involved in the welding process.

The data were manually labeled using ultrasound testing over a sample of the car bodies produced during this period. As can be seen in Figure 2b, we have a highly unbalanced distribution, with 96% good welds and only 4% defects. While this distribution is not ideal for machine learning, it reflects the real distribution of faulty welds in an operating plant. In Section 3.3, we discuss a possible solution to this problem.



(**a**) (**b**)

Figure 2. Dataset qualitative analysis. (**a**) A visualization of welding voltage time series data. We show 10% of the curves to improve figure readability. (**b**) Distributions of the labels in the dataset: only ∼4% of the data correspond to a fault.

Contextual information, denoted as $x_c \in \mathcal{X}_m \subseteq \mathbb{R}^D$, describes D characteristics of each weld, such as a unique identifier of the welding machine and spot, the chassis, the time of data collection, the type and the thickness of the layers to be joined, and manufacturer-specific wear parameters that indicate, for example, the frequency of electrode maintenance. We will refer to them as "Metadata". As shown in Table 1, in the considered dataset, the contextual data amount to $D = 55$. Among these, we mention the expulsion time, a variable that signals when expulsion has occurred during welding (0 if no expulsion occurred); the wear; and the wear percentage, which describe the condition of the electrode. The tipDressCounter monitors how often the electrode has been used since the last dressing, the UIP actual index, which is a quality index related to the voltage, current, and welding time, and finally, there is a stabilization index, which embodies the deviation from the reference—the farther away from 100, the greater the difference. In Section 4, we will discuss which of these values mainly drive the model's decisions.

Table 1. Preprocessed dataset characteristics.

	Pipeline A	**Pipeline B**	**Pipeline C**
# Samples	1976	1976	1976
# Features	$\lvert\mathcal{X}_m\rvert = 55, \lvert\mathcal{X}_e\rvert = 106$	$\lvert\mathcal{X}_t\rvert = 3 \times 1000$	$\{\lvert\mathcal{X}_m\rvert, \lvert\mathcal{X}_e\rvert\} + \lvert\mathcal{X}_t\rvert$

Time series data represent the voltage, current, and force applied by the electrodes for a particular weld. We refer to these as $x_h(t) \in \mathcal{X}_t \subseteq \mathbb{R}^{H,T}$, where we use H to indicate the number of time series variables considered and T to indicate the duration of the time series. For ease of use, all the time series have been zero-padded to have the same length, as shown in Figure 2a. In particular, in the considered dataset, the padded duration of the time series accounts to $T = 1000$, and the number of considered time series variables (H) ranges from 1 to 3, as shown in Table 1.

The welding time series for the examined welds usually consists of two distinct phases for what concerns voltage and current. Some examples of voltage curves can be found in Figure 2a. The first phase is used to warm up the electrode and is usually short, followed by a period with zero signal. The second phase, on the other hand, is the one in which welding is actually carried out. It begins with a rapid increase, followed by a constant or slightly decreasing signal that lasts longer than the first phase. The total duration of welding, as well as the percentage duration of the two phases, depends on the two welding spots.

3.2. Pipeline A: Classifier Working over Static Data

In this pipeline, we consider standard ML classifiers working over static (i.e., not evolving over time) data; see the left part of Figure 1. We can define the learning problem as $f : \mathcal{X}_s \to \mathcal{Y}$, where $\mathcal{Y} \in [0, 1]$ is the set of welding labels, representing with $y = 1$ a faulty weld and with $y = 0$ a good one, and $\mathcal{X}_s$ is the set of static features.

In the simplest scenario, these static data correspond to the contextual information provided by the welding machine, i.e., $\mathcal{X}_s = \mathcal{X}_m$. This information can be used directly without any preprocessing, except for a normalization of the numerical features (we used a [0, 1] normalization) and one-hot encoding for categorical data.

In a more sophisticated scenario, Pipeline A can also take into account statistical features extracted from the time series $x_h(t) \in \mathcal{X}_t$. In other words, we define a mapping $m : \mathcal{X}_t \to \mathcal{X}_e$, where we denote as $\mathcal{X}_e \subseteq \mathbb{R}^E$ the E extracted engineered features that capture the information contained in the time series. To do this, we first apply a normalization that depends on the individual welding spot (and its year). This allows us to train a classifier that is independent of the spot weld under consideration. Ideally, we should develop a classifier for each spot weld due to the inherent differences between the spot welds, but this is not practical due to the scarcity of labeled training data. Acknowledging the differences in the welding phases, we first divide the time series into two phases: the electrode warm-up phase and the actual welding phase, and we normalize each phase independently in the interval $[0, 1]$.

Since the welding process differs greatly between different welding spots and also between years due to changes in the plant operations, we group the data by spot name and by year and normalize each group separately. Furthermore, to homogenize the duration of the phases across welding spots, we also perform an interpolation to obtain time series data with the same duration. Such preprocessing makes it possible to use all points together to train a model, regardless of the considered welding spots. Finally, in collaboration with the domain experts, we summarize the information content of the normalized data in the form of statistical features. We have derived a list of aggregated features for each phase: max, min, mean, standard deviation, percentiles (1, 5, 10, 25, 75, 95, 99), actual welding duration, and energy. We repeated the feature extraction and normalization process for each of the time series variables, i.e., voltage, current, and force.

The set $\mathcal{X}_e$ represents the engineered features extracted from all the time series (current, voltage, and force), plus the time series duration. However, we can also consider different subsets $\mathcal{X}_{e'} \subseteq \mathcal{X}_e$, which isolate information from a particular time series (e.g., $\mathcal{X}_c$ encompasses only features from the current time series and comprises 35 features). This information can possibly be used in combination with the context information $\mathcal{X}_m$. All these combinations will be explored in Section 4.

3.3. Pipeline B: Classifier Working over Temporal Data

In this pipeline, we explore the application of time series classifiers to directly analyze time series data collected automatically during the RSW process, as in Figure 1, center. More specifically, we aim to learn a function $g : \mathcal{X}_t \rightarrow Y$. In contrast with Pipeline A, which uses classifiers on static data (context features and engineered features), Pipeline B aims to directly utilize the temporal dynamics of the process data to predict defective welds.

We consider the raw time series of voltage, current, and force, zero-padded to make them of the same length. Since the dataset is very unbalanced, we also consider a subsampled version of it, $\mathcal{X}_{t'} \subseteq \mathcal{X}_t$ ("Sub."). This makes the dataset balanced and better suited for training models. In fact, complex models (such as time series analysis) suffer even more from the imbalance of the dataset. Subsampling is a possible technique to deal with an unbalanced dataset.

3.4. Pipeline C: Ensemble Model Combining the Predictions of a Classifier from Different Pipelines

Pipeline C focuses on the development and evaluation of ensemble models that aggregate the predictions of the individual models from Pipeline A (static data classifiers) and Pipeline B (time series classifiers). The goal is to leverage the strengths of different models to improve the overall accuracy of defect detection in resistance spot welding. In principle, we could apply ensemble methods to all the combinations of models trained in Pipelines A and B. However, to limit the search space, we only aggregate the predictions of the best-performing model from Pipelines A and Pipeline B.

Various ensemble methods can be employed to combine the predictions of the selected base models, such as Average Voting, Weighted Voting, and Meta-Learning. Average voting aggregates prediction probabilities by taking the average vote from the base models. Each model outputs the fault probability: the sample is then considered faulty in case the average probability is higher than 0.5. In Weighted Voting, models are assigned weights based on their individual performance during training. The final prediction is determined by the weighted sum of the votes. In Meta-Learning, the predictions of the base models are used as input features to a meta-learner, which is typically a simple model like logistic regression or a more complex model like a gradient boosting classifier. The meta-learner is trained to make the final prediction based on the combined inputs.

4. Experiments

In this section, we describe our experimental setup (Section 4.1) and present the results (Sections 4.2 and 4.3). Within each section, we discuss the implications of our results.

The aim of our experiments is twofold. First, we use the entire dataset described in Section 3.1, following pipelines A–B–C, and we train and evaluate the merits of different ML approaches. Furthermore, we seek to understand what influenced the decision of the models, either by applying explainability methods (such as feature importance) or by examining the correctly identified faulty welds. Second, since this last analysis indicates a bias in our dataset, we remove some specific data points and repeat the analysis following the different pipelines (Section 4.3).

4.1. Experimental Setting

First, we consider approaches that make use of static data (Pipeline A). We study the *Metadata* $\mathcal{X}_m$ collected by the welding robot for each weld.

We also use the engineered features $\mathcal{X}_e$ *(EF)* extracted from the time evolution for each time series *Voltage (EF)*, *Current (EF)*, *Force (EF)*. With *All (EF)*, we refer to the case in which all voltage-, current-, and force-related features are used together. Finally, we refer to the combination of approaches with a "+" sign. For instance, *Voltage (EF) + Metadata* uses jointly the engineered features from voltage and the metadata. We use different ML models specific for tabular data, i.e., a decision tree (*DT*) with maximum five splits, a random forest (*RF*) with 100 trees, and k-nearest neighbors (*K-NN*), where we employed different numbers of neighbors from 1 (*1-NN*) to 6 (*6-NN*).

Second, we consider the time series *(TS)* using Pipeline B. We consider either the single time series ($h = 1$) or all the time series together (ALL, $h = 3$). Additionally, to balance the dataset, we subsample the data points in the majority class, i.e., "Good" welds, to have a more balanced dataset (*Sub.*). As models, we employ a Time Series Forest (*TSF*) [7], Canonical Interval Forest (*CIF*) [8], Diverse Representation Canonical Interval Forest (*DrCIF*) [9], different CNN configurations, a simple *CNN*, a *ResNet* CNN [11], and an *Inception* CNN [12] trained with a batch size equal to 8 for 200 epochs, and, lastly, the *K-NN* specialized for time series data, which considers the Euclidean distance between the curves. We point out that we also considered other distances for the K-NN (e.g., DTW) that exhibited similar or slightly worse performances and are omitted here for brevity.

Third, we consider the best model based on static data and the best model based on time series, and we test three ensemble methods (Pipeline C) to try to exploit the joint effect of the two approaches. More in detail, we employ a shallow *Average Voting* schema, a *Weighted Voting* schema, and a *Meta-Learning* approach based on a random forest classifier. The last two ensemble techniques are calibrated over the training data predictions of the best models discovered in the two previous pipelines.

Unless otherwise stated, we use classical k-fold cross-validation with $k = 5$; i.e., we divide the dataset into k folds of equal size, train the model over $k - 1$ folds, and test it with the remaining subset. The procedure is repeated k times. As evaluation metrics, we first use the F1 score, which reflects the model's ability to correctly identify positive cases and its precision in the cases it deems positive. It is formally defined by two other measures, namely, *precision* and *recall*. Precision represents the fraction of *true* positives across all positively classified records, i.e., $\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}}$. The recall, on the other hand, represents the fraction of positively classified data records over all truly positive data records, i.e., $\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}}$. The F1 score can then be defined as follows:

$$\mathrm{F1} = \frac{2}{\frac{1}{precision} + \frac{1}{recall}} = \frac{2\mathrm{TP}}{2\mathrm{TP} + \mathrm{FP} + \mathrm{FN}} \tag{1}$$

Note that considering as a performance metric the *accuracy* (i.e., the fraction of correctly classified examples out of the total number) is not appropriate in a highly imbalanced environment like ours. Indeed, a void classifier that always outputs "Good" would achieve an accuracy of 96%.

As a second performance measure, we consider the *confusion matrix*, a representation used to understand how well an algorithm predicts different classes. The rows represent the *actual* classes, and the columns are the *predicted* classes, so the diagonal elements are the correctly predicted instances. Since we consider the "positive" class as the "NOK" class (we recall that our task is fault detection), the first row of the matrix contains the True Negatives (TNs), i.e., the correctly classified non-defective (good) welds, and the False Positives (FPs), i.e., the good welds classified as defective (in a production environment, this value should be as low as possible, as it represents a situation of *false alarm*). The second row contains the False Negatives (FNs), i.e., the faulty welds that were not identified by the algorithm, and the True Positives (TPs), i.e., the correctly classified faulty welds.

For training the static models, we employed the `sklearn` (https://scikit-learn.org, accessed on 5 August 2024) library, while for the time series one, we employed `sktime` (https://www.sktime.net, accessed on 5 August 2024) for *K-NN*, *TSF*, *CIF*, and *DrCIF*, and we used `aeon` (https://www.aeon-toolkit.org, accessed on 5 August 2024) for the CNN-based models. For all algorithms, we left the default hyperparameters unless for those above-stated. We trained all models over an Intel i7-13700H (Intel, Santa Clara, CA, USA) equipped with 16 GB of RAM and an Nvidia 4060 GPU (Nvidia Corporation, Santa Clara, CA, USA) with 6 GB VRAM.

4.2. Training over the Entire Dataset

In Table 2, we report a comparison of the various models for Pipeline A with different sets of features, when considering the entire dataset. The best absolute approach in terms of the F1 score is the *Current (EF) + Metadata* with *DT*, i.e., the one considering the features extracted by the current data series and the metadata (contextual information) with a decision tree, which reaches an F1 score of 0.729. However, the best set of features, on average (last column), is *All + Metadata*, i.e., the set including the features extracted from all the data series, and the metadata (the most comprehensive set).

Table 2. F1 score comparison of models working on discrete and statistical features (Pipeline A). Lighter colors denote better results.

Features	RF	DT	1NN	2NN	3NN	4NN	5NN	6NN	AVG
Metadata	0.675	0.700	0.631	0.566	0.634	0.591	0.620	0.562	0.622
All (EF)	0.604	0.653	0.621	0.601	0.684	0.617	0.658	0.599	0.630
Voltage (EF)	0.575	0.654	0.596	0.580	0.650	0.612	0.626	0.593	0.611
Current (EF)	0.489	0.526	0.603	0.509	0.550	0.500	0.520	0.490	0.523
Force (EF)	0.565	0.621	0.592	0.596	0.623	0.595	0.621	0.592	0.601
All (EF) + Metadata	0.673	0.723	0.633	0.594	0.680	0.641	0.699	0.624	0.658
Voltage (EF) + Metadata	0.694	0.678	0.655	0.590	0.669	0.605	0.692	0.582	0.646
Current (EF) + Metadata	0.688	0.729	0.636	0.572	0.662	0.606	0.677	0.601	0.646
Force (EF) + Metadata	0.682	0.705	0.623	0.562	0.657	0.594	0.676	0.583	0.635

In Table 3, we present the performance of the different models based on Pipeline B. Unlike in the previous case, the models that use the *Voltage (TS) Sub.* information perform best on average, with and F1 score of 0.658. Nevertheless, the best-performing approach is the one that uses the *CIF* model over the subsampled version of all time series (*ALL (TS) Sub.*) with an F1 score equal to 0.744, with *DrCIF* coming as a close second (0.737 F1 score). Additionally, note that subsampling the "Good" class (*Sub.*) increases the average performance by 1–3% in all cases. In line with the previous results, considering *ALL* time series is also a good approach, comparable with the *Voltage (TS)* one, and superior when employing *CIF, DrCIF* and *Inception* CNN. Finally, note that the two empty cells in Table 3 are due to the fact that the ML library used does not allow multiple time series as input in the TSF model.

Table 3. F1 score comparison of models working on time series features (Pipeline B). Lighter colors denote better results.

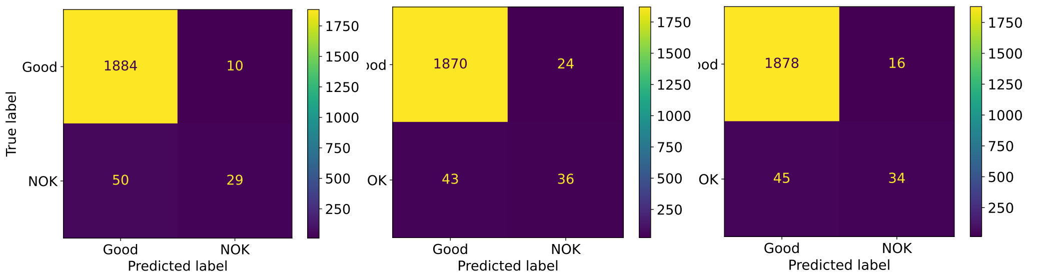
Features	1NN	2NN	3NN	4NN	5NN	6NN	TSF	CIF	DrCIF	CNN	ResNet	Inception	AVG
Voltage (TS)	0.653	0.645	0.714	0.655	0.737	0.711	0.717	0.678	0.682	0.490	0.528	0.689	0.658
Current (TS)	0.613	0.561	0.616	0.563	0.636	0.590	0.594	0.589	0.546	0.490	0.400	0.522	0.560
Force (TS)	0.595	0.604	0.631	0.574	0.664	0.618	0.561	0.558	0.571	0.490	0.501	0.577	0.579
All (TS)	0.638	0.637	0.659	0.660	0.707	0.684	-	0.688	0.697	0.490	0.445	0.695	0.636
Voltage (TS) Sub.	0.666	0.701	0.677	0.726	0.722	0.717	0.723	0.731	0.737	0.493	0.431	0.620	0.662
Current (TS) Sub.	0.615	0.620	0.635	0.631	0.629	0.634	0.658	0.630	0.662	0.493	0.441	0.513	0.597
Force (TS) Sub.	0.610	0.652	0.668	0.664	0.683	0.665	0.586	0.636	0.642	0.509	0.410	0.568	0.608
All (TS) Sub.	0.642	0.657	0.664	0.684	0.691	0.683	-	0.744	0.737	0.493	0.587	0.641	0.657

Finally, Table 4 reports the performance of the three tested ensemble techniques when working over the best model based on static data, i.e., the *DT* using *Current (EF) + Metadata* and the best model based on time series, i.e., *CIF* working over *All (TS) Sub.* time series data. We can notice how both *Weighted Voting* and *Meta Learning* results are more effective than both best models. The best ensemble model reaches an F1 score of 0.752.

Table 4. F1 score comparison of ensemble techniques (Pipeline C) when employing the two best models from previous pipelines. Lighter colors denote better results.

Features	Average Voting	Weighted Voting	Meta Learning	AVG
Ensemble	0.729	0.752	0.748	0.743

We further evaluate the models by looking at the confusion matrices of the best approach from each Pipeline in Figure 3. Confusion matrices provide deeper insights into the predictive capabilities of a model. Domain experts in the automotive sector are particularly interested in the second column of this matrix (FP and TP, from the top). They clearly want the TP to be as high as possible as it represents the correctly classified defective welds. However, they also want the FP, i.e., the welds that are classified as defective but are actually flawless, to be as low as possible. High rates of FP can lead to unnecessary production halts for inspections, thereby reducing productivity due to increased downtime. The first two approaches differ considerably with respect to the confusion matrices. The first approach (Figure 3a) is able to correctly classify 29 of the 79 defects in the dataset (TP), while the second (Figure 3b) performs better and correctly identifies 36 defective welds. However, this comes at the cost of producing many more *false alarms*, as the class of FP grows from only 10 in the first approach to more than double (24) in the second. There is, therefore, a trade-off between the ability to detect faults and the generation of false alarms. In Figure 3c, the Weighted Ensemble method demonstrates its ability to combine the strengths of the two methods, maintaining a high number of TPs (34) while substantially lowering the FPs to 16 (as required by domain experts). This method, as indicated by the higher F1 score, appears to be the most promising.



(**a**) (**b**) (**c**)

Figure 3. Confusion matrices for the best approaches of Section 4.2, following Pipeline A: DT over *Current (EF) + Metadata* features (**a**), Pipeline B CIF over *ALL* time series subsampled (**b**), and Pipeline C Weighted Ensemble of the models on the left (**c**).

Critical Analysis

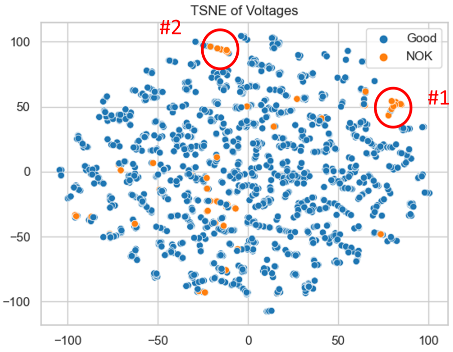
As a further step in understanding the decision process behind the trained model decisions, we determined the *feature importance* for the *DT* approach and the t-distributed stochastic neighbor embedding (t-SNE) [24] over the time series data.

The *feature importance* quantifies the relevance of the input features in predicting the target, and it is reported in Figure 4a for the *DT*. Interestingly, the most important feature is the *expulsion time*, which indicates whether and at what time an expulsion (ejection of molten metal) took place during the welding process. The second and third features refer to the (numerical) identifier of two specific welding spots, which we refer to as #1 and #2. Additionally, the most important remaining features are some characteristics of the first welding phase. When manually inspecting the two welding spots (#1 and #2), we found that spot #1 is defective in over 80% of the cases, while spot #2 is defective in over 60% of the cases. This points to a possible bias of the faults in certain spots.

Similar conclusions can be drawn from Figure 4b, in which t-SNE is applied to the *Voltage* time series. t-SNE enables the visualization of high-dimensional data (such as time series) and gives an idea of how "close" the data points can be in a reduced space. In Figure 4b, we see that the defects cluster mainly in certain areas, and these areas correspond to the welding spots #1 and #2.

Features	Importance
Expulsion time	0.2438
Spot name #1	0.2384
Spot name #2	0.1724
Current first phase std	0.0668
Current first phase 75th	0.0466
Car body #1	0.0309
Current first phase 25th	0.0193
Current second phase 95th	0.0162
Car body #2	0.0161
Resistance value	0.0139

(**a**)



(**b**)

Figure 4. (**a**) Feature importance of the best model of Pipeline A, i.e., DT on *Current (EF) + Metadata*. (**b**) t-SNE on *Voltage* time series. We can identify two clusters corresponding to two specific welding spots.

4.3. Removing the Bias in the Dataset

These previous results hint that there may be a bias in the dataset at hand. Therefore, we decided to remove the two overly defective spots (#1 and #2), reducing the number of defective welds from 79 to 54. In the following, we report the same scenarios explored in Section 4.2, applying Pipeline A, Pipeline B, and Pipeline C to this *filtered* version of the dataset.

Table 5 shows the results of Pipeline A. When comparing these results with those obtained on the entire dataset, we can appreciate a consistent drop in performance. In particular, while the previous best result for Pipeline A attained an F1 score of 0.729, on the filtered dataset, the best approach for Pipeline A (i.e., the *DT* working over the *Metadata*) attains an F1 score of 0.606. The percentage performance drop amounts to ≈17%.

Table 6 reports the result of Pipeline B. Additionally, in this case, the performance has degraded with respect to that over the entire dataset. The previous best result in Table 3 reached an F1 score of 0.744, while in this case, we obtain at most 0.657, with the *CIF* model working over *Voltage (TS) Sub*. The percentage performance drop in this case amounts to ≈12%, suggesting that the *Metadata* may have been more affected by the bias. Indeed, the spot names of the removed welding points were identified as the second and third most important features.

Table 5. F1 score comparison of models working on discrete and statistical features (Pipeline A) on the *filtered dataset*. Lighter colors denote better results.

Features	RF	DT	1NN	2NN	3NN	4NN	5NN	6NN	AVG
Metadata	0.540	0.606	0.564	0.542	0.571	0.525	0.533	0.521	0.550
Voltage (EF)	0.493	0.514	0.530	0.492	0.502	0.492	0.524	0.492	0.505
Current (EF)	0.493	0.504	0.544	0.523	0.544	0.522	0.538	0.510	0.522
Force (EF)	0.546	0.594	0.565	0.521	0.596	0.540	0.598	0.574	0.567
All (EF)	0.509	0.577	0.553	0.527	0.568	0.509	0.519	0.521	0.536
Voltage (EF) + Metadata	0.541	0.561	0.596	0.542	0.559	0.538	0.563	0.520	0.553
Current (EF) + Metadata	0.528	0.576	0.601	0.526	0.605	0.540	0.548	0.524	0.556
Force (EF) + Metadata	0.524	0.599	0.599	0.566	0.582	0.523	0.521	0.511	0.553
All (EF) + Metadata	0.527	0.565	0.568	0.540	0.599	0.509	0.552	0.509	0.546

In Pipeline C, we consider the *DT* working over the *Metadata* as the best model for Pipeline A and the *CIF* working over *Force (TS) Sub.* features for the model from Pipeline B. In Table 7, we report the results of the three considered ensemble methods. In this case, the performance is lower than the original pipeline B approach. This is likely due to the low performance of the best model working along Pipeline A, which mostly introduces noises in the predictions of the ensemble models.

Table 6. F1 score comparison of models working on time series features (Pipeline B) on the *filtered dataset*. Lighter colors denote better results.

Features	1NN	2NN	3NN	4NN	5NN	6NN	TSF	CIF	DrCIF	CNN	ResNet	Inception	AVG
Voltage (TS)	0.589	0.536	0.627	0.589	0.599	0.542	0.612	0.628	0.585	0.493	0.492	0.625	0.576
Current (TS)	0.568	0.492	0.519	0.507	0.505	0.493	0.492	0.493	0.493	0.493	0.493	0.478	0.502
Force (TS)	0.568	0.582	0.612	0.601	0.639	0.628	0.598	0.618	0.609	0.493	0.492	0.571	0.584
All (TS)	0.558	0.509	0.546	0.523	0.551	0.509	-	0.610	0.612	0.493	0.493	0.561	0.542
Voltage (TS) Sub.	0.616	0.557	0.606	0.605	0.617	0.628	0.643	0.650	0.652	0.520	0.244	0.589	0.577
Current (TS) Sub.	0.549	0.583	0.574	0.545	0.567	0.542	0.576	0.599	0.612	0.510	0.411	0.512	0.548
Force (TS) Sub.	0.580	0.612	0.618	0.653	0.657	0.642	0.610	0.657	0.612	0.536	0.460	0.609	0.604
All (TS) Sub.	0.575	0.573	0.587	0.599	0.570	0.594	-	0.634	0.629	0.510	0.541	0.570	0.580

Table 7. F1 score comparison of ensembles techniques (Pipeline C) when employing the two best models from previous pipelines and working on the *filtered dataset*. Lighter colors denote better results.

Features	Average Voting	Weighted Voting	Meta Learning	AVG
Ensemble	0.607	0.603	0.614	0.608

5. Discussion

Resistance spot welding is a widely adopted process in the automotive industry that is known for its high reliability and ease of automation. For this reason, detecting the few defective welds in a production environment is of high importance. In contrast with experiments in laboratory-controlled conditions, in a production environment, there is a wide variety of welding conditions and process parameters (e.g., see Figure 2a), making fault detection significantly more challenging. Previous studies, often conducted in controlled laboratory settings, do not adequately capture the variability and unpredictability present in real-world production lines. This high variability in conditions and process parameters impacts the consistency and reliability of weld quality predictions. Additionally, due to the high reliability of RSW, production data are intrinsically highly unbalanced: the dataset has a much higher number of non-faulty welds than faulty ones (e.g., see Figure 2b). This imbalance becomes even more pronounced when addressing intrinsic biases in the data, such as removing data points that are predominantly faulty.

In summary, our study has revealed the following insights. Pipeline A classifiers based on discrete metadata and statistical features show limited performance, particularly when biased samples are removed. These features alone provide little discriminatory power to predict weld faults accurately. In this pipeline, the most informative set of features is the metadata with the statistical features extracted from the voltage time series (i.e., Voltage

(EF) + Metadata) or from all temporal variables (i.e., All (EF) + Metadata). The best model on average is the decision tree.

Pipeline B models that analyze raw and normalized time series data perform better. The temporal dynamics captured in these features offer more insights into the welding process, leading to improved, though still limited, predictive accuracy even in the filtered scenario. In this case, the most informative temporal variable is again the Voltage, particularly when we subsample the more represented class (i.e., Voltage (TS) Sub.). In this case, the model associated with the highest performance is CIF both on the standard and on the filtered dataset, where all model performance is normally lower.

Pipeline C methods combining predictions from Pipelines A and B offer more improvements. When averaging over the two experiments, Meta-Learning seems to be the best technique for mixing the predictions of the models trained on the previous pipelines. Overall, we believe that this is the best technique, although it is slightly more complex and requires employing both pipelines (and both preprocessing) at the same time. When biased samples are excluded, however, the improvements from ensemble methods are limited, since the performance is hindered by the lower accuracy of models relying on metadata and statistical features.

Guidelines

Between the Pipeline A, Pipeline B, and Pipeline C approaches, there is a trade-off between performance and computational cost. The first models require less computational resources and are suitable for real-time predictions directly on the production line, even with modest computational resources. The approaches in Pipelines B and C, meanwhile, require longer training and more computational resources, making them more difficult to use directly on the production line.

Considering only the first experiment (using the entire dataset), it seems unjustified to use models based on the time series data, which offer only a limited improvement in performance despite the dramatic increase in resource requirements: $F1 = 0.729$ (best model of Pipeline A) vs. $F1 = 0.744$ (best model of Pipeline B), an improvement of 2%. However, the second experiment with the *filtered* dataset shows that the models based on data series are more robust and perform consistently better (see Table 6), with CIF or DrCIF outperforming virtually all Pipeline A approaches. Pipeline B's best approach guarantees a gain of almost 8% compared with Pipeline A's best approach. On the contrary, Pipeline C's approaches that combine the predictions of the best models from previous pipelines demonstrate some but limited improvements. In particular, they seem to be conditioned by the worse performing model, particularly when there is an important performance gap between the two considered models (e.g., Table 7). In this case, the performance is worse than considering a single model based on time series data.

To summarize, while using static features seems more appealing (and is the most common approach in the literature), models based on time series data can better capture the specifics of the welding process without employing shortcuts in the decision process. Indeed, they perform reasonably well, even removing the inherent bias of the dataset. Mixing the two approaches with an MoE seems to be a promising approach, but the current performance is limited, and it may require further research.

6. Conclusions

In this paper, we conducted a comparative analysis of machine learning models for detecting welding faults on a real-world dataset. We propose several pipelines working over different sets of features and combinations of them. Unlike the recent literature, mostly testing in laboratory conditions, we show that fault detection in real-world conditions is an intrinsically difficult task, even when employing sophisticated machine learning and deep learning pipelines. Additionally, differently than the current literature, we have shown that models analyzing static, contextual data are more prone to relying on biases in the dataset. On the contrary, analyzing time series data associated with the evolution of welding variables during the welding process, as proposed in this work, can capture more meaningful information, although with a higher computational cost.

Future work. To address the limitations observed in our study, in the following, we present possible solutions and directions for future research. (i) Expanding the data collection, e.g., the amount of data for each spot name, could help in mitigating the effects of data imbalance, thus enhancing the model's ability to learn and generalize. (ii) Incorporating additional data sources could also improve fault detection capabilities by providing information that is not available from process data alone. Potential data sources might include the following: (a) Images of the welds—a visual inspection can provide additional context and help identify surface defects not captured by process parameters alone [19,22,23]. (b) Thermal imaging—thermocameras can detect thermal profiles and anomalies occurring during the welding process [25]. (c) Acoustic signals—emissions during welding can reveal underlying issues in the weld quality, as demonstrated in related studies on Variable Polarity Plasma Arc Welding [26]. (iii) Improving data labeling with more refined scanning techniques such as C-scan may provide better quality annotations, thus enhancing the accuracy and reliability of the trained machine learning models [14]. In conclusion, while machine learning models can provide some predictive capability for weld faults using automatically collected contextual and process-specific data, their accuracy remains limited in a production environment. Addressing data imbalance and incorporating more diverse data sources are crucial steps toward developing more robust and reliable fault detection systems for resistance spot welding.

Author Contributions: Conceptualization, G.C., F.G., D.G., T.C. and M.M.; methodology, G.C., F.G., D.G., T.C. and M.M.; software, G.C. and F.G.; investigation, G.C. and F.G.; data curation, G.C., F.G. and D.G.; writing—original draft preparation, G.C. and F.G.; writing—review and editing, D.G., T.C. and M.M.; supervision, D.G., T.C. and M.M.; funding acquisition, T.C. and M.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the project MANAGE 5.0 (MANufacturing Automotive Green Evolution 5.0—F/310302/01-05/X56) under the Fondo Crescita Sostenibile—ACCORDO PER L'INNOVAZIONE.

Data Availability Statement: We will provide an anonymized version of the used labeled dataset upon paper acceptance.

Acknowledgments: We extend our gratitude to our industrial partners at the automotive company for the invaluable support they provided, namely, Alessandro Cacciatore, Rosemma Gario, and Giorgio Pasquettaz. Their provision of data and their participation in insightful discussions were essential to the success of this research. We deeply appreciate their collaboration and expertise, which greatly enhanced the quality and impact of our work.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

RSW	resistance spot welding
ML	machine learning
DL	deep learning
CNNs	Convolutional Neural Networks
DNN	Deep Neural Network
K-NNs	k-nearest neighbors
DT	decision tree
RF	random forest
TSF	Time Series Forest

CIF	Canonical Interval Forest
DrCIF	Diverse Representation Canonical Interval Forest
MLP	Multi-Layer Perceptron
DTW	Dynamic Time Warping
PCA	principal component analysis
UT	Ultrasonic Testing
EF	engineered features
TS	time series
Sub.	subsampled
t-SNE	t-distributed stochastic neighbor embedding
TN	True Negative
FP	False Positive
FN	False Negative
TP	True Positive

References

1. Tanmoy, D. Resistance Spot Welding: Principles and Its Applications. In *Engineering Principles*; Cooke, K.O., Cozza, R.C., Eds.; IntechOpen: Rijeka, Croatia, 2022; Chapter 4. [CrossRef]
2. Thornton, M.; Han, L.; Shergold, M. Progress in NDT of resistance spot welding of aluminium using ultrasonic C-scan. *Ndt E Int.* **2012**, *48*, 30–38. [CrossRef]
3. Zamanzad Gavidel, S.; Lu, S.; Rickli, J.L. Performance analysis and comparison of machine learning algorithms for predicting nugget width of resistance spot welding joints. *Int. J. Adv. Manuf. Technol.* **2019**, *105*, 3779–3796. [CrossRef]
4. Kim, K.Y.; Ahmed, F. Semantic weldability prediction with RSW quality dataset and knowledge construction. *Adv. Eng. Inform.* **2018**, *38*, 41–53. [CrossRef]
5. Zhou, B.; Pychynski, T.; Reischl, M.; Kharlamov, E.; Mikut, R. Machine learning with domain knowledge for predictive quality monitoring in resistance spot welding. *J. Intell. Manuf.* **2022**, *33*, 1139–1163. [CrossRef]
6. Dai, W.; Li, D.; Tang, D.; Jiang, Q.; Wang, D.; Wang, H.; Peng, Y. Deep learning assisted vision inspection of resistance spot welds. *J. Manuf. Process.* **2021**, *62*, 262–274. [CrossRef]
7. Deng, H.; Runger, G.; Tuv, E.; Vladimir, M. A time series forest for classification and feature extraction. *Inf. Sci.* **2013**, *239*, 142–153. [CrossRef]
8. Middlehurst, M.; Large, J.; Bagnall, A. The canonical interval forest (CIF) classifier for time series classification. In Proceedings of the 2020 IEEE International Conference on Big Data (Big Data), Atlanta, GA, USA, 10–13 December 2020; pp. 188–195. [CrossRef]
9. Middlehurst, M.; Large, J.; Flynn, M.; Lines, J.; Bostrom, A.; Bagnall, A. HIVE-COTE 2.0: A new meta ensemble for time series classification. *Mach. Learn.* **2021**, *110*, 3211–3243. [CrossRef]
10. Zhao, B.; Lu, H.; Chen, S.; Liu, J.; Wu, D. Convolutional neural networks for time series classification. *J. Syst. Eng. Electron.* **2017**, *28*, 162–169.
11. Wang, Z.; Yan, W.; Oates, T. Time series classification from scratch with deep neural networks: A strong baseline. In Proceedings of the 2017 International Joint Conference on Neural Networks (IJCNN), Anchorage, AK, USA, 14–19 May 2017; pp. 1578–1585.
12. Ismail Fawaz, H.; Lucas, B.; Forestier, G.; Pelletier, C.; Schmidt, D.F.; Weber, J.; Webb, G.I.; Idoumghar, L.; Muller, P.A.; Petitjean, F. Inceptiontime: Finding alexnet for time series classification. *Data Min. Knowl. Discov.* **2020**, *34*, 1936–1962. [CrossRef]
13. Williams, N.; Parker, J. Review of resistance spot welding of steel sheets Part 1 Modelling and control of weld nugget formation. *Int. Mater. Rev.* **2004**, *49*, 45–75. [CrossRef]
14. Summerville, C.; Adams, D.; Compston, P.; Doolan, M. Nugget Diameter in Resistance Spot Welding: A Comparison between a Dynamic Resistance Based Approach and Ultrasound C-scan. *Procedia Eng.* **2017**, *183*, 257–263. [CrossRef]
15. Zhang, H.; Wang, F.; Xi, T.; Zhao, J.; Wang, L.; Gao, W. A novel quality evaluation method for resistance spot welding based on the electrode displacement signal and the Chernoff faces technique. *Mech. Syst. Signal Process.* **2015**, *62–63*, 431–443. [CrossRef]
16. Sun, H.; Yang, J.; Wang, L. Resistance spot welding quality identification with particle swarm optimization and a kernel extreme learning machine model. *Int. J. Adv. Manuf. Technol.* **2016**, *91*, 1879–1887. [CrossRef]
17. Amiri, N.; Farrahi, G.; Kashyzadeh, K.R.; Chizari, M. Applications of ultrasonic testing and machine learning methods to predict the static & fatigue behavior of spot-welded joints. *J. Manuf. Process.* **2020**, *52*, 26–34. [CrossRef]
18. Hao, M.; Osman, K.; Boomer, D.; Newton, C. Developments in characterization of resistance spot welding of aluminum. *Weld.-J.-Incl. Weld. Res. Suppl.* **1996**, *75*, 1–4.
19. Hou, W.; Wei, Y.; Guo, J.; Jin, Y.; Zhu, C. Automatic Detection of Welding Defects using Deep Neural Network. *J. Phys. Conf. Ser.* **2018**, *933*, 012006. [CrossRef]
20. Martín, O.; Pereda, M.; Santos, J.I.; Galán, J.M. Assessment of resistance spot welding quality based on ultrasonic testing and tree-based techniques. *J. Mater. Process. Technol.* **2014**, *214*, 2478–2487. [CrossRef]
21. Óscar, M.; De Tiedra, P.; López, M. Artificial neural networks for pitting potential prediction of resistance spot welding joints of AISI 304 austenitic stainless steel. *Corros. Sci.* **2010**, *52*, 2397–2402. [CrossRef]

22. Ye, S.; Guo, Z.; Zheng, P.; Wang, L.; Lin, C. A Vision Inspection System for the Defects of Resistance Spot Welding Based on Neural Network. In *Computer Vision Systems*; Springer International Publishing: Berlin/Heidelberg, Germany, 2017; pp. 161–168. [CrossRef]
23. Zheng, T.; Yang, Y.; Zheng, P.; Benz, L.; Wang, L. An Appearance Inspection Method for Resistance Spot Welding Based on Semantic Segmentation. *IOP Conf. Ser. Mater. Sci. Eng.* **2020**, *790*, 012088. [CrossRef]
24. van der Maaten, L.; Hinton, G. Visualizing Data using t-SNE. *J. Mach. Learn. Res.* **2008**, *9*, 2579–2605.
25. Nandhitha, N. Artificial neural network based prediction techniques for torch current deviation to produce defect-free welds in GTAW Using IR thermography. In Proceedings of the 3rd International Conference on Advanced Computing, Networking and Informatics: ICACNI 2015, Orissa, India, 23–25 June 2015; Springer: Berlin/Heidelberg, Germany, 2016; Volume 1, pp. 137–142.
26. Wu, D.; Chen, H.; He, Y.; Song, S.; Lin, T.; Chen, S. A prediction model for keyhole geometry and acoustic signatures during variable polarity plasma arc welding based on extreme learning machine. *Sens. Rev.* **2016**, *36*, 257–266. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.