



Semi-supervised techniques to address the scarcity of experimental data: a case study of single point incremental forming

Jose Alberto Maestro-Prieto¹ · Pablo E. Romero² · José Miguel Ramírez-Sanz¹ · Andres Bustillo¹

Received: 13 February 2025 / Accepted: 2 October 2025
© The Author(s) 2025

Abstract

A lack of experimental data can be especially critical in new manufacturing processes. Although experimental datasets for industrial processes are reported in various research works, their lack of homogeneity complicates any fitting with conventional numerical models. Artificial Intelligence (AI) models can be an optimal alternative to extract useful information from those unconnected datasets, while generating models that can help explain the hidden patterns within datasets and interpret the predictions of the model for final users. Moreover, an AI algorithm that could be trained with limited labeled datasets would be in high demand, as it could effectively lower implementation costs. Semi-Supervised Learning (SSL) techniques might therefore be a promising solution to respond to industrial demand for the analysis of manufacturing processes. In this research, the use of SSL techniques is proposed in a case study of surface quality prediction in single point incremental forming, a promising new manufacturing technique. Datasets were extracted from the existing bibliography to generate a 234-instance dataset with 4 different industrial specifications of roughness. The best results were obtained using a semi-supervised Co-Training algorithm. Semi-supervised methods systematically improved the results obtained with the reference supervised methods, although statistical significance has not been mainly achieved due to the limited dataset size. The results obtained with the unbalanced dataset were very promising for its industrial implementation with an extended training dataset optimized for the range of process conditions of each end-user.

Keywords Single point incremental forming · SPIF · Surface roughness · Semi-supervised learning · Multiple data sources

Introduction

The remarkable increase in computing power and the use of Artificial Intelligence (AI), and especially Machine Learning (ML), in industry has led to several innovative improvements to industrial systems and machines. As recent reviews outline (Mokhtarzadeh et al., 2025; Malik et al., 2024; Jan et al., 2023; Chen et al., 2023; Lee & Lim, 2021; Peres et al., 2020), those improvements are mainly focused on energy optimization, predictive maintenance and quality control, enabling what is now referred to as Industry 4.0 and smart manufacturing.

However, insufficient data has been identified as one of the major limitations to the industrial implementation of AI models (Chen et al., 2023). This lack of data stems from two industrial conditions: (i) labeling manufacturing process data is costly and (ii) extending process conditions beyond optimal ones is economically unreasonable. These restrictions greatly hinder the optimal training of traditional ML

✉ Andres Bustillo
abustillo@ubu.es

Jose Alberto Maestro-Prieto
jamaestro@ubu.es

Pablo E. Romero
p62rocap@uco.es

José Miguel Ramírez-Sanz
jmrsanz@ubu.es

¹ Departamento de Ingeniería Informática, Universidad de Burgos, Avda. Cantabria, s/n, 09006 Burgos, Spain

² Department of Mechanical Engineering, University of Cordoba, Leonardo da Vinci Building, Campus of Rabanales, 14014 Córdoba, Spain

techniques, which require extensive labeled datasets that include a broad range of process conditions, reducing therefore the accuracy (and applicability) of ML models.

Two different approaches have been proposed to overcome the limitations of high-diverse labeled datasets. First, data augmentation methods can be used to extend existing industrial labeled datasets. One example is the well-known SMOTE technique. Second, ML techniques that can take advantage of partially labeled datasets can be an alternative solution (Jan et al., 2023), such as Semi-Supervised Learning (SSL) methods; in contrast to traditional ML techniques, such as supervised and unsupervised learning methods, which can extract very little information from this kind of datasets. This research focuses on the second approach to provide an industrial solution with low labeling costs.

Although semi-supervised techniques can address the challenge of partially labelled datasets, they cannot directly overcome the issue of limited diversity under manufacturing conditions. Academic research on manufacturing processes focuses on fitting analytical models to very specific conditions and developing benchmark datasets that are publicly available through established repositories. These datasets facilitate reproducibility and enable direct performance benchmarking of novel algorithms against existing ones. Unfortunately, AI models cannot be trained with these datasets to reliably solve industrial problems due to their limited size and diversity (Bustillo et al., 2022). Therefore, new strategies should be developed to leverage the potential of AI models in manufacturing processes under real industrial conditions.

The sheet metal manufacturing industry is one of the manufacturing industries with such limitations. Over the past six decades, it has undergone significant transformation in multiple areas (Modad et al., 2025). Among the most recent sheet metal forming techniques, incremental forming processes have become a promising rapid prototyping method due to their cost-effectiveness, which enables flexible production of small batches of sheet metal parts without the need for presses or dies. This process usually depends on a machining center, a widely used piece of equipment in manufacturing workshops (Jeswiet et al., 2005). A semi-spherical tipped punch guided by the machine's numerical control system follows a path along which it progressively deforms the sheet metal until the desired geometry is achieved (Dufflou et al., 2018). This novel sheet metal forming technology is used in the defense, aerospace, automotive, nuclear, medical, and architectural sectors (Afonso et al., 2018; Kumar et al., 2021).

Since the first publication addressing the application of AI techniques to incremental forming processes in 2008 (Han

et al., 2008), fewer than one hundred studies have been published in this area. Most of these studies focus on analyzing surface roughness, formability, geometrical accuracy, and forming forces (Harfoush et al., 2021). Surface roughness has been extensively studied, because it is particularly difficult to predict when relying on numerical models. The most frequently investigated process parameters include step depth, tool diameter, feed rate, blank thickness, and wall angle and Aluminium alloys are the most studied materials. Recent reviews of AI methods and techniques applied in incremental forming can be found in (He et al., 2025; Harfoush et al., 2021). Artificial Neural Networks (ANN) and Adaptive Neuro-Fuzzy Inference Systems (ANFIS) are the most prevalent ML approaches used (Nagargoje et al., 2023). However, the scope of the published studies to date remains limited, as the developed models are typically based on relatively small datasets. Therefore, there is a clear need for researchers to explore novel strategies to address this limitation and to develop more robust, generalizable models for incremental forming industrial applications.

This work uses a novel strategy based on: (i) using semi-supervised techniques and (ii) training them with a compilation of published, unconnected datasets and (iii) discretizing the process quality indicator in line with industrial demands. This strategy is finally validated to predict the surface quality of parts manufactured through Single Point Incremental Forming (SPIF). Firstly, the use of semi-supervised learning algorithms has not been previously reported to this manufacturing process. This approach is particularly relevant in industrial processes where economic and scheduling constraints, including limited human and material resources, slow metrological procedures, and strict customer deadlines, mean that only a subset of manufactured parts can be inspected for surface quality. Secondly, using a compilation of datasets extracted from different academic works for training purposes is supported in the capability of AI models to detect patterns and to overcome the effects of noise that differences between experimental data acquisition could produce. The considered unconnected datasets are typically very small (containing less than 15 instances each). The unified dataset has several advantages: (i) it contains a larger number of instances, close to 250, suitable for AI modelling purposes, (ii) parameters that are typically fixed in individual datasets become variables, enabling a more extensive modelling of the process, and (iii) the range of values for each variable is broader. These advantages allow to verify whether the authors' conclusions can be extrapolated to other operating windows beyond those defined in their specific studies. Thirdly, the discretization of the output variable, surface roughness, reformulates the problem as a

classification task. This classification approach aligns more closely with industrial practice than the traditional regression-based perspective because industry standards focus on classifying workpieces as acceptable or non-acceptable.

Finally, it is necessary to outline the main reasons for validating this strategy in an SPIF process. First, the process is relatively simple and well-defined, governed by a limited number of controllable parameters. Second, the process is executed on CNC machining centers, which introduce minimal uncontrolled variability and ensure reproducibility. Third, the experimental results obtained from this process have been modeled in the literature using statistical tools such as the Taguchi method, analysis of variance (ANOVA), or response surface methodology (RSM), but never SSL techniques.

The remainder of this paper is organized as follows. In Sect. “[State of the art](#)”, the basic background of incremental forming processes and its modelling by means of SSL approaches is presented. Then, in Sect. “[Methods](#)”, the main SSL key technologies and approaches are briefly presented. In Sect. “[Study case: SSL modeling of SPIF industrial process](#)”, the dataset preparation and the experimental design of the AI modeling process are firstly described, after which a comparative presentation of the models performance precedes a discussion of the results in light of the existing bibliography. Finally, the conclusions are presented in Sect. “[Conclusions and future work](#)”.

State of the art

This section is divided into two subsections. Firstly, the state of the art of SPIF process is introduced, highlighting the conditions and requirements for its successful implementation in industry. Secondly, once the state of the art of the manufacturing process is presented, it is time to introduce the state of the art of the machine learning techniques that will be mainly used to model this process, Semi-Supervised Learning, and their advantages and limitations for this task.

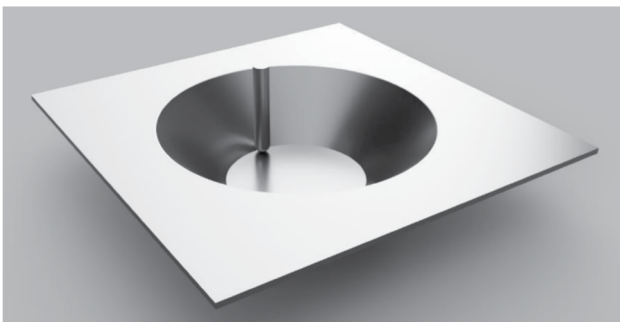


Fig. 1 Infographic related to the single point incremental forming (SPIF) process

Single point incremental forming (SPIF) performance and quality evaluation in industry

There are several incremental forming techniques (Popp et al., 2024), the most common of which is known as SPIF (Behera et al., 2017). The process is usually performed on a machining center (Afonso et al., 2019): an elevated frame is mounted on the machine table to which the sheet metal is secured for forming; a tool or punch with a semi-spherical tip is installed on the machine’s spindle; the numerical control of the machine guides the punch along a predetermined path; the punch progressively deforms the sheet metal, as Fig. 1 shows. More information on the process can be found in the following references (Martins et al., 2008; McAnulty et al., 2017; Rodriguez-Alabanda et al., 2019).

So far, researchers have been working to develop analytical and numerical models that relate the input variables of the process (step depth, tool diameter, feed rate, blank thickness, wall angle, spindle speed, lubricant, tool material, tool shape) to the output parameters (surface roughness, formability, springback/geometrical accuracy, forming force) (Nagargoje et al., 2023). At this point, it should be noted that aluminum alloy sheets and steel sheets are used more than any other in the literature (Azevedo et al., 2015).

One of the main drawbacks of SPIF, an extensively parameter studied in the literature (Nagargoje et al., 2023), is the high surface roughness it generates on manufactured parts. There are several methods to measure surface roughness, although the most widespread is based on the use of a profilometer. The device shown below in (Fig. 2) records surface irregularity with a stylus in contact with the material, although non-contact optical methods have now largely superseded that system. The arithmetic average Roughness (R_a) is a mean value obtained by integrating the surface profiles, as shown in expression (Eq. 1). High R_a values are associated with high roughness (poor surface finish), whereas low R_a values are linked to an excellent surface finish.

$$R_a = \frac{1}{l_m} \int_0^{l_m} |y| dx \quad (1)$$

Semi-supervised learning (SSL) and its application to SPIF

The main goal of ML techniques is to discover relationships and patterns in data that allow some kind of value-added task to be performed automatically, such as detecting

Fig. 2 View of the profilometer used for measuring surface roughness or Ra

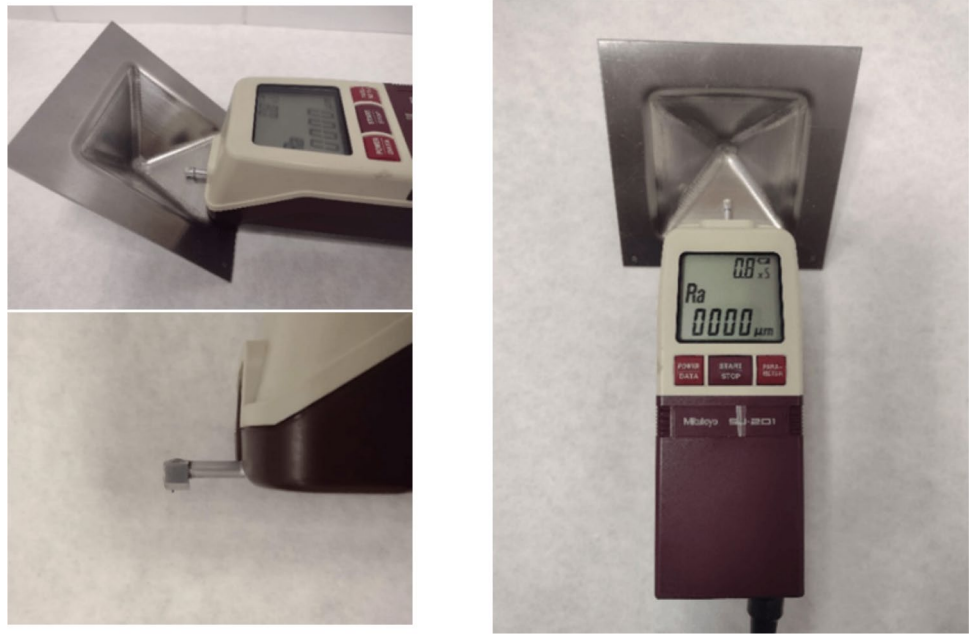
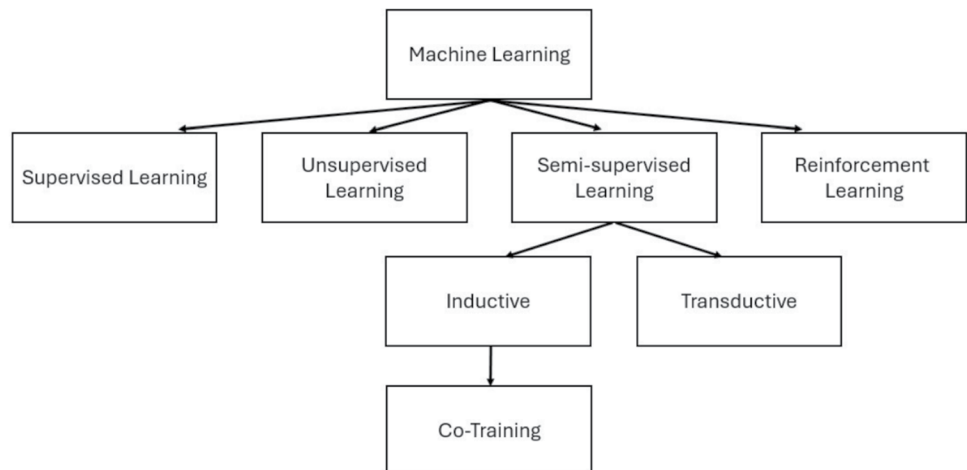


Fig. 3 Simple classification of ML approaches and methods. Co-training is classified as an SSL inductive method



normal-abnormal behavior, classifying a data instance, or process optimization. However, the starting point is the availability of data collected from the process that is to be automated. Today, many processes, including many industrial processes, collect data from machines almost by default, due to monitoring processes already in place, automated control systems, and other activities. The data are usually available in the form of registers, also called instances or observations, that are normally collected in a timely manner and contain values associated with various measured or calculated features. For machine learning purposes, available data are typically organized into datasets.

In (van Engelen & Hoos, 2020; Alloghani et al., 2020), ML approaches are classified into four main types, mainly depending on the requirements that the available data must

meet and how they are used: Supervised Learning (SL), Unsupervised Learning (UL), SSL, and Reinforcement Learning (RL). When using SL, the available data must already contain the ground truth feature(s) of interest for learning, such as a measure of tool wear or a classification for a quality measure. Providing these ground truth values is a process commonly referred to as “labeling”. The unlabeled data instances are noted as such: “unlabeled”. UL attempts to discover relationships between the available unlabeled data using only the relationships that can be hidden in the collected data without adding any other information. A model that has previously been trained can be refined with RL, depending on its performance. However, that refinement can only be achieved if reliable information on errors in the model predictions can be automatically obtained. A

perhaps oversimplified graphical representation of the classification of major ML approaches is shown in Fig. 3. An SSL Co-Training algorithm is included as an inductive ML method. For a more complete classification, see (van Engelen & Hoos, 2020).

The mandatory need for labels for each data instance is the main drawback commonly cited in the literature on SL. Labels are usually manually attached by an operator, who in some cases has to be an expert. It is usually time consuming and can be more expensive than data collection and model learning. Using UL is not a solution, because obviously less available information implies less achievable precision in the learned model, and usually fewer models/methods/approaches that can be used to exploit the available data. Thus, other approaches have emerged that try to mitigate these problems by mixing SL and UL approaches into one, such as Active Learning (AL) and SSL. AL tries to build a model with as little ground truth information as possible, thus reducing the time and cost of obtaining it. Some available data are sent to an external oracle—usually a human—to be labeled. A model is built and tested and, if the desired accuracy is not achieved, more “unlabeled” data instances are selected and sent to the oracle. This process is iterated until the trained model reaches the desired accuracy. The differences between AL approaches are basically related to the criterion for selecting the unlabeled instances to be labeled. However, this approach still requires the availability of the oracle. SSL attempts to use a dataset that usually contains few labeled data instances along with far more unlabeled data instances. The goal in SSL is to use the unlabeled instances to improve the accuracy of the model that could be obtained using only the available labeled instances. Section “[Data gathering and preparation](#)” briefly describes and classifies SSL methods.

Although it exists several SSL methods and techniques and the SSL approach for ML is obviously attractive, its practical use for model construction is, up to now, limited. Possibly, some of the known drawbacks of SSL methods limit its general use, especially the strong influence on the model’s accuracy of the subset of labeled instances (Ramírez-Sanz et al., 2023). In addition, mislabeled instances can lead to catastrophic results. Moreover, in general, there is no guarantee that a dataset that includes both labeled and unlabeled instances will generate a more accurate model than a model trained using only the subset of labeled instances. To overcome this drawback, there are some methods classified as Safe Semi-Supervised Learning (Ma et al., 2023; Li & Liang, 2019) that try to guarantee a higher accuracy of the SSL model obtained with respect to the model that

can be obtained by training only with the subset of labeled instances.

Despite the obvious improvement that the use of SSL can bring, especially for industrial applications, it is not easy to find examples of its use to solve real problems in the literature (Aggogeri et al., 2021). In industrial processes, the rare cases of use of these algorithms are mainly related to fault diagnosis, due to the industrial constraint to label working conditions in terms of tools or mechanical chains wear levels. For example, some proposals for using SSL techniques in fault diagnosis can be found in (Huang et al., 2023) for engine fault diagnosis, (Ma et al., 2022) for chemical processes fault diagnosis (using the Tennessee Eastman process testbench dataset), (Yuan et al., 2022) for rotor fault diagnosis, (Tao et al., 2021; Yu et al., 2021) for bearing fault diagnosis or (Maestro-Prieto et al., 2024) for wind turbine gearbox fault diagnosis.

Focusing on machining processes, such as forming, it should be noted that there are not many proposals for the use of SSL techniques. For example, recent reviews on surface roughness in incremental forming (Kumar, 2024; Murugesan et al., 2021) mainly focus on the relative importance of features and do not consider SSL approaches. Furthermore, specific reviews of AI methods used for incremental forming, such as (Yang et al., 2024; Nagargoje et al., 2023) mainly focus on supervised learning methods. The second one includes only one reference to a semi-supervised learning (clustering) solution. To the best of authors knowledge, this is the first attempt to use an SSL approach to surface roughness classification in incremental forming.

Methods

This section provides a brief introduction to general approaches to SSL. Then, the KEEL tool, which includes several SSL methods for training models, is described. Finally, the Co-Training SSL method, which obtains the best results for the dataset developed in this work, is described in more detail.

Taxonomy of SSL approaches

In (van Engelen & Hoos, 2020; Ramírez-Sanz et al., 2023) there are defined two main approaches to SSL: the transductive approach and the inductive approach. The transductive approach usually does not obtain a model. Instead, it looks for the best possible solution to a known set of instances. Since there is no proper model, this approach cannot be used to label new, unseen, unlabeled instances. The transductive

approach typically relies on graph-based methods, though there are also transductive versions of inductive methods, such as the Transductive Support Vector Machine (TSVM).

The inductive approach to SSL contains the most diverse set of methods for learning models. According to the taxonomy in (van Engelen & Hoos, 2020), these methods fall into three main groups: wrapper, unsupervised preprocessing, and intrinsically semi-supervised. Wrapper methods use one or more supervised methods to pseudo-label a few unlabeled instances in an iterative cycle of training and labeling. This group includes Self-training, Co-Training, and Boosting methods. Unsupervised preprocessing methods include SSL methods for a) feature extraction, b) labeling by clustering, and c) pretraining (i.e., first training a neural network using unlabeled instances that must subsequently be fine-tuned and retrained using labeled instances). Intrinsically semi-supervised methods include: a) Margin maximization approaches, such as Semi-Supervised Support Vector Machines (S3VMs); b) perturbation-based methods, which typically involve adding noise (i.e., Gaussian noise) to increase the robustness of the learned model; c) manifold approaches, which typically reduce dimensionality before constructing the model; and d) generative models, which typically rely on neural networks (e.g., generative adversarial networks (GANs)). For a more complete description of the various approaches and methods for SSL, see (van Engelen & Hoos, 2020).

The variety of existing approaches and methods that learn from both labeled and unlabeled instances demonstrates that none of them are universally applicable. Therefore, it is essential to carefully assess which approaches are effective for a particular problem. Additionally, some features may constrain the possible approaches or methods that can be used. Active learning would not be feasible if an oracle is not available on demand. If new, unseen instances arise, the transductive approach would not be applicable. The inductive approach may be more promising for constructing SSL solutions using datasets from industrial sources to solve typical industrial problems, such as monitoring, diagnosis, prognosis, and machine condition monitoring. For a recent review of SSL approaches and methods used in industry, see (Ramírez-Sanz et al., 2023).

Knowledge extraction based on evolutionary learning (KEEL)

A tool called KEEL (Knowledge Extraction based on Evolutionary Learning) was chosen for learning and testing the SSL models (Triguero et al., 2017). It is an open-source software tool programmed in Java, which includes different

kinds of algorithms including evolutionary and soft computing techniques for typical ML problems such as classification, regression, and association rules. KEEL is split into three main modules: an SL module, an SSL module, and a module specialized in methods for learning from unbalanced datasets. Some algorithms for data pre-processing are also included.

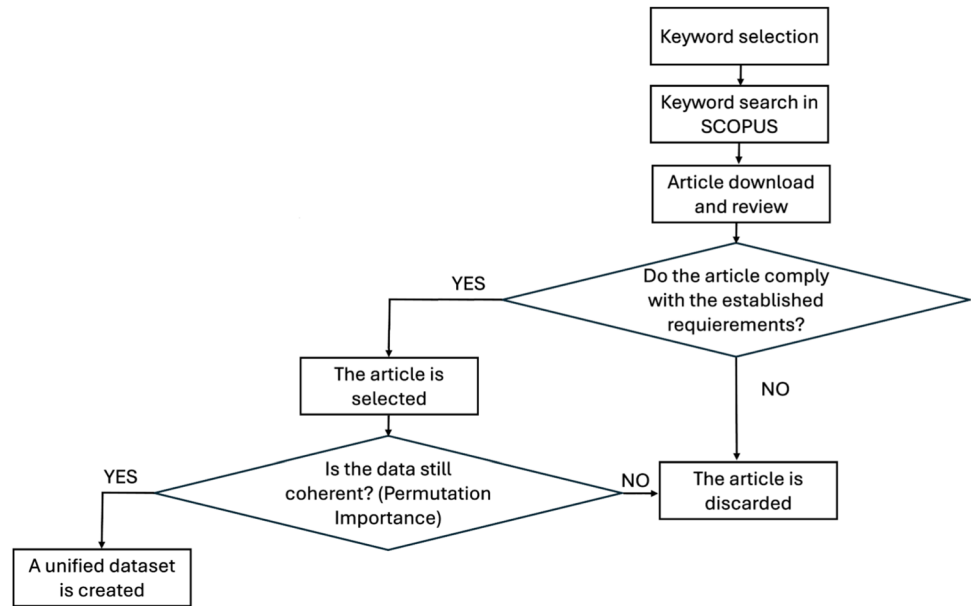
The SSL module implemented several methods such as Co-Forest (Li & Zhou, 2007), ADE-CoForest (Deng & Guo, 2011), Self-training (Yarowsky, 1995), Co-Training (Blum & Mitchell, 1998), RASCO (Wang et al., 2008), Rel-RASCO (Yaslan & Cataltepe, 2010), Democratic Co-learning (Zhou & Goldman, 2004), CoBC (Hady et al., 2010), CLCC (Huang et al., 2010), APSSC (Halder et al., 2013), SETRED (Li & Zhou, 2005), SNNRCE (Wang et al., 2010), and Tri-Training (Zhou & Li, 2005), various regression types (LDA, logistic, and others), and versions of basic supervised methods—the C45 implementation of a DT, Nearest Neighbor (NN), Naive Bayes (NB), and Support Vector Machine (SVM)—for use with semi-supervised datasets.

Some of the SSL algorithms, such as Co-Training, Tri-Training, RASCO, or REL-RASCO, use a learning base algorithm (C45, NN, NB, and SVM) or combination of base algorithms whose results are to be improved. Since the basic algorithms are selectable, many possible combinations of basic algorithms can be obtained for the same SSL method. A choice between Adaboost and Bagging learning algorithms, each of which requires a learning base algorithm, is also a feature of the CoBC algorithm.

In KEEL, the expected way of running an experiment involves a cross-validation process. Basically, two different experiments can be chosen: a 10-fold cross-validation and a 5x2-fold cross-validation. In 10-fold cross-validation, the dataset is randomly divided into 10 equal parts, and 10 experiments are performed: the model is trained using 9 of the 10 parts, and the remaining part is used for testing. This process is repeated 10 times, using each of the 10 parts for testing, and the results are averaged. When 5x2 cross-validation is selected, five different datasets are randomly generated, each containing typically 50% of the instances for training and the other 50% for testing. The results are averaged over the five datasets.

SSL co-training algorithm

Since there are many SSL algorithms and several of them are implemented in KEEL, only the algorithms that appear in the results are briefly described. Actually, all of them are Co-Training. For a better general overview of SSL

Fig. 4 Procedure followed to develop the unified dataset

algorithms, see (van Engelen & Hoos, 2020; Ramírez-Sanz et al., 2023) or the corresponding reference for each SSL method.

Co-Training (Blum & Mitchell, 1998) is a form of bootstrapping where two views (X_1 , X_2) of the same data are created by randomly splitting the feature set. Co-Training is an iterative process, where each iteration trains two equal or different learning algorithms (A_1 , A_2), each using the labeled instances of one view. The unlabeled instances are then predicted using both algorithms. At most, a usually small number, p , of unlabeled instances labeled with sufficient confidence by algorithm A_1 will be “pseudo-labeled” in view X_2 of algorithm A_2 . Similarly, algorithm A_2 will pseudo-label some unlabeled instances in view X_1 of algorithm A_1 . This process is repeated until a number of iterations k (a parameter of the algorithm) is reached. An underlying assumption of Co-Training is that the features of the dataset can be split into two independent subsets and that each subset still retains enough information to train a good model. In (Blum & Mitchell, 1998) it is claimed that using a subset of unlabeled instances, rather than the whole set of unlabeled instances, leads to a better result, the size of which is controlled by the parameter u .

The Co-Training algorithm in KEEL is a variation of the above-mentioned algorithm. Arbitrary values chosen for parameters p , k , and u , are permitted with KEEL, but some default values are provided. Perhaps the biggest difference in the KEEL implementation is that 3 classifiers have to be selected. Two algorithms are used as described above, but a third is used to calculate an accuracy metric at each iteration. Default values for Co-Training parameters in KEEL are: $p = 1$, $k = 40$ and $u = 75$ which are values close to those proposed in (Blum & Mitchell, 1998). In the

Table 1 Codification of the different alloys and bibliographical references

| Alloys | Code | References |
|---|------|---|
| AA1050-0, AA1100, AA1100-H0, AA1200-H14 | 1 | (Echrfif & Hrairi, 2014; Patel et al., 2015; Hamilton & Jeswiet, 2010; Bermudez et al., 2016; Desai et al., 2014) |
| AA2024-0 | 2 | (Kumar et al., 2018) |
| AA3105, AA3003, AA3003-H14 | 3 | (Taherkhani et al., 2019; Hagan & Jeswiet, 2004; Hamilton & Jeswiet, 2010) |
| AA5052, AA5052-H32 | 5 | (Mugendiran et al., 2014; Kumar et al., 2018) |
| AA6063 | 6 | (Gulati et al., 2016) |
| AA7075-T0, AA7075-0 | 7 | (Durante et al., 2009, 2011; Liu et al., 2014) |

experiments carried out, the default values for the parameters of the different algorithms tested were used.

Study case: SSL modeling of SPIF industrial process

This section is divided in seven subsections. The first subsection presents how the data have been gathered and prepared to define the original dataset, included in Appendix 1. The second one, presents the modeling settings selected to train and validate the machine learning algorithms. The following four sections present the modeling results, firstly from a general point of view and then focusing on the capabilities of Semi-Supervised algorithms, the comparison between different performance metrics and the evaluation of the sensitivity to tuning procedure of the Co-Training algorithm, the one that outperforms the other methods under

standard tuning conditions. A discussion of the modeling results closes this section.

Data gathering and preparation

The dataset used to train the models generated using semi-supervised techniques was built using data extracted from different studies published in indexed journals and available in the literature. Figure 4 shows the methodology followed to identify these articles, which are listed in Table 1. First, the keywords of interest for this study were defined: ‘aluminum’, ‘single point incremental forming’, and ‘surface roughness’. A systematic search was conducted in the SCOPUS database, targeting the title, abstract, and keyword fields. This search resulted in a total of 86 publications. All retrieved articles were downloaded and thoroughly reviewed to identify studies employing comparable methodologies, thereby ensuring the coherence of the dataset. Studies that incorporated technological innovations falling outside the standard framework defined in the present work were excluded. For instance, some authors explored strategies to enhance the formability of the material through the application of heat, which was not considered within the scope of this study. Finally, a total of 13 articles were selected (Table 1).

The studies used to build the unified dataset, published between 2008 and 2019, typically employed classical statistical methods such as Taguchi, Response Surface Methodology (RSM), or ANOVA, implemented using commercial software such as Statgraphics or Minitab, to analyze the results obtained from experimental trials. These methods enable identification of the most influential parameters in the output variable (in this case, surface roughness) and allow prediction of its numerical value. The variables investigated include (Table 2): spindle speed, feed rate, step sizes, tool diameters, wall angles. However, each study typically defines the aluminum alloy to be formed, the sheet thickness employed, the geometry to be produced, and the lubricant applied. The aluminum alloys listed in Table 2 were codified and simplified. The characteristics (hardness, density, strength, appearance ...) of each alloy were assumed to prevail with respect to the alloy components. The alloys associated with the instances were coded in such a way that six

Table 2 Parameters included in the unified dataset (see Appendix 1) and the values they assume

| Parameter | Values | Total |
|---------------------|---|-------------------|
| Aluminum alloys | AA5052, AA1050-0, AA7075-0, AA1100, AA3105, AA3003-H14, AA2024-0, AA6063, AA1200-H14, AA5052-H32 | 10 alloys |
| Thickness (mm) | 0.5, 0.55, 0.8, 0.81, 0.9, 0.91, 1, 1.02, 1.09, 1.2, 1.5, 1.6, 1.67, 2.54 | 14 thickness |
| Spindle-speed (rpm) | 0, 96, 200, 250, 400, 500, 600, 1000, 1200, 1250, 1500, 1750, 2000, 2300 | 14 spindle-speeds |
| Feed-rate (mm/min) | 0, 25, 50, 317.5, 500, 600, 635, 800, 1000, 1100, 1270, 1400, 1500, 2000, 2500, 3500, 4000, 5000, 5080, 6000, 6985, 8890 | 22 feed-rates |
| Step-size (mm) | 0.004, 0.01, 0.051, 0.10, 0.13, 0.20, 0.25, 0.30, 0.381, 0.40, 0.50, 0.51, 0.60, 0.75, 0.76, 0.80, 1.00, 1.20, 1.25, 1.30, 1.50 | 21 step-sizes |
| Tool diameter (mm) | 5, 6, 7.5, 8, 9, 10, 11.6, 12, 12.7, 13, 14, 15, 15.6, 20, 25, 30 | 17 tool diameters |
| Wall angle (°) | 29.74, 30, 34, 40, 45, 46, 47, 48, 50, 52, 55, 60, 62, 63, 63.4, 64, 65, 67, 69, 70, 72, 73, 76, 80, 82, 84 | 26 wall angles |
| Geometry | Cone, pyramid, cone-pyramid, oval | 4 geometries |
| Lubrication | Dry, mineral oil, water-oil emulsion, grease | 4 lubrications |

main alloys were taken into account, as shown in Table 1. Figure 5 summarizes the main process parameters. That variability resulted in a rich and varied dataset, containing (Table 2): 10 different aluminum alloys, 14 thicknesses, 14 spindle speeds, 22 feed rates, 21 step sizes, 17 tool diameters, 26 wall angles, 4 geometries, and 4 different lubrication methods.

The considered works contain typically small datasets, with 9 or 15 instances (a number that coincides with standard Taguchi or Box-Behnken designs). This limited number of instances makes the application of artificial intelligence techniques unfeasible. However, the unified dataset presents several advantages: (i) it contains a larger number of instances, close to 250, which could be considered ‘small-data’ in the field; (ii) values that are typically fixed in individual datasets become variables, enabling a deeper understanding of the process; (iii) the range of values for

Fig. 5 Graphical description of the different parameters of the forming process

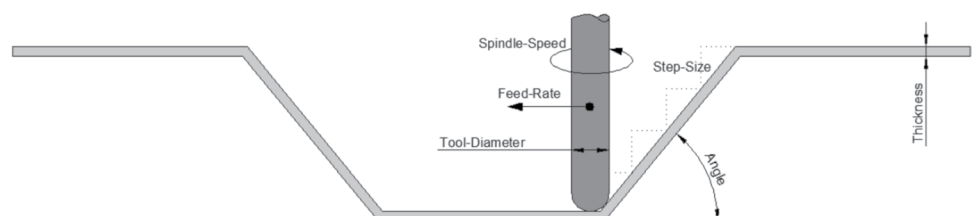


Table 3 Surface roughness intervals assigned to each class of surface finish, as per ISO 1302

| Surface roughness class | Lower end (μm) | Upper end (μm) |
|-------------------------|-----------------------------|-----------------------------|
| N1 | 0.000 | 0.025 |
| N2 | 0.026 | 0.050 |
| N3 | 0.051 | 0.100 |
| N4 | 0.101 | 0.200 |
| N5 | 0.201 | 0.400 |
| N6 | 0.401 | 0.800 |
| N7 | 0.801 | 1.600 |
| N8 | 1.601 | 3.200 |
| N9 | 3.201 | 6.300 |
| N10 | 6.301 | 12.500 |
| N11 | 12.501 | 25.000 |
| N12 | 25.001 | 50.000 |

each parameter is much broader, allowing verification of whether the authors' conclusions can be extrapolated to other operating windows beyond those defined in their specific studies. For example, the step size ranges from 0.01 to 1.50 mm, with intervals equal to or smaller than 0.05 mm.

To obtain the final dataset some minor changes have been done in the data. Firstly, attributes *geometry*, *lubrication* and *alloy*, all of them with more than 2 possible literal values, were split in binary attributes (e.g. *Geometry* could take 3 values: Pyramid, Cone or Oval in the original dataset was split in *Geometry_Pyramid*, *Geometry_Cone* and *Geometry_Oval* binary attributes). Secondly, the output variable is discretized into categories, thereby reformulating the problem as a classification task. This classification approach is more aligned with industrial practice than the traditional regression-based perspective because it facilitates the discrimination between acceptable and non-acceptable workpiece depending on the roughness class. ISO 1302 specifications were followed for this discretization. This norm defines several levels of surface finish, ranging from N1 to N12, considering different industrial needs. Table 3 summarizes the surface ranges defined for the first 12 levels: N1 represents the best surface finish ($0.000 < Ra < 0.025$

μm) while N12 represents the worst surface finish ($25.001 < Ra < 50.000 \mu\text{m}$). Figure 6 illustrates graphically the various roughness finishes for classes N5 to N12. Unfortunately, these advantages from an industrial point of view will also make it impossible to compare the obtained modeling results with the original references, as the new modeling task is, instead a regression between close operating conditions, a classification task with a very diverse dataset in terms of operating conditions.

Finally, to assure the coherence between the datasets, a new attribute called *Machine* was added to the dataset, which identifies the origin of the data (from 1 to 13 to identify the original manuscript where it has been extracted) and the feature importance ranking was performed. The feature importance calculated using permutation importance is shown in Fig. 7 using a Box Plot diagram. Permutation-based importance is computed by randomly shuffling the values of each feature and evaluating the model's performance. This performance is assessed using an estimator—in this case, a Random Forest (Breiman, 2001) with the default settings in scikit-learn. Moreover, different metrics, or scoring methods, can be used to evaluate performance; in this case, accuracy was chosen. To obtain more robust results, the permutation process can be repeated multiple times; here, 10 repetitions were performed. In this figure, the *Machine* attribute ranking was 5th. Besides, almost all the attributes that have a lower ranking compared with *Machine* are the binary attributes that were generated from the 3 attributes *Geometry*, *Lubrication* and *Alloy* which were split in many different binary attributes; therefore, their influence is expected to be lower. Therefore, it can be concluded that the influence of the attribute *Machine* is very low and that the origin of the dataset plays no role in the information extraction procedure of the machine learning algorithms. This finding supports the coherence of the compiled dataset and its suitability for the intended machine learning tasks.

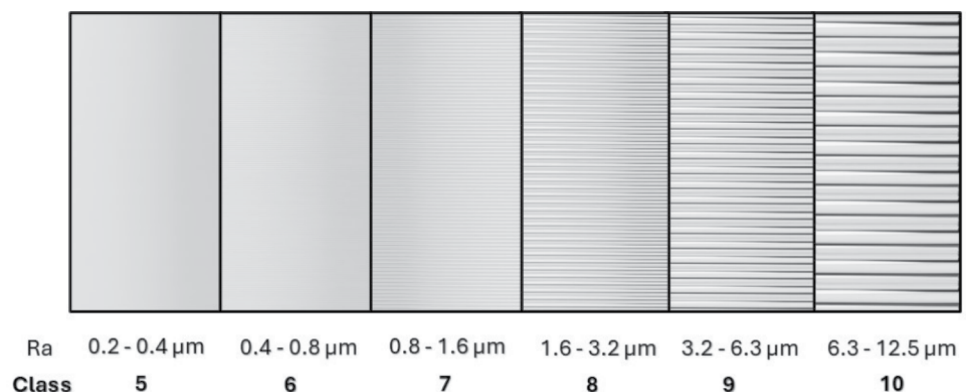
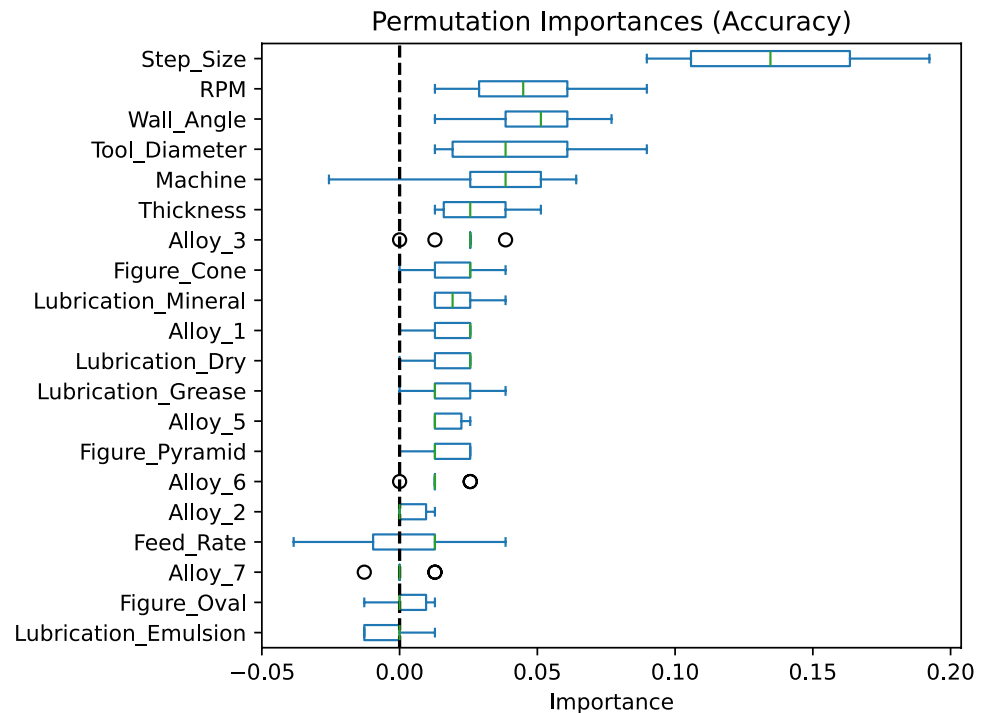
Fig. 6 Graphical images with different roughness finishes, ranging from N5 to N12

Fig. 7 Permutation importance for calculating the features importance



AI-based modeling settings

Two central decisions must be taken in the design of an SSL experiment: both a validation strategy and a method to unlabeled instances must be chosen. There are many strategies for validating the experiment, but they all basically consist of dividing the dataset into two subsets, one for training the model and the other for evaluating (testing) the performance of the model. The model is tested on a subset of instances other than the one used for training to evaluate the error prediction for new, unseen instances, so as to select a better model. Evaluating the true error is very important, because the model may over-fit the instances used for training, in which case the true error may be too high to be a useful model. Since this is a semi-supervised learning problem, only some instances need to keep their labels, and the others need to lose their labels. For most algorithms, it is important that the labeled subset of instances remains representative and contains examples of all possible labels. Otherwise, normally unknown labels will not be recognized when they appear and will be misclassified.

Different metrics can be used to measure the results obtained. Specific components of the calculation can be checked with particular metrics. For example, the accuracy metric calculates the score by giving equal weight to each result for each instance. Sometimes the overall result is not as important as focusing on the correctly or incorrectly classified instances. Hence, the definition of different metrics.

The accuracy metric is a standard metric that is focused on correctly classifying each instance. The recall metric is focused on the correctly classified instances for each class among all the instances classified in that class. Since this is a multi-class problem, the calculations should be performed for each class. Figure 8 illustrates the overall process.

One of two typical validation strategies can be selected in KEEL: 10-fold cross validation and 5x2-fold cross validation. 10-fold cross validation consists of randomly dividing the dataset into 10 folds with the same number of instances. 10 different runs are defined, each of which selects a different fold for testing, while the other 9 are used for training. After the 10 runs, a final score is computed as the average of the results. 5x2-fold cross-validation consists of randomly dividing the dataset into two equal parts, one for training and one for testing. Since this can rely too much on the random selection of instances, the process is repeated 5 times, and the final result is the average of the 5 runs. Actually, in an SL setting, two different experiments are run each time: the first half is used for training and the other half for testing, and then *vice versa*, so that two results are generated in each run. In KEEL, 5x2-fold cross-validation only runs one of the experiments, since the training half is partially unlabeled and the testing half is fully labeled.

A total of 243 instances were collected from the references included in Table 1. The number of instances of this initial dataset constructed from the available data is shown in Table 4. As can be seen, this dataset is strongly unbalanced:

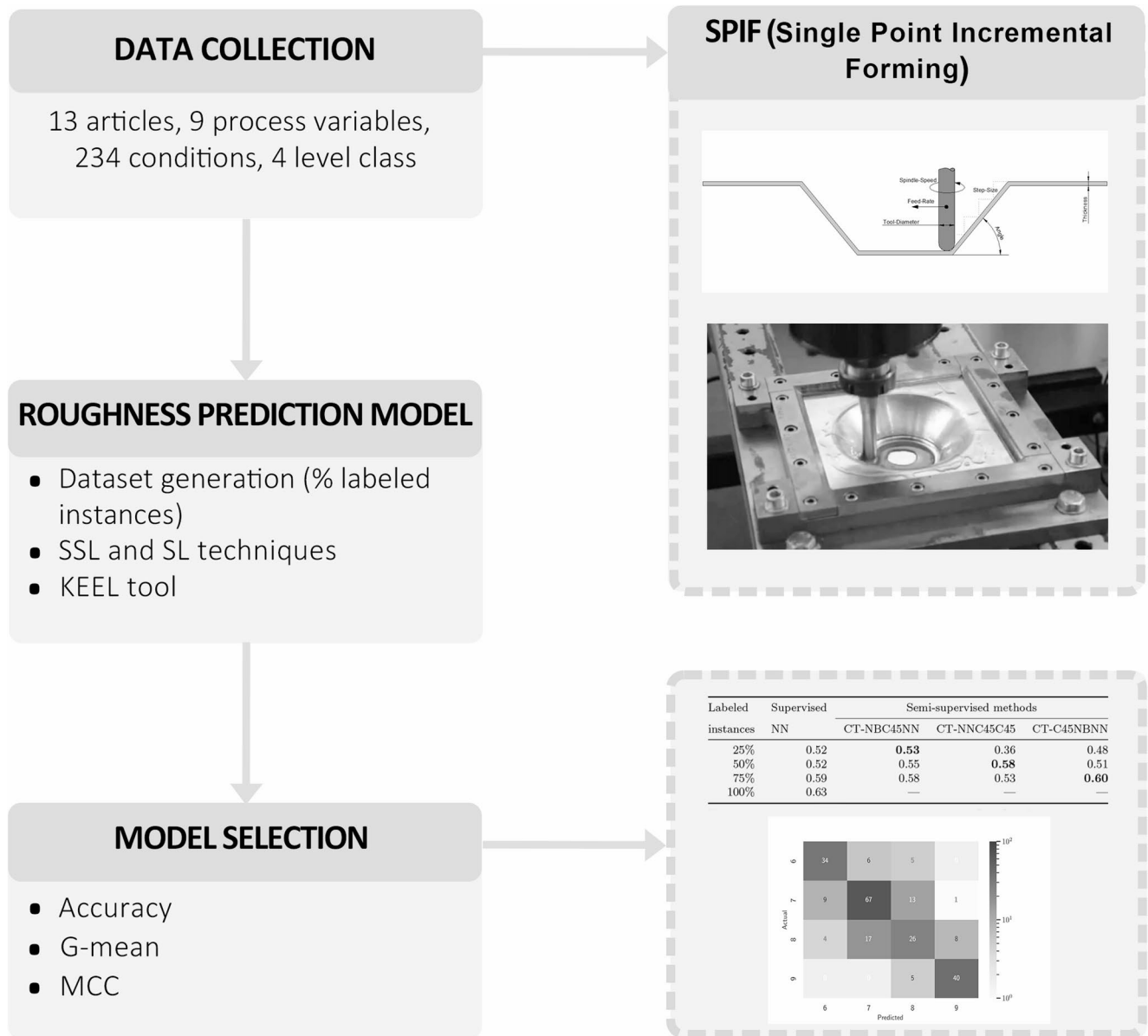


Fig. 8 Graphical description of the entire process

Table 4 Distribution of instances for each class in the initial dataset

| Class 5 | Class 6 | Class 7 | Class 8 | Class 9 | Class 10 | Total |
|---------|---------|---------|---------|---------|----------|-------|
| 7 | 43 | 91 | 57 | 43 | 2 | 243 |

class 5 and class 10 are clearly underrepresented. Class 5 represents 2.88% of the dataset and class 10 represents 0.82% of the dataset. Thus, classes 6-9 represent 96.30% of the dataset. As the industrial interest in classes 5 and 10 may be lower with respect to the others (Class 5, because it represents a very high quality that could not be detected as a process failure, and Class 10, because such an obvious failure in the manufacturing process is very uncommon in

industrial processes under very controlled conditions), and due to the low number of available instances, classes 5 and 10 were removed from the dataset.

A 5 × 2-fold cross-validation set of files was generated using Python to run the experiments. However, an 80-20% split was performed instead of the usual 50-50%, due to the limited number of instances available, so the full set of 234

Table 5 Percentage of labeled instances, number of instances for training, unlabeled and labeled instances for each training dataset and the distribution of labeled instances for each class

| % of labeled instances | Inst. for training | N. of unlabeled | N. of labeled | Class | Class | Class | Class |
|------------------------|--------------------|-----------------|---------------|-------|-------|-------|-------|
| | | | | 6 | 7 | 8 | 9 |
| 25% | 187 | 140 | 47 | 9 | 18 | 12 | 8 |
| 50% | 187 | 93 | 94 | 17 | 37 | 23 | 17 |
| 75% | 187 | 46 | 141 | 26 | 55 | 35 | 25 |

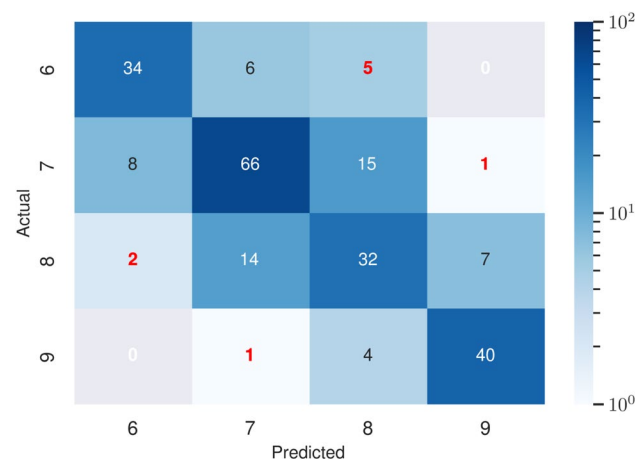
Table 6 Percentage of labeled instances, number of instances for the test dataset and the distribution of instances for each class

| % of labeled instances | Inst. for test | Class | Class | Class | Class |
|------------------------|----------------|-------|-------|-------|-------|
| | | 6 | 7 | 8 | 9 |
| 100% | 47 | 9 | 18 | 11 | 9 |

Table 7 Percentile accuracy of the different SL algorithms

| Labeled instances (%) | Supervised methods | | | |
|-----------------------|--------------------|---------|--------|--------------|
| | SMO (%) | C45 (%) | NB (%) | NN |
| 25 | 57.87 | 60.85 | 59.57 | 64.68 |
| 50 | 56.60 | 62.98 | 59.57 | 65.11 |
| 75 | 59.57 | 68.51 | 62.98 | 70.21 |
| 100 | 60.85 | 71.06 | 62.13 | 73.19 |

Each row represents the results, using a dataset with a different percentage of labeled instances, ranging from 25 to 100%. Numbers that appear in bold type are the best result in each row

**Fig. 9** Confusion matrix for the NN algorithm trained using the dataset containing the 100% labeled instances. The number of misclassifications by more than one class is shown in bold type and in a red font

instances was split into two subsets of 187 and 47 instances, respectively. The files with 80% of the instances were used for training and the files with 20% of the instances were used for testing. The original file was randomly split into two, five different times. Once the different files had been created, an unlabeled process was performed and 3 datasets were created, keeping 25%, 50%, and 75% of the original labeled instances. A fourth dataset containing 100% labeled instances was kept for use with the SL algorithms for comparative purposes. It was decided to perform an incremental

unlabeling process, so that the labeled instances in the 25% labeled files would also appear in the 50% and the 75% labeled files, and so on. During the unlabeled process, it was checked that at least one labeled instance for each class was included in the files. The number of instances, labeled and unlabeled, for each training dataset is shown in Table 5.

The percentages, 25, 50, and 75%, were chosen in accordance with the number of instances for each class in the original dataset. The number of instances for the test dataset is shown in Table 6. The number of instances *per* class follows approximately the same proportions in the training and test files. As can be seen in Tables 5 and 6, classes 6 and 9 are underrepresented compared to classes 8 and especially 7.

Results of the supervised algorithms

KEEL has some basic SL algorithms that have been adapted to use the same semi-supervised dataset as the SSL algorithms for comparative purposes. Those basic algorithms are a C45 DT classifier, a Nearest Neighbor (NN) classifier, a Naive Bayes (NB) classifier, and a Support Vector Machine (SVM), although it is preferred to use the acronym of the KEEL learning algorithm, SMO (Sequential Minimal Optimization), to refer to the SVM.

In comparison, Table 7 shows the results of the accuracy metric using the SL algorithms when used with the SSL datasets and the 100% labeled dataset.

As can be seen in Table 7, more labeled instances in the dataset in general implies higher accuracy of any AI model, as may be expected. The highest accuracy was obtained by the NN classification algorithm. It achieved an accuracy of 73.19% when trained using the 100% labeled instances, as was also expected. The C45 algorithm also achieved close accuracy.

Figure 9 shows the confusion matrix for the results of the NN algorithm trained on the dataset with 100% labeled instances. The main diagonal shows the number of correctly classified instances. As can be seen, the misclassified instances are mostly classified in the previous or next class. Misclassifications of that sort are a major industrial requirement, because critical quality failures, rather than between consecutive roughness levels, occur when the real level is at some distance from the predicted level. Only very few

Table 8 Percentile accuracy metric scores of the different algorithms

| Labeled instances (%) | SL | Semi-supervised methods | | | | | | |
|-----------------------|--------|-------------------------|----------------|--------------|-----------------|----------|----------------|----------|
| | NN (%) | σ (%) | CT-NBC45NN (%) | σ (%) | CT-NNC45C45 (%) | σ | CT-C45NBNN (%) | σ |
| 25 | 64.7 | 7.8 | 65.5 | 3.1 | 53.2 | 3.6 | 60.9 | 4.9 |
| 50 | 65.1 | 6.0 | 66.8 | 3.1 | 69.8 | 4.9 | 64.3 | 2.6 |
| 75 | 70.2 | 5.6 | 69.4 | 3.5 | 66.4 | 1.6 | 71.1 | 5.1 |
| 100 | 73.2 | 4.9 | – | – | – | – | – | – |

Each row represents the results obtained by training the algorithm on a dataset with a different percentage of labeled instances, ranging from 25 to 100%. The first column contains the percentage of labeled instances in the dataset, the second column contains the results of the SL algorithm for comparison, and the remaining columns are the results for the different SSL algorithms. Numbers that appear in bold type are the best result for each row. The standard deviation (σ) of the 5 repetitions to calculate the averaged model's accuracy is also included

Table 9 Results obtained from the hypotheses test

| Labeled instances (%) | SL | Semi-supervised methods | | | | | |
|-----------------------|--------|-------------------------|----------|-----------------|------------|----------------|------------|
| | NN (%) | CT-NBC45NN (%) | p-value | CT-NNC45C45 (%) | p-value | CT-C45NBNN (%) | p-value |
| 25 | 64.7 | 65.5 | 0.296499 | 53.2 | 0.000843** | 60.85 | 0.013768** |
| 50 | 65.1 | 66.8 | 0.190376 | 69.8 | 0.015273* | 64.26 | 0.248098 |
| 75 | 70.2 | 69.4 | 0.109903 | 66.4 | 0.013768** | 71.06 | 0.081212 |

instances were misclassified by more than one class in the NN algorithm (9 instances, less than 5% of the dataset).

KEEL implements both SL and SSL algorithms for classification. Unfortunately, it is not possible to use these SL algorithms with SSL-ready datasets in KEEL, so testing with them is not straightforward. However, it is worth using the basic C45, NN, SVM (SMO) or NB algorithms to compare results, as these are widespread and perform acceptably on many datasets. Many SSL algorithms also use one or more of these basic algorithms as a base classifier. Therefore, it is possible to determine whether the SSL approach improves upon these base algorithms. Additionally, the datasets' reduced size makes them incompatible with almost any neural networks or deep learning techniques.

Accuracy of the semi-supervised algorithms

Many KEEL algorithms require the selection of basic classifiers when an experiment is defined. For example, the Co-Training algorithm requires the specification of three basic classifiers (C45, NN, NB, SMO), two of which are used by the algorithm itself and a third of which is used to obtain the result. Co-Training is not the only algorithm with this feature. Tri-Training, DE-Tritraining, RASCO and REL-RASCO also require the specification of one or more basic classifiers. CoBC requires the user to specify whether it uses bagging or boosting, as well as a basic classifier. Therefore, we tested approximately 400 algorithms and combinations of algorithms for each dataset. The default values for the other parameters of the algorithms proposed by KEEL were kept fixed and fine-tuning was not performed to avoid very long computing times, especially because

the implementation of the SMO algorithm in KEEL is very slow. The Accuracy score has been chosen as the first option for ranking the results obtained.

Equation 2 shows the formula for computing the (micro-averaged) Accuracy score for a multiclass problem. For a problem of K classes, C_{kk} is the value in the row k column k in the confusion matrix. N is the number of instances.

$$Accuracy = \frac{1}{N} \sum_{k=1}^K C_{kk} \quad (2)$$

Table 8 shows the best accuracy metric results obtained for the SSL algorithms. For each dataset, defined as percentages ranging from 25 to 75% of labeled instances in the training files, the accuracy metric of the SL NN algorithm used as the supervised reference is given. It also includes the SSL algorithms that obtained the best accuracy metric result, along with the standard deviation for each result, as the accuracy results are the average of five repetitions. The accuracy's standard deviation (in the range 1.6–7.8%) is rather significant, due to the strong noise that the random instance selection process introduces in the unlabeling process; nevertheless, its range doesn't compromise the stability of the proposed methodology. For each dataset, the same Co-Training algorithm, but with different basic classification algorithms, obtained the best accuracy result. So, depending on the available dataset, the best result was obtained with the Co-Training algorithm with a specific combination of basic algorithms. The best SSL result improved the SL result in each row. The best SSL result for the accuracy metric was 71.06%, not even 1% more than

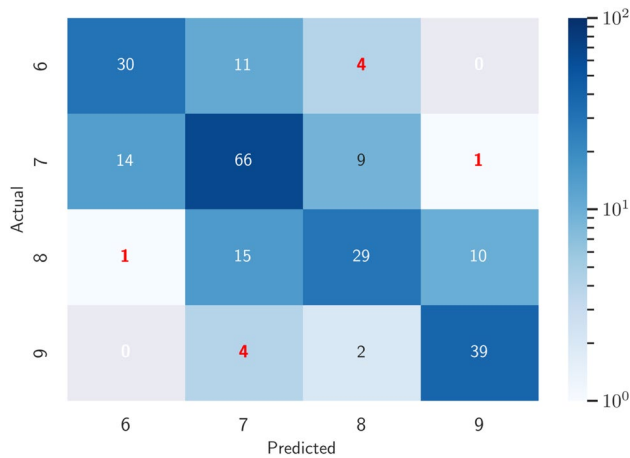


Fig. 10 Confusion matrix for the SSL Co-training NN-C45-C45 algorithm trained using the dataset containing the 50% labeled instances. The number of misclassifications by more than one class is shown in bold type and in a red font

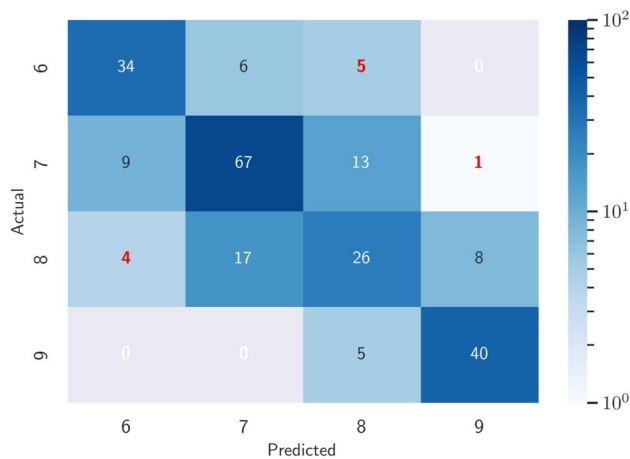


Fig. 11 Confusion matrix for the SSL Co-training C45-NB-NN algorithm trained using the dataset containing the 75% labeled instances. The number of misclassifications by more than one class is shown in bold type and in a red font

the result obtained for the NN algorithm. The largest difference in accuracy scores can be found for the dataset containing 50% of labeled instances, where the SL NN algorithm

obtained an accuracy level of 65.11% and the Co-Training (NN-C45-C45), an accuracy level of 69.79%.

To assess whether there is a statistically significant difference between the best result obtained using the SSL algorithm and the reference result obtained using the SL NN algorithm, a hypothesis test is performed. The basic data and p-values are shown in Table 9. Bold values identifies the most accurate models for each dataset (percentage of labeled instances). The null hypothesis (H_0) is that the two scores are equal. The `kfold_ttest()` method from the `Correctipy` Python package was used to perform the hypothesis test.

A two-tailed t-test was performed to assess whether a statistically significant difference emerged between the results. As can be seen, most of the p-values indicate that there is no significant difference between the results (they are greater than 0.025). Only those cases where the difference is greater than 10% of the averaged accuracy of the NN model (around 5% of the accuracy value) achieve a statistically significant difference with respect to the corresponding NN algorithm. Most of these cases refer to conditions where SSL algorithms perform worse than the NN algorithm and only in the dataset 50% dataset is the best SSL technique able to achieve a positive statistical difference with the NN algorithm. In the rest of the cases, while some SSL algorithms have been shown to outperform the NN algorithm, the difference is insufficient to be considered statistically significant. It can also be seen from Table 9 that using unlabeled instances does not necessarily improve the results. Furthermore, the result obtained using an SSL algorithm is statistically significantly worse than the SL benchmark. This is evident in the case of the CT-NNC45C45 Co-Training algorithm using the 25% labeled dataset, despite the fact that the same algorithm is the best option for another dataset (the 50% labeled dataset).

Figures 10 and 11 show the confusion matrix for the Co-Training NN-C45-C45 algorithm, trained on the dataset with 50% labeled instances, and the Co-Training C45-NB-NN algorithm, trained on the dataset with 75% labeled instances, respectively. It is worth noting that in both figures only 10 instances (< 5% of the test dataset) were misclassified by

Table 10 The accuracy of the different algorithms is shown as a percentage

| Labeled instances | Semi-supervised methods | | | | | | | | | |
|-------------------|-------------------------|--------------|----------|--------------|----------|--------------|---------|--------------|--------|--------|
| | STRD (%) | σ (%) | CA-N (%) | σ (%) | CB-C (%) | σ (%) | DEM (%) | σ (%) | ST (%) | ϕ |
| 25% | 61.3 | 5.2 | 59.2 | 8.7 | 55.7 | 7.0 | 64.7 | 4.8 | 60.4 | 6.3 |
| 50% | 63.8 | 4.1 | 66.0 | 4.6 | 62.6 | 4.6 | 66.4 | 8.7 | 66.0 | 4.8 |
| 75% | 69.4 | 3.9 | 66.8 | 5.4 | 65.0 | 4.3 | 67.2 | 6.5 | 70.6 | 3.9 |

Each row shows the results of the accuracy metric and standard deviation using a dataset with a percentage of labeled instances ranging from 25 to 75%. The column headings are as follows: STRD stands for SETRED; CA-N stands for CoBC-AdaBoost-NN; CB-C stands for CoBC-Bagging-C45; DEM stands for Democratic CoLearning; and ST stands for SelfTraining

more than one class. That result was similar to the result obtained with the SL reference NN algorithm trained on 100% labeled dataset, which misclassified 9 instances by more than 1 class (see Fig. 9). It is an important industrial result, as the model obtained using the SSL techniques in no way reduced the reliability of the classification system. Reliability under industrial conditions is mainly related to strong prediction failures: a workpiece of average surface quality, with a deviation of perhaps 1 level in the Norm between the real and the predicted surface quality would not be critical; but a failure of 2 levels would mean that the system could predict a proper workpiece, while an inferior quality piece might be sent to the client, which would clearly be an inadmissible situation.

It is also worth noting that in Fig. 11, compared to Fig. 10, class 6 improves its result and class 8 has the worst result. In fact, Class 6 is the class that improves its classification accuracy the most and Class 8 is the only one that worsens its classification accuracy. It means that the number of instances available in the dataset is a critical key to obtain a fine-tuned and accurate classifier. A final industrial implementation of that solution in a workshop might therefore require a sufficient number of instances to train an accurate and reliable classifier, so that higher accuracy could be achieved for each class and misclassifications by more than one class could be completely avoided.

While the best results were achieved using specific combinations of basic classifiers in Co-Training methods, many other methods and combinations of basic algorithms were tested and discarded. Some of these are shown for comparative purpose in Table 10. Table 10 includes the results obtained using the SETRED algorithm, which is considered a safe SSL method. The table also includes two ensemble approaches: one CoBC method using Adaboost boosting and one using bagging. In both cases, it is necessary to select a basic algorithm. In general, ensemble methods are considered to be more accurate than other methods. The

Table 11 The scores of the MCC metric for the different algorithms

| Labeled instances (%) | Super-vised NN | Semi-supervised methods | | |
|-----------------------|----------------|-------------------------|-------------|-------------|
| | | CT-NBC45NN | CT-NNC45C45 | CT-C45NBNN |
| 25 | 0.52 | 0.53 | 0.36 | 0.48 |
| 50 | 0.52 | 0.55 | 0.58 | 0.51 |
| 75 | 0.59 | 0.58 | 0.53 | 0.60 |
| 100 | 0.63 | – | – | – |

Each row represents the results obtained by training the algorithm on a dataset with a different percentage of labeled instances, ranging from 25 to 100%. The first column contains the percentage of labeled instances in the dataset, the second column contains the results of the SL algorithm for comparison, and the remaining columns are the results of the different SSL algorithms. Numbers that appear in bold type are the best result in each row

Democratic Co-Learning method, which is an ensemble method using several algorithms instead of several views of the same dataset in an attempt to reduce learning bias, has also been included. The self-training method has also been included. This is one of the first SSL methods, as well as being one of the simplest approaches. It uses a basic C45 algorithm to pseudo-label unlabeled instances in a loop. First, a C45 model is trained, and then this model is used for pseudo-labeling. However, none of these methods outperform Co-Training methods when using our datasets.

Other metrics

The accuracy metric is the most commonly used to compare the results of different algorithms. It yields an overall performance of the different algorithms when faced with the classification task. However, that metric may not be the best option to obtain a precise idea of algorithm performance, if special attention has to be paid to classification errors or the unbalanced nature of the dataset. In (Luque et al., 2019), both the use the Geometric Mean (G-Mean), as an unbiased metric, and the Matthews Correlation Coefficient (MCC), to take into account the errors when dealing with unbalanced datasets, were proposed. Therefore, other measures besides accuracy may be worth calculating.

A multi-class problem can calculate many measures in two different ways, the micro-average and the macro-average. The counts of true positives, true negatives, false positives, and false negatives are aggregated for all classes to compute micro-averages and then the corresponding metric is computed using the total counts. Micro-averaging weights each data instance equally, regardless of the class label and the number of instances of each class. Macro-averaging first calculates each class metric value and then takes the arithmetic mean of all class metric values. Macro-averaging weights each class equally, regardless of the number of instances.

$$MCC = \frac{c \times s - \sum_{k=1}^K p_k \times t_k}{\sqrt{(s^2 - \sum_{k=1}^K p_k^2) \times (s^2 - \sum_{k=1}^K t_k^2)}} \quad (3)$$

Equation 3 shows the formula for computing the MCC metric for a multiclass problem. For a problem of K classes, s is the number of instances, c is the number of instances correctly classified, t_k is the number of instances of class k and p_k is the number of times class k was predicted.

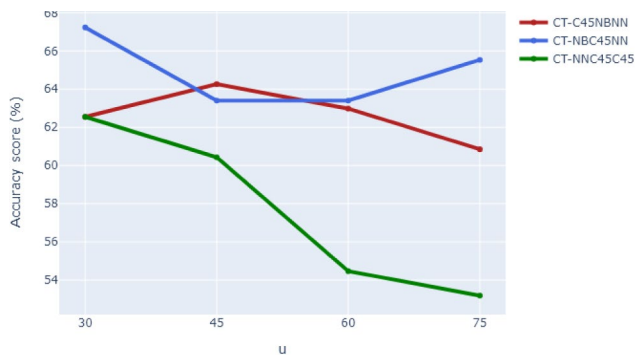
Table 12 The G-mean metric scores of the different algorithms

| Labeled instances (%) | Super-vised NN | Semi-supervised methods | | |
|-----------------------|----------------|-------------------------|-------------|-------------|
| | | CT-NBC45NN | CT-NNC45C45 | CT-NNNNNN |
| 25 | 0.64 | 0.67 | 0.52 | 0.64 |
| 50 | 0.64 | 0.65 | 0.69 | 0.63 |
| 75 | 0.69 | 0.69 | 0.64 | 0.70 |
| 100 | 0.73 | – | – | – |

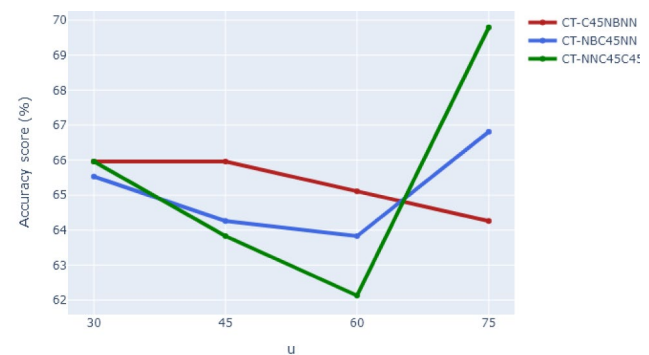
Each row represents the results obtained by training the algorithm on a dataset with a different percentage of labeled instances, ranging from 25 to 100%. The first column contains the percentage of labeled instances in the dataset, the second column contains the results of the SL algorithm for comparison, and the remaining columns are the results for the different SSL algorithms. Numbers that appear in bold type are the best result in each row

Table 11 shows the MCC (Jurman et al., 2012) scores obtained for the algorithms for each dataset containing

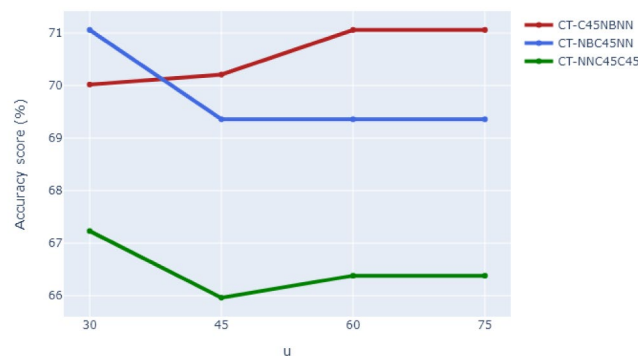
the most accurate models for each dataset (percentage of labeled instances). As had been expected, the MCC scores were lower than those obtained for the accuracy metric, because the MCC scores were only obtained after all the measures (TP, TN, FP, and FN) had yielded good results, but the algorithms had made some misclassifications. The MCC score ranges from -1 up to 1 for perfect agreement between predicted and actual values, and a score of 0 means that the predicted values are random compared to the actual values. The MCC scores in Table 11 are not very high, but the best score for each dataset is higher than 0.50 and is relatively close to the maximum score that can be obtained using the reference NN algorithm trained on the fully labeled dataset, which is 0.63.



(a) 25% labeled dataset.



(b) 50% labeled dataset.



(c) 75% labeled dataset.

Fig. 12 Sensibility to parameters tuning in terms of accuracy of the 3 best co-training algorithms varying the u parameter of 30, 45, 60 and 75 trained on the 25%, 50% and 75% labeled datasets

a percentage of labeled instances. Bold values identifies

$$G - \text{mean}(k) = \sqrt{\frac{C_{kk}}{\sum_{i=1, i \neq k}^K C_{ik}} \times \frac{\sum_{i=1, i \neq k}^K \sum_{j=1, j \neq k}^K C_{ij}}{\sum_{i=1, i \neq k}^K \sum_{j=1, j \neq k}^K C_{ij}}} \quad (4)$$

Equation 4 shows the formula for calculating the multi-class geometric mean for class k , when an instance is classified into one of K classes. C_{ij} represents the value in the confusion matrix for the row i and column j . The overall average G-mean score of the classifier can be calculated in two ways: by averaging the G-mean score for each of the K classes (macro-average), or by calculating the formula values for each class, adding them together, and then using the sum to calculate the average G-mean score (micro-average).

Table 12 shows the G-Mean (Fernández et al., 2018) scores obtained for the algorithms for each dataset containing a percentage of labeled instances.

First and second SSL algorithm coincide in Tables 11 and 12, and the third one is different. However, the results are similar, with the G-Mean scores higher than the MCC scores. G-Mean scores are closer to the accuracy scores shown in Table 8 (in percentages).

Study of the sensitivity to parameters values of the co-training algorithm

A study has been conducted to investigate how sensitive the Co-Training algorithm is to the k and u parameters. The value of parameter u (the size of the unlabeled instance subset) was varied through 30, 45, 60 and 75. The value of the k parameter (the number of iterations) was varied through 15, 30 and 40. An accuracy score was computed for each combination of these parameters for the three different datasets. The chosen algorithms are those appearing in Table 8.

Figure 12 illustrates the variation in accuracy of the three algorithms presented in Table 8, depending on four different values of the parameter u (30, 45, 60, and 75). Each sub-figure shows the accuracy score for each algorithm at each value of u for one of three datasets containing 25%, 50%, or 75% labeled data. The red line shows the accuracy scores for the CT-C45NBNNCT algorithm; the blue line shows the accuracy scores for the CT-NBC45NN algorithm; and the green line shows the accuracy scores for the CT-NNC45C45 algorithm. In most of the cases the influence of the tuning procedure is low; for instance, in 75% dataset, the effect is under 2% for the 3 algorithms. Only in two cases for the CT-NNC45C45 algorithm, the effect increases up to 8.5% of the accuracy, showing the low sensibility of these modeling solution to parameters tuning under the 3 considered labeling percentages.

The k parameter has no effect on the results. The values obtained for each combination of algorithm and dataset are identical, regardless of k 's value. This behavior can possibly be explained by the fact that Co-Training uses a filter for pseudo-labeling an instance: sufficient confidence is required. When no instance fulfills the criterion, none are pseudo-labeled. However, the accuracy score obtained is affected by u 's value, although its effect seems neither uniform nor predictable. For example, when using the 25% labeled dataset, the CT-C45NBNN algorithm produced its best result with $u = 45$, whereas the CT-NBC45NN and CT-NNC45C45 algorithms produced their best results with $u = 30$. Using the 50% labeled dataset, the CT-NNC45C45 algorithm achieved its best result with $u = 75$, while using the 75% labeled dataset, the CT-C45NBNN algorithm achieved its best result with either $u = 60$ or $u = 75$, whereas the CT-NBC45NN and CT-NNC45C45 algorithms achieved their best results with $u = 30$. For the dataset labeled with 25% of the data, the worst-case difference in results for the same algorithm is 17.6% (represented by the green line). For the dataset with 50% labels, the difference is 12.3% (also represented by the green line). For the 75% labeled dataset, the difference is 1.9% (also represented by the green line).

According to the results in Fig. 12, it is difficult to select a unique value for parameter u that enables any given algorithm to achieve its optimal result. It is possible that each algorithm and dataset will require its own value for this parameter in order to achieve the best possible score. However, the results obtained using $u = 75$ are the best, or close to the best, except for the CT-NNC45C45 algorithm for the 25% labeled dataset. The default Co-Training parameter $u = 75$ might be considered a good general value. As might be expected, the fewer labeled instances in the dataset, the greater the effect of the parameter value. Using a dataset with 75% labeled instances results in less variation in the results obtained for the three Co-Training algorithm combinations.

Discussion

The results obtained using input data from other experiments and studies were better than expected. Although interpretable as a first evaluation before building a proper classification, these results are promising. The accuracy obtained is quite good, taking into account the specific characteristics of the process: not many instances, collected from publicly

available datasets from other studies and publications, and approaching the problem as a semi-supervised one.

From the results shown in Table 8, it can be concluded that a semi-supervised based solution is worthwhile even with a relatively small dataset. The best results obtained using an SSL approach systematically outperformed the reference SL algorithm trained on the same subset of labeled instances. Moreover, the best SSL result trained on the dataset containing 75% labeled and 25% unlabeled instances achieved almost the same accuracy as the SL reference trained with the 100% labeled instances.

The accuracy results were also consistent with the results shown in Tables 12 and 11 for the MCC and G-mean metrics used as reference metrics for unbalanced datasets.

Obviously, the results were far from perfect, but the samples were not taken from one machine and one specific operation, nor from the same machine and the same operation. They were taken from various academic articles authored by different researchers at different facilities. The databases were compiled at an insignificant cost in terms of both time and money. They might represent an estimate of what could be obtained by using them together for a first evaluation of the classifier results. The machine learning model was also built using SSL methods. This initial, fast, and inexpensive model can be used to assess whether or not it is worth collecting actual data from the target machine(s) and process(es), and building the specific classifier for the process that is under development.

Conclusions and future work

Firstly, this research has proposed an information fusion approach based in collecting different datasets from very different experiments in terms of authors, facilities and manufacturing conditions. The ranking of the features in the new dataset has assured the balance between homogeneity and diversity in the new dataset. The discretization of the output quality indicator, in terms of industrial requirements and standards, is also a key element in this approach. This dataset generation procedure has demonstrated to be capable of training a machine learning model with high accuracy for a new promising manufacturing process: single point incremental forming and the prediction of workpiece surface roughness.

From the industrial point of view, this procedure will be suitable for an initial, quick and inexpensive evaluation in terms of accuracy and reliability requirements of machine learning models to decide whether or not the results are

good enough to use the model. Besides, in the latter case, whether it might be worthwhile to collect one's own dataset and train a model based on one's own machines and developed processes to create an *ad-hoc* model for each set of workshop conditions and for each workpiece.

The SSL approach has also helped to improve the results when some instances are labeled and other unlabeled instances are also available, a very common industrial situation where instance labeling is a costly task. In addition, it has shown low sensibility for fine-tuning of the main parameters of the algorithm, an issue of special interest for industrial implementation. Unfortunately, statistical significance has not been mainly proved between SL and SSL models performance, due to limited dataset size.

The results encourage further study of the single-point incremental deformation process. In future work, semi-supervised methods will be used to try to predict formability, geometry accuracy, and forming forces generated during the process. Besides, techniques for balancing relatively scarce unbalanced datasets and learning methods that take into account the unbalanced nature of the dataset also need to be considered. Unbalanced datasets are very common in industrial data. The absence of instances of the worst defects or the presence of relatively few instances of faulty working conditions is common in industrial datasets, and that situation needs to be considered as the norm rather than the exception, for the sorts of dataset-based solutions described in this study. Finally, the extension of the dataset and the battery of repetitions will help to assure the statistical significance in the different performance of SL and SSL techniques.

Appendix 1: Dataset

Table 13 shows the dataset for the experiments. Two columns were coded, to reduce the width of the table. In the column headed Figure, 'C' stands for Cone; 'P' stands for Pyramid; 'C/P' stands for Cone/Pyramid; and 'O' stands for Oval. In the column Lubrication, 'D' stands for Dry; 'M' stands for Mineral Oil; 'W' stands for Water-oil Emulsion; and 'G' stands for Grease.

Table 13 Dataset

| Machine | Alloy | RPM | Feed | Rate | Step | Size | Tool | Diameter | “Wall angle” | Figure | Thickness | Lubrication | NX |
|---------|----------|------|------|------|------|------|------|----------|--------------|--------|-----------|-------------|----|
| 1 | AA5052 | 1500 | 500 | | 0,25 | | 12 | | 63,43 | C | 1 | D | N9 |
| 1 | AA5052 | 2000 | 500 | | 0,25 | | 12 | | 63,43 | C | 1 | D | N9 |
| 1 | AA5052 | 1500 | 800 | | 0,25 | | 12 | | 63,43 | C | 1 | D | N9 |
| 1 | AA5052 | 2000 | 800 | | 0,25 | | 12 | | 63,43 | C | 1 | D | N9 |
| 1 | AA5052 | 1500 | 500 | | 0,75 | | 12 | | 63,43 | C | 1 | D | N9 |
| 1 | AA5052 | 2000 | 500 | | 0,75 | | 12 | | 63,43 | C | 1 | D | N9 |
| 1 | AA5052 | 1500 | 800 | | 0,75 | | 12 | | 63,43 | C | 1 | D | N9 |
| 1 | AA5052 | 2000 | 800 | | 0,75 | | 12 | | 63,43 | C | 1 | D | N9 |
| 2 | AA1050-O | 500 | 500 | | 0,25 | | 5 | | 29,74 | C | 1 | D | N8 |
| 2 | AA1050-O | 500 | 1100 | | 0,25 | | 10 | | 29,74 | C | 1 | D | N8 |
| 2 | AA1050-O | 1000 | 1500 | | 0,5 | | 10 | | 29,74 | C | 1 | D | N7 |
| 2 | AA1050-O | 2000 | 800 | | 1,00 | | 10 | | 29,74 | C | 1 | D | N9 |
| 2 | AA1050-O | 500 | 1500 | | 0,25 | | 20 | | 29,74 | C | 1 | D | N8 |
| 2 | AA1050-O | 1000 | 1100 | | 0,5 | | 20 | | 29,74 | C | 1 | D | N8 |
| 2 | AA1050-O | 1500 | 800 | | 0,75 | | 20 | | 29,74 | C | 1 | D | N6 |
| 2 | AA1050-O | 2000 | 500 | | 1,00 | | 20 | | 29,74 | C | 1 | D | N7 |
| 2 | AA1050-O | 500 | 800 | | 0,25 | | 30 | | 29,74 | C | 1 | D | N8 |
| 2 | AA1050-O | 1000 | 500 | | 0,5 | | 30 | | 29,74 | C | 1 | D | N7 |
| 2 | AA1050-O | 1500 | 1500 | | 0,75 | | 30 | | 29,74 | C | 1 | D | N6 |
| 2 | AA1050-O | 2000 | 1100 | | 1,00 | | 30 | | 29,74 | C | 1 | D | N5 |
| 3 | AA7075-O | 0 | 4000 | | 0,2 | | 20 | | 55 | P | 1,6 | M | N6 |
| 3 | AA7075-O | 0 | 6000 | | 0,2 | | 20 | | 55 | P | 1,6 | M | N6 |
| 3 | AA7075-O | 0 | 4000 | | 0,8 | | 20 | | 55 | P | 1,6 | M | N7 |
| 3 | AA7075-O | 0 | 6000 | | 0,8 | | 20 | | 55 | P | 1,6 | M | N7 |
| 3 | AA7075-O | 0 | 5000 | | 0,5 | | 15 | | 55 | P | 1,02 | M | N6 |
| 3 | AA7075-O | 0 | 5000 | | 0,5 | | 25 | | 55 | P | 1,02 | M | N6 |
| 3 | AA7075-O | 0 | 5000 | | 0,5 | | 15 | | 55 | P | 2,54 | M | N7 |
| 3 | AA7075-O | 0 | 5000 | | 0,5 | | 25 | | 55 | P | 2,54 | M | N6 |
| 3 | AA7075-O | 0 | 5000 | | 0,2 | | 15 | | 55 | P | 1,6 | M | N6 |
| 3 | AA7075-O | 0 | 5000 | | 0,8 | | 15 | | 55 | P | 1,6 | M | N7 |
| 3 | AA7075-O | 0 | 5000 | | 0,2 | | 25 | | 55 | P | 1,6 | M | N5 |
| 3 | AA7075-O | 0 | 5000 | | 0,8 | | 25 | | 55 | P | 1,6 | M | N6 |
| 3 | AA7075-O | 0 | 4000 | | 0,5 | | 20 | | 55 | P | 1,02 | M | N6 |
| 3 | AA7075-O | 0 | 4000 | | 0,5 | | 20 | | 55 | P | 2,54 | M | N6 |
| 3 | AA7075-O | 0 | 6000 | | 0,5 | | 20 | | 55 | P | 1,02 | M | N6 |
| 3 | AA7075-O | 0 | 6000 | | 0,5 | | 20 | | 55 | P | 2,54 | M | N6 |
| 3 | AA7075-O | 0 | 5000 | | 0,2 | | 20 | | 55 | P | 1,02 | M | N6 |
| 3 | AA7075-O | 0 | 5000 | | 0,2 | | 20 | | 55 | P | 2,54 | M | N6 |
| 3 | AA7075-O | 0 | 5000 | | 0,8 | | 20 | | 55 | P | 1,02 | M | N7 |
| 3 | AA7075-O | 0 | 5000 | | 0,8 | | 20 | | 55 | P | 2,54 | M | N8 |
| 3 | AA7075-O | 0 | 4000 | | 0,5 | | 15 | | 55 | P | 1,6 | M | N6 |
| 3 | AA7075-O | 0 | 4000 | | 0,5 | | 25 | | 55 | P | 1,6 | M | N6 |
| 3 | AA7075-O | 0 | 6000 | | 0,5 | | 15 | | 55 | P | 1,6 | M | N6 |
| 3 | AA7075-O | 0 | 6000 | | 0,5 | | 25 | | 55 | P | 1,6 | M | N6 |
| 3 | AA7075-O | 0 | 5000 | | 0,5 | | 20 | | 55 | P | 1,6 | M | N6 |
| 3 | AA7075-O | 0 | 5000 | | 0,5 | | 20 | | 55 | P | 1,6 | M | N6 |
| 3 | AA7075-O | 0 | 5000 | | 0,5 | | 20 | | 55 | P | 1,6 | M | N6 |
| 4 | AA1100 | 0 | 0 | | 0,25 | | 5 | | 30 | C/P | 1 | M | N7 |
| 4 | AA1100 | 0 | 0 | | 0,75 | | 5 | | 30 | C/P | 1 | M | N8 |
| 4 | AA1100 | 0 | 0 | | 1,25 | | 5 | | 30 | C/P | 1 | M | N9 |
| 4 | AA1100 | 0 | 0 | | 0,25 | | 5 | | 50 | C/P | 1 | M | N7 |
| 4 | AA1100 | 0 | 0 | | 0,75 | | 5 | | 50 | C/P | 1 | M | N9 |
| 4 | AA1100 | 0 | 0 | | 1,25 | | 5 | | 50 | C/P | 1 | M | N9 |
| 4 | AA1100 | 0 | 0 | | 0,25 | | 5 | | 70 | C/P | 1 | M | N7 |
| 4 | AA1100 | 0 | 0 | | 0,75 | | 5 | | 70 | C/P | 1 | M | N9 |

Table 13 (continued)

| Machine | Alloy | RPM | Feed | Rate | Step | Size | Tool | Diameter | “Wall angle” | Figure | Thickness | Lubrication | NX |
|---------|---------|-----|------|------|------|------|------|----------|--------------|--------|-----------|-------------|----|
| 4 | AA1100 | 0 | 0 | | 1,25 | | 5 | | 70 | C/P | 1 | M | N8 |
| 4 | AA1100 | 0 | 0 | | 0,25 | | 10 | | 30 | C/P | 1 | M | N8 |
| 4 | AA1100 | 0 | 0 | | 0,75 | | 10 | | 30 | C/P | 1 | M | N8 |
| 4 | AA1100 | 0 | 0 | | 1,25 | | 10 | | 30 | C/P | 1 | M | N7 |
| 4 | AA1100 | 0 | 0 | | 0,25 | | 10 | | 50 | C/P | 1 | M | N7 |
| 4 | AA1100 | 0 | 0 | | 0,75 | | 10 | | 50 | C/P | 1 | M | N8 |
| 4 | AA1100 | 0 | 0 | | 1,25 | | 10 | | 50 | C/P | 1 | M | N8 |
| 4 | AA1100 | 0 | 0 | | 0,25 | | 10 | | 70 | C/P | 1 | M | N6 |
| 4 | AA1100 | 0 | 0 | | 0,75 | | 10 | | 70 | C/P | 1 | M | N8 |
| 4 | AA1100 | 0 | 0 | | 1,25 | | 10 | | 70 | C/P | 1 | M | N7 |
| 4 | AA1100 | 0 | 0 | | 0,25 | | 15 | | 30 | C/P | 1 | M | N7 |
| 4 | AA1100 | 0 | 0 | | 0,75 | | 15 | | 30 | C/P | 1 | M | N7 |
| 4 | AA1100 | 0 | 0 | | 1,25 | | 15 | | 30 | C/P | 1 | M | N7 |
| 4 | AA1100 | 0 | 0 | | 0,25 | | 15 | | 50 | C/P | 1 | M | N6 |
| 4 | AA1100 | 0 | 0 | | 0,75 | | 15 | | 50 | C/P | 1 | M | N6 |
| 4 | AA1100 | 0 | 0 | | 1,25 | | 15 | | 50 | C/P | 1 | M | N7 |
| 4 | AA1100 | 0 | 0 | | 0,25 | | 15 | | 70 | C/P | 1 | M | N6 |
| 4 | AA1100 | 0 | 0 | | 0,75 | | 15 | | 70 | C/P | 1 | M | N8 |
| 4 | AA1100 | 0 | 0 | | 1,25 | | 15 | | 70 | C/P | 1 | M | N7 |
| 5 | AA3105 | 0 | 600 | | 0,2 | | 5 | | 45 | P | 0,5 | M | N6 |
| 5 | AA3105 | 0 | 600 | | 0,2 | | 13 | | 45 | P | 0,5 | M | N5 |
| 5 | AA3105 | 0 | 600 | | 1 | | 5 | | 45 | P | 0,5 | M | N8 |
| 5 | AA3105 | 0 | 600 | | 1 | | 13 | | 45 | P | 0,5 | M | N6 |
| 5 | AA3105 | 0 | 1400 | | 0,2 | | 5 | | 45 | P | 0,5 | M | N6 |
| 5 | AA3105 | 0 | 1400 | | 0,2 | | 13 | | 45 | P | 0,5 | M | N5 |
| 5 | AA3105 | 0 | 1400 | | 1 | | 5 | | 45 | P | 0,5 | M | N8 |
| 5 | AA3105 | 0 | 1400 | | 1 | | 13 | | 45 | P | 0,5 | M | N7 |
| 5 | AA3105 | 0 | 600 | | 0,2 | | 5 | | 45 | P | 1,5 | M | N7 |
| 5 | AA3105 | 0 | 600 | | 0,2 | | 13 | | 45 | P | 1,5 | M | N6 |
| 5 | AA3105 | 0 | 600 | | 1 | | 5 | | 45 | P | 1,5 | M | N8 |
| 5 | AA3105 | 0 | 600 | | 1 | | 13 | | 45 | P | 1,5 | M | N7 |
| 5 | AA3105 | 0 | 1400 | | 0,2 | | 5 | | 45 | P | 1,5 | M | N7 |
| 5 | AA3105 | 0 | 1400 | | 0,2 | | 13 | | 45 | P | 1,5 | M | N6 |
| 5 | AA3105 | 0 | 1400 | | 1 | | 5 | | 45 | P | 1,5 | M | N8 |
| 5 | AA3105 | 0 | 1400 | | 1 | | 13 | | 45 | P | 1,5 | M | N7 |
| 5 | AA3105 | 0 | 1000 | | 0,6 | | 5 | | 45 | P | 1 | M | N7 |
| 5 | AA3105 | 0 | 1000 | | 0,6 | | 13 | | 45 | P | 1 | M | N7 |
| 5 | AA3105 | 0 | 1000 | | 0,2 | | 9 | | 45 | P | 1 | M | N7 |
| 5 | AA3105 | 0 | 1000 | | 1 | | 9 | | 45 | P | 1 | M | N7 |
| 5 | AA3105 | 0 | 600 | | 0,6 | | 9 | | 45 | P | 1 | M | N8 |
| 5 | AA3105 | 0 | 1400 | | 0,6 | | 9 | | 45 | P | 1 | M | N6 |
| 5 | AA3105 | 0 | 1000 | | 0,6 | | 9 | | 45 | P | 1 | M | N7 |
| 5 | AA3105 | 0 | 1000 | | 0,6 | | 9 | | 45 | P | 1 | M | N7 |
| 5 | AA3105 | 0 | 1000 | | 0,6 | | 9 | | 45 | P | 1 | M | N7 |
| 5 | AA3105 | 0 | 1000 | | 0,6 | | 9 | | 45 | P | 1 | M | N7 |
| 5 | AA3105 | 0 | 1000 | | 0,6 | | 9 | | 45 | P | 1 | M | N7 |
| 5 | AA3105 | 0 | 1000 | | 0,6 | | 9 | | 45 | P | 1 | M | N6 |
| 5 | AA3105 | 0 | 1000 | | 0,6 | | 9 | | 45 | P | 1 | M | N7 |
| 5 | AA3105 | 0 | 1000 | | 0,6 | | 9 | | 45 | P | 1 | M | N7 |
| 5 | AA3105 | 0 | 1000 | | 0,6 | | 9 | | 45 | P | 1 | M | N7 |
| 6 | 7075 T0 | 0 | 1000 | | 0,5 | | 10 | | 60 | P | 1 | M | N8 |
| 6 | 7075 T0 | 200 | 1000 | | 0,5 | | 10 | | 60 | P | 1 | M | N8 |
| 6 | 7075 T0 | 400 | 1000 | | 0,5 | | 10 | | 60 | P | 1 | M | N8 |
| 7 | 7075 T0 | 600 | 1000 | | 0,2 | | 5 | | 60 | P | 1 | M | N5 |
| 7 | 7075 T0 | 600 | 1000 | | 0,2 | | 10 | | 60 | P | 1 | M | N7 |

Table 13 (continued)

| Machine | Alloy | RPM | Feed | Rate | Step | Size | Tool | Diameter | “Wall angle” | Figure | Thickness | Lubrication | NX |
|---------|----------|------|------|------|-------|------|-------|----------|--------------|--------|-----------|-------------|-----|
| 7 | 7075 T0 | 600 | 1000 | | 0,2 | | 15 | 60 | | P | 1 | M | N7 |
| 7 | 7075 T0 | 600 | 1000 | | 0,4 | | 5 | 60 | | P | 1 | M | N5 |
| 7 | 7075 T0 | 600 | 1000 | | 0,4 | | 10 | 60 | | P | 1 | M | N7 |
| 7 | 7075 T0 | 600 | 1000 | | 0,4 | | 15 | 60 | | P | 1 | M | N8 |
| 7 | 7075 T0 | 600 | 1000 | | 0,6 | | 5 | 60 | | P | 1 | M | N7 |
| 7 | 7075 T0 | 600 | 1000 | | 0,6 | | 10 | 60 | | P | 1 | M | N8 |
| 7 | 7075 T0 | 600 | 1000 | | 0,6 | | 15 | 60 | | P | 1 | M | N9 |
| 7 | 7075 T0 | 0 | 1000 | | 0,2 | | 5 | 60 | | P | 1 | M | N5 |
| 7 | 7075 T0 | 0 | 1000 | | 0,2 | | 10 | 60 | | P | 1 | M | N7 |
| 7 | 7075 T0 | 0 | 1000 | | 0,2 | | 15 | 60 | | P | 1 | M | N7 |
| 7 | 7075 T0 | 0 | 1000 | | 0,4 | | 5 | 60 | | P | 1 | M | N6 |
| 7 | 7075 T0 | 0 | 1000 | | 0,4 | | 10 | 60 | | P | 1 | M | N7 |
| 7 | 7075 T0 | 0 | 1000 | | 0,4 | | 15 | 60 | | P | 1 | M | N9 |
| 7 | 7075 T0 | 0 | 1000 | | 0,6 | | 5 | 60 | | P | 1 | M | N7 |
| 7 | 7075 T0 | 0 | 1000 | | 0,6 | | 10 | 60 | | P | 1 | M | N8 |
| 7 | 7075 T0 | 0 | 1000 | | 0,6 | | 15 | 60 | | P | 1 | M | N10 |
| 8 | 3003 H14 | 1200 | 6985 | | 0,01 | | 12,7 | 48 | | O | 0,81 | W | N9 |
| 8 | 3003 H14 | 400 | 5080 | | 0,016 | | 12,7 | 34 | | O | 0,81 | W | N9 |
| 8 | 3003 H14 | 2000 | 8890 | | 0,016 | | 12,7 | 34 | | O | 0,81 | W | N9 |
| 8 | 3003 H14 | 2000 | 5080 | | 0,004 | | 12,7 | 62 | | O | 0,81 | W | N9 |
| 8 | 3003 H14 | 400 | 8890 | | 0,004 | | 12,7 | 62 | | O | 0,81 | W | N9 |
| 8 | 3003 H14 | 1200 | 6985 | | 0,01 | | 12,7 | 48 | | O | 0,81 | W | N9 |
| 8 | 3003 H14 | 400 | 8890 | | 0,016 | | 12,7 | 62 | | O | 0,81 | W | N9 |
| 8 | 3003 H14 | 1200 | 6985 | | 0,01 | | 12,7 | 48 | | O | 0,81 | W | N9 |
| 8 | 3003 H14 | 2000 | 5080 | | 0,016 | | 12,7 | 62 | | O | 0,81 | W | N9 |
| 8 | 3003 H14 | 1200 | 6985 | | 0,01 | | 12,7 | 48 | | O | 0,81 | W | N9 |
| 8 | 3003 H14 | 400 | 5080 | | 0,004 | | 12,7 | 34 | | O | 0,81 | W | N9 |
| 8 | 3003 H14 | 2000 | 8890 | | 0,004 | | 12,7 | 34 | | O | 0,81 | W | N9 |
| 8 | 3003 H14 | 2000 | 8890 | | 0,004 | | 12,7 | 62 | | O | 0,81 | W | N9 |
| 8 | 3003 H14 | 1200 | 6985 | | 0,01 | | 12,7 | 48 | | O | 0,81 | W | N9 |
| 8 | 3003 H14 | 2000 | 5080 | | 0,016 | | 12,7 | 34 | | O | 0,81 | W | N8 |
| 8 | 3003 H14 | 400 | 5080 | | 0,004 | | 12,7 | 62 | | O | 0,81 | W | N9 |
| 8 | 3003 H14 | 400 | 8890 | | 0,016 | | 12,7 | 34 | | O | 0,81 | W | N10 |
| 8 | 3003 H14 | 400 | 8890 | | 0,004 | | 12,7 | 34 | | O | 0,81 | W | N9 |
| 8 | 3003 H14 | 2000 | 5080 | | 0,004 | | 12,7 | 34 | | O | 0,81 | W | N9 |
| 8 | 3003 H14 | 2000 | 8890 | | 0,016 | | 12,7 | 62 | | O | 0,81 | W | N9 |
| 8 | 3003 H14 | 400 | 5080 | | 0,016 | | 12,7 | 62 | | O | 0,81 | W | N9 |
| 9 | 2024 O | 1000 | 1500 | | 0,2 | | 7,52 | 64 | | C | 1,2 | M | N7 |
| 9 | 2024 O | 1000 | 1500 | | 0,2 | | 11,6 | 64 | | C | 1,2 | M | N6 |
| 9 | 2024 O | 1000 | 1500 | | 0,2 | | 15,66 | 64 | | C | 1,2 | M | N6 |
| 9 | 2024 O | 1000 | 1500 | | 0,5 | | 7,52 | 64 | | C | 1,2 | M | N7 |
| 9 | 2024 O | 1000 | 1500 | | 0,5 | | 11,6 | 64 | | C | 1,2 | M | N7 |
| 9 | 2024 O | 1000 | 1500 | | 0,5 | | 15,66 | 64 | | C | 1,2 | M | N6 |
| 9 | 2024 O | 1000 | 1500 | | 0,8 | | 7,52 | 64 | | C | 1,2 | M | N7 |
| 9 | 2024 O | 1000 | 1500 | | 0,8 | | 11,6 | 64 | | C | 1,2 | M | N7 |
| 9 | 2024 O | 1000 | 1500 | | 0,8 | | 15,66 | 64 | | C | 1,2 | M | N7 |
| 9 | 2024 O | 1000 | 1500 | | 1,2 | | 7,52 | 64 | | C | 1,2 | M | N7 |
| 9 | 2024 O | 1000 | 1500 | | 1,2 | | 11,6 | 64 | | C | 1,2 | M | N7 |
| 9 | 2024 O | 1000 | 1500 | | 1,2 | | 15,66 | 64 | | C | 1,2 | M | N7 |
| 9 | 2024 O | 0 | 1500 | | 0,5 | | 7,52 | 64 | | C | 1,2 | M | N7 |
| 9 | 2024 O | 0 | 1500 | | 0,5 | | 11,6 | 64 | | C | 1,2 | M | N7 |
| 9 | 2024 O | 0 | 1500 | | 0,5 | | 15,66 | 64 | | C | 1,2 | M | N7 |
| 9 | 2024 O | 500 | 1500 | | 0,5 | | 7,52 | 64 | | C | 1,2 | M | N7 |
| 9 | 2024 O | 500 | 1500 | | 0,5 | | 11,6 | 64 | | C | 1,2 | M | N7 |

Table 13 (continued)

| Machine | Alloy | RPM | Feed | Rate | Step | Size | Tool | Diameter | “Wall angle” | Figure | Thickness | Lubrication | NX |
|---------|----------|------|-------|------|-------|------|-------|----------|--------------|--------|-----------|-------------|----|
| 9 | 2024 O | 500 | 1500 | | 0,5 | | 15,66 | 64 | | C | 1,2 | M | N7 |
| 9 | 2024 O | 1000 | 1500 | | 0,5 | | 7,52 | 64 | | C | 1,2 | M | N7 |
| 9 | 2024 O | 1000 | 1500 | | 0,5 | | 11,6 | 64 | | C | 1,2 | M | N7 |
| 9 | 2024 O | 1000 | 1500 | | 0,5 | | 15,66 | 64 | | C | 1,2 | M | N6 |
| 9 | 2024 O | 1500 | 1500 | | 0,5 | | 7,52 | 64 | | C | 1,2 | M | N7 |
| 9 | 2024 O | 1500 | 1500 | | 0,5 | | 11,6 | 64 | | C | 1,2 | M | N6 |
| 9 | 2024 O | 1500 | 1500 | | 0,5 | | 15,66 | 64 | | C | 1,2 | M | N6 |
| 10 | 6063 | 0 | 1000 | | 0,5 | | 8 | 55 | | C | 0,55 | D | N8 |
| 10 | 6063 | 250 | 2000 | | 1 | | 8 | 65 | | C | 0,55 | W | N8 |
| 10 | 6063 | 500 | 2500 | | 1,5 | | 8 | 76 | | C | 0,55 | G | N8 |
| 10 | 6063 | 0 | 2000 | | 0,5 | | 8 | 63 | | C | 1,09 | W | N8 |
| 10 | 6063 | 250 | 2500 | | 1 | | 8 | 73 | | C | 1,09 | G | N8 |
| 10 | 6063 | 500 | 1000 | | 1,5 | | 8 | 62 | | C | 1,09 | D | N9 |
| 10 | 6063 | 250 | 1000 | | 0,5 | | 8 | 84 | | C | 1,67 | G | N8 |
| 10 | 6063 | 500 | 2000 | | 1 | | 8 | 55 | | C | 1,67 | D | N9 |
| 10 | 6063 | 0 | 2500 | | 1,5 | | 12 | 64 | | C | 1,67 | W | N9 |
| 10 | 6063 | 500 | 2500 | | 0,5 | | 12 | 64 | | C | 0,55 | W | N8 |
| 10 | 6063 | 0 | 1000 | | 1 | | 12 | 67 | | C | 0,55 | G | N7 |
| 10 | 6063 | 250 | 2000 | | 1,5 | | 12 | 47 | | C | 0,55 | D | N9 |
| 10 | 6063 | 250 | 2500 | | 0,5 | | 12 | 52 | | C | 1,09 | D | N8 |
| 10 | 6063 | 500 | 1000 | | 1 | | 12 | 73 | | C | 1,09 | W | N8 |
| 10 | 6063 | 0 | 2000 | | 1,5 | | 12 | 69 | | C | 1,09 | G | N8 |
| 10 | 6063 | 500 | 2000 | | 0,5 | | 12 | 82 | | C | 1,67 | G | N8 |
| 10 | 6063 | 0 | 2500 | | 1 | | 12 | 46 | | C | 1,67 | D | N9 |
| 10 | 6063 | 250 | 1000 | | 1,5 | | 12 | 72 | | C | 1,67 | W | N8 |
| 11 | 3003 | 96 | 25 | | 1,3 | | 12,7 | 45 | | P | 1 | D | N9 |
| 11 | 3003 | 96 | 25 | | 1 | | 12,7 | 45 | | P | 1 | D | N8 |
| 11 | 3003 | 96 | 25 | | 0,76 | | 12,7 | 45 | | P | 1 | D | N7 |
| 11 | 3003 | 96 | 25 | | 0,51 | | 12,7 | 45 | | P | 1 | D | N7 |
| 11 | 3003 | 96 | 25 | | 0,25 | | 12,7 | 45 | | P | 1 | D | N6 |
| 11 | 3003 | 96 | 25 | | 0,13 | | 12,7 | 45 | | P | 1 | D | N6 |
| 11 | 3003 | 96 | 25 | | 0,051 | | 12,7 | 45 | | P | 1 | D | N6 |
| 11 | 3003 | 0 | 25 | | 0,51 | | 12,7 | 45 | | P | 1 | D | N7 |
| 11 | 3003 | 1000 | 25 | | 0,51 | | 12,7 | 45 | | P | 1 | D | N7 |
| 11 | 3003 | 2300 | 25 | | 0,51 | | 12,7 | 45 | | P | 1 | D | N7 |
| 12 | 1200 H14 | 1250 | 50 | | 0,5 | | 6 | 80 | | C | 0,91 | M | N8 |
| 12 | 1200 H14 | 1500 | 50 | | 0,5 | | 6 | 80 | | C | 0,91 | M | N9 |
| 12 | 1200 H14 | 1750 | 50 | | 0,5 | | 6 | 80 | | C | 0,91 | M | N9 |
| 12 | 1200 H14 | 2000 | 50 | | 0,5 | | 6 | 80 | | C | 0,91 | M | N9 |
| 12 | 1200 H14 | 1250 | 50 | | 0,4 | | 6 | 80 | | C | 0,91 | M | N8 |
| 12 | 1200 H14 | 1250 | 50 | | 0,3 | | 6 | 80 | | C | 0,91 | M | N8 |
| 12 | 1200 H14 | 1250 | 50 | | 0,2 | | 6 | 80 | | C | 0,91 | M | N8 |
| 13 | 5052 H32 | 0 | 317,5 | | 0,381 | | 12,7 | 45 | | P | 0,8 | G | N7 |
| 13 | 5052 H32 | 0 | 635 | | 0,381 | | 12,7 | 45 | | P | 0,8 | G | N7 |
| 13 | 5052 H32 | 0 | 1270 | | 0,381 | | 12,7 | 45 | | P | 0,8 | G | N7 |
| 13 | 5052 H32 | 0 | 317,5 | | 0,762 | | 12,7 | 45 | | P | 0,8 | G | N8 |
| 13 | 5052 H32 | 0 | 635 | | 0,762 | | 12,7 | 45 | | P | 0,8 | G | N8 |
| 13 | 5052 H32 | 0 | 1270 | | 0,762 | | 12,7 | 45 | | P | 0,8 | G | N8 |
| 13 | 5052 H32 | 0 | 317,5 | | 0,381 | | 12,7 | 60 | | P | 0,8 | G | N7 |
| 13 | 5052 H32 | 0 | 635 | | 0,381 | | 12,7 | 60 | | P | 0,8 | G | N7 |
| 13 | 5052 H32 | 0 | 1270 | | 0,381 | | 12,7 | 60 | | P | 0,8 | G | N7 |
| 13 | 5052 H32 | 0 | 317,5 | | 0,762 | | 12,7 | 60 | | P | 0,8 | G | N7 |
| 13 | 5052 H32 | 0 | 635 | | 0,762 | | 12,7 | 60 | | P | 0,8 | G | N7 |
| 13 | 5052 H32 | 0 | 1270 | | 0,762 | | 12,7 | 60 | | P | 0,8 | G | N7 |

Table 13 (continued)

| Machine | Alloy | RPM | Feed Rate | Step Size | Tool | Diameter | “Wall angle” | Figure | Thickness | Lubrication | NX |
|---------|---------|------|-----------|-----------|------|----------|--------------|--------|-----------|-------------|----|
| 14 | 1100 H0 | 0 | 2000 | 0,5 | 8 | 60 | 60 | P | 1 | M | N7 |
| 14 | 1100 H0 | 1500 | 2000 | 0,5 | 8 | 60 | 60 | P | 1 | M | N8 |
| 14 | 1100 H0 | 0 | 3500 | 0,5 | 8 | 60 | 60 | P | 1 | M | N7 |
| 14 | 1100 H0 | 1500 | 3500 | 0,5 | 8 | 60 | 60 | P | 1 | M | N8 |
| 14 | 1100 H0 | 0 | 2000 | 1 | 8 | 60 | 60 | P | 1 | M | N8 |
| 14 | 1100 H0 | 1500 | 2000 | 1 | 8 | 60 | 60 | P | 1 | M | N8 |
| 14 | 1100 H0 | 0 | 3500 | 1 | 8 | 60 | 60 | P | 1 | M | N8 |
| 14 | 1100 H0 | 1500 | 3500 | 1 | 8 | 60 | 60 | P | 1 | M | N8 |
| 15 | 1100 | 500 | 1000 | 0,1 | 14 | 40 | 40 | P | 0,9 | G | N8 |
| 15 | 1100 | 500 | 1000 | 0,1 | 14 | 40 | 40 | P | 0,9 | M | N8 |
| 15 | 1100 | 500 | 1500 | 0,2 | 14 | 50 | 50 | P | 0,9 | G | N7 |
| 15 | 1100 | 500 | 1500 | 0,2 | 14 | 50 | 50 | P | 0,9 | M | N7 |
| 15 | 1100 | 500 | 2000 | 0,3 | 14 | 60 | 60 | P | 0,9 | G | N7 |
| 15 | 1100 | 500 | 2000 | 0,3 | 14 | 60 | 60 | P | 0,9 | M | N8 |
| 15 | 1100 | 1000 | 1000 | 0,1 | 14 | 60 | 60 | P | 0,9 | G | N7 |
| 15 | 1100 | 1000 | 1000 | 0,1 | 14 | 60 | 60 | P | 0,9 | M | N7 |
| 15 | 1100 | 1000 | 1500 | 0,2 | 14 | 40 | 40 | P | 0,9 | G | N7 |
| 15 | 1100 | 1000 | 1500 | 0,2 | 14 | 40 | 40 | P | 0,9 | M | N8 |
| 15 | 1100 | 1000 | 2000 | 0,3 | 14 | 50 | 50 | P | 0,9 | G | N7 |
| 15 | 1100 | 1000 | 2000 | 0,3 | 14 | 50 | 50 | P | 0,9 | M | N7 |
| 15 | 1100 | 1500 | 1000 | 0,1 | 14 | 50 | 50 | P | 0,9 | G | N7 |
| 15 | 1100 | 1500 | 1000 | 0,1 | 14 | 50 | 50 | P | 0,9 | M | N7 |
| 15 | 1100 | 1500 | 1500 | 0,2 | 14 | 60 | 60 | P | 0,9 | G | N7 |
| 15 | 1100 | 1500 | 1500 | 0,2 | 14 | 60 | 60 | P | 0,9 | M | N7 |
| 15 | 1100 | 1500 | 2000 | 0,3 | 14 | 40 | 40 | P | 0,9 | G | N7 |
| 15 | 1100 | 1500 | 2000 | 0,3 | 14 | 40 | 40 | P | 0,9 | M | N8 |

Funding Open access funding provided by FEDER European Funds and the Junta de Castilla y León under the Research and Innovation Strategy for Smart Specialization (RIS3) of Castilla y León 2021-2027. This work was supported by the Junta de Castilla y León under project BU055P20 (JCyL/FEDER, UE), the Spanish Ministry of Science and Innovation under projects TED2021-129648B-I00, TED2021-129485B-C43 and PID2020-119894GBI00/AEI/10.13039/501100011033, co-financed through European Union FEDER funds and the “NextGenerationEU/PRTR” Recovery, Transformation and Resilience Plan.

Data availability The data supporting the findings of this study are provided within the manuscript.

Declarations

Conflict of interest The authors declare that they have no Conflict of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright

holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Afonso, D., Sousa, R., & Torcato, R. (2018). Integration of design rules and process modelling within SPIF technology—a review on the industrial dissemination of single point incremental forming. *The International Journal of Advanced Manufacturing Technology*, 94(9), 4387–4399. <https://doi.org/10.1007/s00170-017-1130-3>
- Alloghani, M., Al-Jumeily, D., Mustafina, J., Hussain, A., & Aljaaf, A. J. (2020). A systematic review on supervised and unsupervised machine learning algorithms for data science. In M. W. Berry, A. Mohamed, & B. W. Yap (Eds.), *Supervised and unsupervised learning for data science* (pp. 3–21). Cham: Springer. https://doi.org/10.1007/978-3-030-22475-2_1
- Afonso, D., Sousa, R. A., Torcato, R., & Pires, L. (2019). *Incremental forming as a rapid tooling process*. Springer, Cham: Springer-Briefs in Applied Sciences and Technology. <https://doi.org/10.1007/978-3-030-15360-1>
- Azevedo, N. G., Farias, J. S., Bastos, R. P., Teixeira, P., Davim, J. P., & Sousa, R. J. (2015). Lubrication aspects during single point incremental forming for steel and aluminum materials. *International Journal of Precision Engineering and Manufacturing*, 16(3), 589–595. <https://doi.org/10.1007/s12541-015-0079-0>
- Aggogeri, F., Pellegrini, N., & Tagliani, F. L. (2021). Recent advances on machine learning applications in machining processes. *Applied Sciences*, 11(18), 8764. <https://doi.org/10.3390/app11188764>

- Bermudez, G., Bustamante-Correa, A., & Adrian, B. (2016). Statistical analysis of the main incremental forming process parameters that contribute to change the roughness in an experimental geometry. *Dyna (Bilbao)*, 91(6), 688–693.
- Behera, A. K., de Sousa, R. A., Ingarao, G., & Oleksik, V. (2017). Single point incremental forming: An assessment of the progress and technology trends from 2005 to 2015. *Journal of Manufacturing Processes*, 27, 37–62. <https://doi.org/10.1016/j.jmappro.2017.03.014>
- Blum, A., Mitchell, T. (1998). Combining labeled and unlabeled data with co-training. In Proceedings of the annual ACM conference on computational learning theory (pp. 92–100). <https://doi.org/10.1145/279943.279962>
- Breiman, L. (2001). Random Forests. *Machine Learning*, 45(1), 5–32. <https://doi.org/10.1023/A:1010933404324>
- Bustillo, A., Reis, R., Machado, A., & Pimenov, D. (2022). Improving the accuracy of machine-learning models with data from machine test repetitions. *Journal of Intelligent Manufacturing*, 33(1), 203–221. <https://doi.org/10.1007/s10845-020-01661-3>
- Chen, T., Sampath, V., May, M., Shan, S., Jorg, O., Aguilar Martín, J., Stamer, F., Fantoni, G., Tosello, G., & Calaon, M. (2023). Machine learning in manufacturing towards industry 4.0: From ‘for now’ to ‘four-know’. *Applied Sciences*, 13(3), 1903. <https://doi.org/10.3390/app13031903>
- Desai, B. V., Desai, K. P., & Raval, H. K. (2014). Die-less rapid prototyping process: Parametric investigations. *Procedia Materials Science*, 6, 666–673. <https://doi.org/10.1016/j.mspro.2014.07.082>. 3rd International Conference on Materials Processing and Characterisation (ICMPC 2014).
- Durante, M., Formisano, A., & Langella, A. (2011). Observations on the influence of tool-sheet contact conditions on an incremental forming process. *Journal of Materials Engineering and Performance*, 20(6), 914–946. <https://doi.org/10.1007/s11665-010-9742-x>
- Durante, M., Formisano, A., Langella, A., & Capece Minutolo, F. M. (2009). The influence of tool rotation on an incremental forming process. *Journal of Materials Processing Technology*, 209(9), 4621–4626. <https://doi.org/10.1016/j.jmatprotec.2008.11.028>
- Deng, C., & Guo, M. Z. (2011). A new co-training-style random forest for computer aided diagnosis. *Journal of Intelligent Information Systems*, 36(3), 253–281. <https://doi.org/10.1007/s10844-009-0105-8>
- Dufloy, J. R., Habraken, A.-M., Cao, J., Malhotra, R., Bambach, M., Adams, D., Vanhove, H., Mohammadi, A., & Jeswiet, J. (2018). Single point incremental forming: state-of-the-art and prospects. *International Journal of Material Forming*, 11(6), 743–773. <https://doi.org/10.1007/s12289-017-1387-y>
- Echraf, S. B. M., & Hrairi, M. (2014). Significant parameters for the surface roughness in incremental forming process. *Materials and Manufacturing Processes*, 29(6), 697–703. <https://doi.org/10.1080/10426914.2014.901519>
- Fernández, A., García, S., Galar, M., Prati, R. C., Krawczyk, B., & Herrera, F. (2018). Imbalanced classification with multiple classes. *Learning from imbalanced data sets* (pp. 197–226). Cham: Springer. https://doi.org/10.1007/978-3-319-98074-4_8
- Gulati, V., Aryal, A., Katyal, P., & Goswami, A. (2016). Process Parameters Optimization in Single Point Incremental Forming. *Journal of The Institution of Engineers (India): Series C*, 97(2), 185–193. <https://doi.org/10.1007/s40032-015-0203-z>
- Halder, A., Ghosh, S., & Ghosh, A. (2013). Aggregation pheromone metaphor for semi-supervised classification. *Pattern Recognition*, 46, 2239–2248. <https://doi.org/10.1016/j.patcog.2013.01.002>
- Harfoush, A., Haapala, K. R., & Tabai, A. (2021). Application of artificial intelligence in incremental sheet metal forming: A review. *Procedia Manufacturing*, 53, 606–617. <https://doi.org/10.1016/j.promfg.2021.06.061>. 49th SME North American Manufacturing Research Conference (NAMRC 49, 2021).
- Hagan, E., & Jeswiet, J. (2004). Analysis of surface roughness for parts formed by computer numerical controlled incremental forming. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, 218(10), 1307–1312. <https://doi.org/10.1243/0954405042323559>
- Hamilton, K., & Jeswiet, J. (2010). Single point incremental forming at high feed rates and rotational speeds: Surface and structural consequences. *CIRP Annals*, 59(1), 311–314. <https://doi.org/10.1016/j.cirp.2010.03.016>
- Han, F., Mo, J.-H., & Gong, P. (2008). Incremental sheet NC forming springback prediction using genetic neural network. *Journal of Huazhong University of Science and Technology: Nature Science Edition*, 36, 121–124.
- Hady, M., Schwenker, F., & Palm, G. (2010). Semi-supervised learning for tree-structured ensembles of RBF networks with Co-Training. *Neural Networks*, 23(4), 497–509. <https://doi.org/10.1016/j.neunet.2009.09.001>
- He, X., Welo, T., & Ma, J. (2025). In-process monitoring strategies and methods in metal forming: A selective review. *Journal of Manufacturing Processes*, 138, 100–128. <https://doi.org/10.1016/j.jmappro.2025.02.011>
- Huang, T., Yu, Y., Guo, G., & Li, K. (2010). A classification algorithm based on local cluster centers with a few labeled training examples. *Knowledge Based Systems*, 26(6), 563–571. <https://doi.org/10.1016/j.knsys.2010.03.015>
- Huang, F., Zhang, X., Qin, G., Xie, J., Peng, J., Huang, S., Long, Z., & Tang, Y. (2023). Demagnetization fault diagnosis of permanent magnet synchronous motors using magnetic leakage signals. *IEEE Transactions on Industrial Informatics*, 19(4), 6105–6116. <https://doi.org/10.1109/TII.2022.3165283>
- Jan, Z., Ahamed, F., Mayer, W., Patel, N., Grossmann, G., Stumptner, M., & Kuusk, A. (2023). Artificial intelligence for industry 4.0: Systematic review of applications, challenges, and opportunities. *Expert Systems with Applications*, 216, Article 119456. <https://doi.org/10.1016/j.eswa.2022.119456>
- Jeswiet, J., Micari, F., Hirt, G., Bramley, A., Dufloy, J., & Allwood, J. (2005). Asymmetric Single Point Incremental Forming of Sheet Metal. *CIRP Annals*, 54(2), 88–114. [https://doi.org/10.1016/S0007-8506\(07\)60021-3](https://doi.org/10.1016/S0007-8506(07)60021-3)
- Jurman, G., Riccadonna, S., & Furlanello, C. (2012). A comparison of MCC and CEN error measures in multi-class prediction. *PLOS ONE*, 7(8), 1–8. <https://doi.org/10.1371/journal.pone.0041882>
- Kumar, S. P., Elangovan, S., Mohanraj, R., & Boopathi, S. (2021). Real-time applications and novel manufacturing strategies of incremental forming: An industrial perspective. *Materials Today: Proceedings*, 46, 8153–8164. <https://doi.org/10.1016/j.matpr.2021.03.109>. 3rd International Conference on Materials, Manufacturing and Modelling.
- Kumar, A., Gulati, V., & Kumar, P. (2018). Investigation of surface roughness in incremental sheet forming. *Procedia Computer Science*, 133, 1014–1020. <https://doi.org/10.1016/j.procs.2018.07.074>. International Conference on Robotics and Smart Manufacturing (RoSMa2018).
- Kumar, A. (2024). Critical state-of-the-art literature review of surface roughness in incremental sheet forming: A comparative analysis. *Applied Surface Science Advances*, 23, Article 100625. <https://doi.org/10.1016/j.apsadv.2024.100625>
- Luque, A., Carrasco, A., Martín, A., & de las Heras, A. (2019). The impact of class imbalance in classification performance metrics based on the binary confusion matrix. *Pattern Recognition*, 91, 216–231. <https://doi.org/10.1016/j.patcog.2019.02.023>
- Li, Y.-F., & Liang, D.-M. (2019). Safe semi-supervised learning: A brief introduction. *Frontiers of Computer Science*, 13(4), 669–676. <https://doi.org/10.1007/s11704-019-8452-2>

- Lee, C., & Lim, C. (2021). From technological development to social advance: A review of industry 4.0 through machine learning. *Technological Forecasting and Social Change*, 167, Article 120653. <https://doi.org/10.1016/j.techfore.2021.120653>
- Li, M., & Zhou, Z.-H. (2005). Setred: Self-training with editing. In T. B. Ho, D. Cheung, & H. Liu (Eds.), *Advances in knowledge discovery and data mining. Lecture notes on computer science* (pp. 611–621). Berlin, Heidelberg: Springer. https://doi.org/10.1007/11430919_71
- Li, M., & Zhou, Z. H. (2007). Improve computer-aided diagnosis with machine learning techniques using undiagnosed samples. *IEEE Transactions on Systems and Man and Cybernetics and Part A: Systems and Humans*, 37(6), 1088–1098. <https://doi.org/10.1109/TSMCA.2007.904745>
- Murugesan, M., Bhandari, K. S., Sajjad, M., & Jung, D.-W. (2021). Investigation of surface roughness in single point incremental sheet forming considering process parameters. *International Journal of Mechanical Engineering and Robotics Research*, 10(8), 443–451. <https://doi.org/10.18178/ijmerr.10.8.443-451>
- Martins, P. A. F., Bay, N., Skjoedt, M., & Silva, M. B. (2008). Theory of single point incremental forming. *CIRP Annals*, 57(1), 247–252. <https://doi.org/10.1016/j.cirp.2008.03.047>
- Mugendiran, V., Gnanavelbabu, A., & Ramadoss, R. (2014). Parameter optimization for surface roughness and wall thickness on aa5052 aluminium alloy by incremental forming using response surface methodology. *Procedia Engineering*, 97, 1991–2000. <http://doi.org/10.1016/j.proeng.2014.12.442>
- Ma, Q., Gao, J., Zhan, B., Guo, Y., Zhou, J., & Wang, Y. (2023). Rethinking safe semi-supervised learning: transferring the open-set problem to a close-set one. In 2023 IEEE/CVF International conference on computer vision (ICCV) (pp. 16324–16333). <https://doi.org/10.1109/ICCV51070.2023.01500>
- McAnulty, T., Jeswiet, J., & Doolan, M. (2017). Formability in single point incremental forming: A comparative analysis of the state of the art. *CIRP Journal of Manufacturing Science and Technology*, 16, 43–54. <https://doi.org/10.1016/j.cirpj.2016.07.003>
- Malik, S., Muhammad, K., & Waheed, Y. (2024). Artificial intelligence and industrial applications-a revolution in modern industries. *Ain Shams Engineering Journal*, 15(9), Article 102886. <https://doi.org/10.1016/j.asej.2024.102886>
- Maestro-Prieto, J. A., Ramírez-Sanz, J. M., Bustillo, A., & Rodríguez-Diez, J. J. (2024). Semi-supervised diagnosis of wind-turbine gearbox misalignment and imbalance faults. *Applied Intelligence*, 54(6), 4525–4544. <https://doi.org/10.1007/s10489-024-05373-6>
- Modad, O. A. A., Ryska, J., Chehade, A., & Ayoub, G. (2025). Revolutionizing sheet metal stamping through industry 5.0 digital twins: a comprehensive review. *Journal of Intelligent Manufacturing*, 36(6), 3717–3739. <https://doi.org/10.1007/s10845-024-02453-9>
- Mokhtarzadeh, M., Rodríguez-Echeverría, J., Semanjski, I., & Gautama, S. (2025). Hybrid intelligence failure analysis for industry 4.0: A literature review and future prospective. *Journal of Intelligent Manufacturing*, 36(4), 2309–2334. <https://doi.org/10.1007/s10845-024-02376-5>
- Ma, Y., Shi, H., Tan, S., Tao, Y., & Song, B. (2022). Consistency regularization auto-encoder network for semi-supervised process fault diagnosis. *IEEE Transactions on Instrumentation and Measurement*, 71, 1–15. <https://doi.org/10.1109/TIM.2022.3184346>
- Nargargoje, A., Kankar, P. K., Jain, P. K., & Tandon, P. (2023). Application of artificial intelligence techniques in incremental forming: A state-of-the-art review. *Journal of Intelligent Manufacturing*, 34(3), 985–1002. <https://doi.org/10.1007/s10845-021-01868-y>
- Peres, R. S., Jia, X., Lee, J., Sun, K., Colombo, A. W., & Barata, J. (2020). Industrial artificial intelligence in industry 4.0 - systematic review, challenges and outlook. *IEEE Access*, 8, 220121–220139. <https://doi.org/10.1109/ACCESS.2020.3042874>
- Popp, G.-P., Racz, S.-G., Breaz, R.-E., Oleksik, V. S., Popp, M.-O., Morar, D.-E., Chicea, A.-L., & Popp, I.-O. (2024). State of the art in incremental forming: Process variants, tooling, industrial applications for complex part manufacturing and sustainability of the process. *Materials*, 17(23), 5811. <https://doi.org/10.3390/ma17235811>
- Patel, J., Samvatsar, K., Prajapati, H., & Sharma, U. (2015). Analysis of variance for surface roughness produced during single point incremental forming process. *International Journal of New Technologies in Science and Engineering*, 2(3), 90–97.
- Rodríguez-Alabanda, O., Molleja-Molleja, R., Guerrero-Vaca, G., & Romero, P. E. (2019). Incremental forming of non-stick pre-coated sheets. *The International Journal of Advanced Manufacturing Technology*, 101(9), 3065–3071. <https://doi.org/10.1007/s00170-018-3150-z>
- Ramírez-Sanz, J. M., Maestro-Prieto, J.-A., & Arnaiz-González, B. A. (2023). Semi-supervised learning for industrial fault detection and diagnosis: A systemic review. *ISA Transactions*, 143, 255–270. <https://doi.org/10.1016/j.isatra.2023.09.027>
- Taherkhani, A., Basti, A., Nariman-Zadeh, N., & Jamali, A. (2019). Achieving maximum dimensional accuracy and surface quality at the shortest possible time in single-point incremental forming via multi-objective optimization. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, 233(3), 900–913. <https://doi.org/10.1177/0954405418755822>
- Triguero, I., González, S., Moyano, J. M., García, S., Alcalá-Fdez, J., Luengo, J., Fernández, A., Jesús, M. J., Sánchez, L., & Herrera, F. (2017). KEEL 3.0: An open source software for multi-stage analysis in data mining. *International Journal of Computational Intelligence Systems*, 10, 1238–1249. <https://doi.org/10.2991/ijcis.10.1.82>
- Tao, X., Ren, C., Li, Q., Guo, W., Liu, R., He, Q., & Zou, J. (2021). Bearing defect diagnosis based on semi-supervised kernel local fisher discriminant analysis using pseudo labels. *ISA Transactions*, 110, 394–412. <https://doi.org/10.1016/j.isatra.2020.10.033>
- Engelen, J. E., & Hoos, H. H. (2020). A survey on semi-supervised learning. *Machine Learning*, 109(2), 373–440. <https://doi.org/10.1007/s10994-019-05855-6>
- Wang, J., Luo, S.-w., & Zeng, X.-h. (2008). A random subspace method for co-training. In 2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence) (pp. 195–200). <https://doi.org/10.1109/IJCNN.2008.4633789>
- Wang, Y., Xu, X., Zhao, H., & Hua, Z. (2010). Semi-supervised learning based on nearest neighbor rule and cut edges. *Knowledge-Based Systems*, 23(6), 547–554. <https://doi.org/10.1016/j.knosys.2010.03.012>
- Yarowsky, D. (1995). Unsupervised word sense disambiguation rivaling supervised methods. In 33rd Annual Meeting of the association for computational linguistics (pp. 189–196). <https://doi.org/10.3115/981658.981684>
- Yaslan, Y., & Cataltepe, Z. (2010). Co-training with relevant random subspaces. *Neurocomputing*, 73(10), 1652–1661. <https://doi.org/10.1016/j.neucom.2010.01.018>
- Yu, K., Lin, T. R., Ma, H., Li, X., & Li, X. (2021). A multi-stage semi-supervised learning approach for intelligent fault diagnosis of rolling bearing using data augmentation and metric learning. *Mechanical Systems and Signal Processing*, 146, Article 107043. <https://doi.org/10.1016/j.ymssp.2020.107043>
- Yuan, J., Zhao, R., He, T., Chen, P., Wei, K., & Xing, Z. (2022). Fault diagnosis of rotor based on Semi-supervised Multi-Graph Joint Embedding. *ISA Transactions*, 131, 516–532. <https://doi.org/10.1016/j.isatra.2022.05.006>
- Yang, H., Zheng, H., & Zhang, T. (2024). A review of artificial intelligent methods for machined surface roughness prediction.

Tribology International, 199, Article 109935. <https://doi.org/10.1016/j.triboint.2024.109935>

Zhou, Y., & Goldman, S. (2004). Democratic co-learning. In IEEE international conference on tools with artificial intelligence (pp. 594–602). <https://doi.org/10.1109/ICTAI.2004.48>.

Zhou, Z. H., & Li, M. (2005). Tri-training: Exploiting unlabeled data using three classifiers. *IEEE Transactions on Knowledge and Data Engineering*, 17, 1529–1541. <https://doi.org/10.1109/TKDE.2005.186>

Liu, Z., Liu, S., Li, Y., & Meehan, P. A. (2014). Modeling and optimization of surface roughness in incremental sheet forming using

a multi-objective function. *Materials and Manufacturing Processes*, 29(7), 808–818. <https://doi.org/10.1080/10426914.2013.864405>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.