Materials Today Communications 51 (2026) 114857

Contents lists available at ScienceDirect

Materials Today Communications

journal homepage: www.elsevier.com/locate/mtcomm

A machine learning pipeline for silicon wafer warpage prediction in diamond coated wire slicing

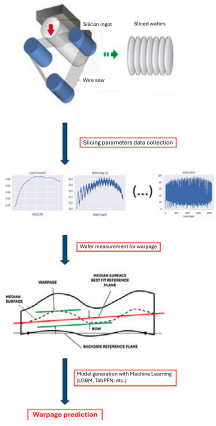
Carlo Zavattari[a,b], Dario Demaria[a], Fabrizio Bonda[a], Giovanni Iacca[b,*]

[a] MEMC Electronic Materials SpA, Novara, Italy
[b] University of Trento, Povo, Italy

HIGHLIGHTS

- We use ML to predict silicon wafer warp based on wire saw slicing parameters.
- We focus on the case of Diamond Coated Wire (DCW) slicing.
- We extensively test traditional ML methods and Tabular Prior-data Fitted Network.
- We collect a 1-year dataset of 1098 5-h cuts, for a total of 2.9M records.
- Our results provide valuable, and explainable predictions of the wafer shape.

GRAPHICAL ABSTRACT



ARTICLE INFO

Keywords:
Silicon wafer
DCW
Warpage
Machine learning
TabPFN

ABSTRACT

High-tech companies that manufacture silicon wafers are nowadays continuously engaged in the effort to produce wafers with increasing quality levels, to fulfill the ever-tightening requests from customers, typically manufacturers of microchips and electronic devices. One of the main measures of the quality of wafers is their shape, specifically their flatness (intended as the characteristic of being as similar as possible to an ideal plane), as flatter wafers guarantee more effective lithographic processing. In this paper, we consider one of the main silicon wafer production steps, that is, the wire sawing of silicon ingots into wafers, which affects wafer warpage. Focusing on Diamond Coated Wire (DCW) slicing, we develop a Machine Learning (ML) pipeline to predict warpage based on several machine parameters' measurements. We test different ML techniques, namely Linear Regression, Random Forest, Light Gradient Boosting Machine, Multi-Layer Perceptron, and the recently released Tabular Prior-data Fitted Network (TabPFN) on an in-house collected dataset consisting of 1098 5-h cuts, for a total of 2.9M records. Across the tested ML models, TabPFN provides better predictions of the average warpage, even with limited preprocessing, while being less effective at predicting the warpage standard deviation. Overall, this study represents a first attempt at using ML to predict wafer warpage based on DCW slicing machine parameters, showing the potential to enable, in future work, closed loops where warpage reduction drives the machine settings.

* Corresponding author.
Email addresses: czavattari@gw-semi.com (C. Zavattari), ddemaria@gw-semi.com (D. Demaria), fbonda@gw-semi.com (F. Bonda), giovanni.iacca@unitn.it (G. Iacca).

https://doi.org/10.1016/j.mtcomm.2026.114857
Received 13 October 2025; Received in revised form 7 January 2026; Accepted 12 February 2026
Available online 17 February 2026

1. Introduction

One of the key operations in producing silicon semiconductor wafers is slicing silicon ingots into wafers. Those wafers are then processed to make them ready to produce microprocessors and other electronic devices. Wafer slicing impacts wafer quality mainly in terms of mechanical/geometrical parameters, which are generated during the process and typically remain unchanged until the end of the line.

In this regard, the two major quality measures are (1) the *warpage* [1,2] (i.e., the wafer general deformation, see Fig. 1) and (2) the *waviness* (i.e., the short-range surface modulation, see Fig. 2), which essentially measure the flatness of a wafer. Flatness is essential for integrated circuit manufacturing, as it ensures that all of the wafer can be effectively used for lithographic processing.[1]

Warpage and waviness are generated during slicing because different forces and temperature gradients interact and deflect the slicing from a perfectly flat surface. Any deviation means a deviation from the wafer ideal plane, which in fact generates warpage and waviness. In the following, we will mainly focus on warpage, since it is the most critical parameter, as waviness somehow depends on warpage and is actually generated by it.

Any wire saw machine has a set of parameters that are monitored during the cutting process, such as the temperatures of the rotating parts, the tension and speed of the wire, the movement of the ingot into the cutting section, etc. Proper monitoring of these parameters and the ability to link them to the wafer warpage could help improve wafer quality and allow for the possibility of predicting the saw performance by analyzing the slicing parameters.

The consolidated standard process in the industry is the so-called loose abrasive technique, where an abrasive fluid is conveyed toward the silicon ingot through a mechanical support (a steel wire). In recent years, inputs from the European Community and the Environmental, Social and Governance (ESG)[2] philosophy to reduce carbon emissions and electricity usage during manufacturing steps forced the industry to introduce a different type of wire saw slicing, via fixed abrasive, where a Diamond Coated Wire (DCW)[3] is the slicing vehicle and agent (Fig. 3).

DCW cuts faster than loose abrasive and consumes less electricity. However, compared to the traditional technology based on steel wire, the wafer quality in terms of flatness achieved with DCW is inferior [3,4]. One possible way to improve wafer quality with DCW while also reducing manufacturing costs would be to work *downstream* (i.e., after slicing) to reduce the generated warpage and waviness. Typically, new manufacturing steps and tools are introduced to this aim, making the process more expensive. Another idea is to work on the source of the issue *upstream*, to prevent it from happening and/or to mitigate its effects. This would allow industries to keep the same number of tools, avoid silicon removal, and maintain all the main steps involved in the wafer production. This second approach would also guarantee better control and know-how of the slicing process, going deeper into the understanding of the mechanisms affecting the wafer warpage. The benefit, besides what has been explained above, would be the ability to generate any desired/customized wafer shape, targeting and tweaking the proper slicing settings.

This is the direction we investigate in this paper, which is a first attempt at using Machine Learning (ML) techniques to generate valuable models that link wire saw slicing parameters and wafer warpage. Recently, several researchers have used ML techniques to correlate DCW slicing with wafer quality. However, to the best of our knowledge, there is a relevant gap in the literature concerning works that consider, as the output of the ML model, the wafer warpage (which, as mentioned, is a key quality measure in the silicon semiconductor wafer industry). In fact, most of the existing works focus on other mechanical parameters, which are not connected to the warpage. We briefly summarize these works in the next section.

Therefore, our main contributions are as follows:

1. We explore several ML methods, namely Linear Regression (LR), Random Forest (RF), Light Gradient Boosting Machine (LGBM), Multi-Layer Perceptron (MLP), and the recently released Tabular Prior-data Fitted Network (TabPFN) [5].
2. We show the varying predictive power of the different models and provide promising insights into the DCW slicing process, identifying the major slicing parameters correlating with warpage. In the near future, these results may be used to control wafer quality.

The rest of the paper is organized as follows. In the next section, we briefly overview the related works, specifically focusing on DCW and ML studies (covering ML applications outside this domain is beyond the scope of this work). Next, in Section 3, we state the main objectives of the present work. Then, Section 4 describes the proposed methodology and the experiments. Section 5 presents the results and discusses the main findings. Finally, Section 6 provides the conclusions of our work.

2. Related works

Most of the current literature on DCW and applied ML is related to the photovoltaic field. A few attempts have also been made at applying ML to the slicing of semiconductor wafers and stone blocks. We summarize some of the existing works below.

2.1. DCW and ML for slicing photovoltaic silicon wafers

Some of the existing works used ML techniques to determine/predict roughness for silicon wafers sliced with DCW. It should be noted that roughness is a different concept compared with warpage, since it describes the surface finishing and not the wafer shape. It is an important parameter especially for solar wafers, since it can determine the efficiency of the cells, while, for semiconductor materials, it is marginal, since the downstream processes (like lapping, polishing) completely remove it. Warpage, instead, is more relevant for microchip manufacturing.

As an example of these works, Kayabasi et al. [6] developed a model using an Artificial Neural Network (ANN) to be applied to silicon wafers for photovoltaic applications. They considered as inputs only three parameters (spool speed, which refers to wire speed, z-axis speed, which refers to feed unit speed, and oil, which refers to coolant).

Li et al. [3] conducted a thorough study of slicing solar silicon wafers with DCW. The purpose of the study was to determine and analyze the sub-surface damage (SSD), which consists of microcracks generated underneath the silicon surface by the diamond grains' scratches. SSD is also generated by the scratching action during slicing, and, as for roughness, it is only marginally considered when slicing semiconductor wafers, since it is removed by the downstream processes.

In this study, the authors used two different methods to determine the damage (namely, a Finite Element Method and a Molecular Dynamics Model). Moreover, they analyzed the cutting forces and their correlation with the Material Removal Rate (MRR) to minimize cracks on the silicon surface.

2.2. DCW and ML for slicing semiconductor wafers

Zhao et al. [7] studied the DCW used for slicing semiconductor materials (including silicon) and explored the distribution density of abrasives and agglomeration on the surface of the wire, since this is considered an important index to evaluate its manufacturing quality (i.e., surface damage and scratches): during the slicing process, the material removal mainly depends on the scratching effect of the abrasives. The

[1] https://vitrek.com/applications/wafer-bow-and-warpage/

[2] A list of the abbreviations used throughout the paper is reported in Tables A.11 and Table A.10 available in the Appendix.

[3] https://www.nature.com/research-intelligence/nri-topic-summaries/diamond-wire-sawing-of-silicon-wafers-micro-187405

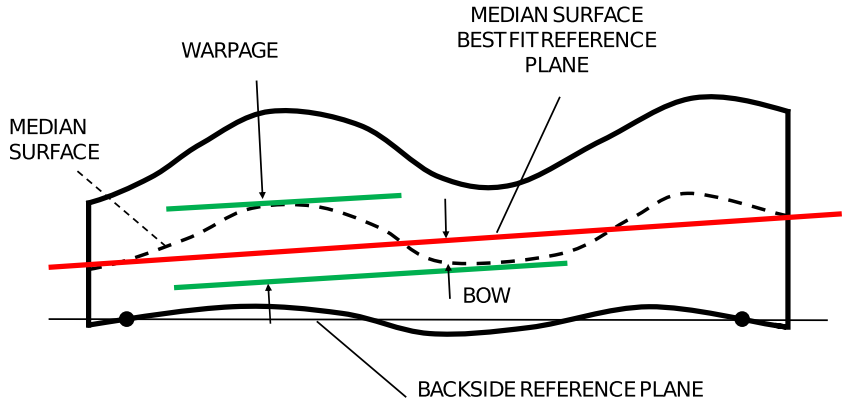


Fig. 1. Graphical representation of silicon wafer warpage: given a median ideal wafer plane (red line) and the real median plane (dotted line), the warpage is the maximum deflection of the dotted line w.r.t. the ideal one. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

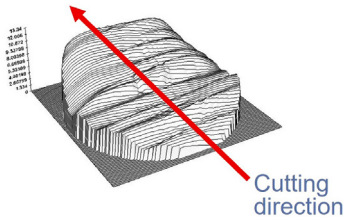


Fig. 2. Waviness visible on a wafer 3D map: it is calculated as the derivative of the peak and valley shapes along the cutting direction.

existence of agglomeration can have a negative impact on this scratching. Scratches generate roughness and SSD; as mentioned earlier, they are removed during the downstream processes, but if an abnormal slicing process occurs and generates higher roughness and SSD, they may persist after the lapping and polishing processes. In this case, wafers are rejected. The problem of diamond grain distribution uniformity on the wire and agglomeration effects is known to some DCW suppliers, who are working on their manufacturing line to prevent or mitigate these effects.

To address this problem, a backbone based on a Convolutional Neural Network (CNN) was used to extract the shape and location of objects (diamond grains) by using representation points and determining the grains' distribution on the wire surface.

2.3. DCW and ML for slicing stone and marble blocks

Zhang et al. [8] explored three ML models (generalized regression, backpropagation neural network, and support vector regression) to predict the average deflection of the DCW (wire bow) during stone machining. This deflection can significantly compromise the geometric accuracy of curved surfaces and corner regions of the sliced stone, with the possibility of rejecting the workpieces. In semiconductor silicon slicing, wire deflection also occurs, and it could be a source of wafer shape degradation when not under control. The modern multi-wire saw machines, like the ones used for our study, indirectly monitor this bow through saw parameters such as the main drive torque.

Jain and Rathore [9] developed an ANN model using certain slicing parameters as input, namely shear strength parameter cohesion and machine parameters (peripheral speed and thrust), to predict the cutting performance of a diamond wire saw, with the aim of planning the production of marble blocks and cost estimation, organizing a more efficient marble quarry and reducing cutting costs. They also compared the results of the ANN with Multiple Regression Models, and they stated that ANN is a better prediction model.

For Mikaeil et al. [10], the focus of the application of ML was instead on predicting the performance of DCW in slicing marble or stones, which means to and optimizing the wire wear-out controlling reduce the manufacturing cost.

2.4. Solution proposed in this article

In contrast to the above-mentioned works, the present work, which is focused on semiconductor silicon wafers, aims to provide ML-based models to link the wafer warpage with about twenty slicing parameters (such as bearing boxes and coolant temperatures, coolant density and flow, main drive torque, wire tension, etc.). It should also be noted that the input parameters we consider in this study are way more than those used in previous works, considering both mechanical and thermal ones.

As stated, we specifically focus on the warpage as the output, since it is an important parameter for semiconductor wafers to properly manufacture microchips. In contrast, previous works mostly focused on parameters like roughness and cutting efficiency.

Finally, concerning the model definition, we used both traditional ML models as well as a newly released ML technique (TabPFN), which, to the best of our knowledge, does not appear to have been used by previous works in this field.

3. Objectives

Given the availability of a sufficiently large database storing various saw parameters and logs related to different cutting processes, our goal is to study how wafer warpage is affected by key saw parameters and generate a model capable of predicting warpage based on the saw settings and behavior. The model should also allow for verifying whether saw parameter shifts or drifts impact wafer warpage, helping in generating troubleshooting and predictive maintenance actions without waiting for failures. This approach can be supported by the use of ML techniques. [11,12].

The main limitation of this approach is the time lag between slicing and wafer measurements, which, however, cannot be eliminated. So, it is very important to achieve computationally efficient yet precise predictions (which are the focus of the present study) of the wafer warpage just by analyzing the saw parameters in real-time; in turn, these predictions can be used to change the process settings on-the-fly to reduce the wafer warpage without waiting for the wafers to be measured (which will be the focus of future studies).

The envisioned generation of this closed loop between the saw parameters and the "as-cut" (i.e., post-slicing) wafer shape properties (warpage and waviness) is shown in Fig. 4.

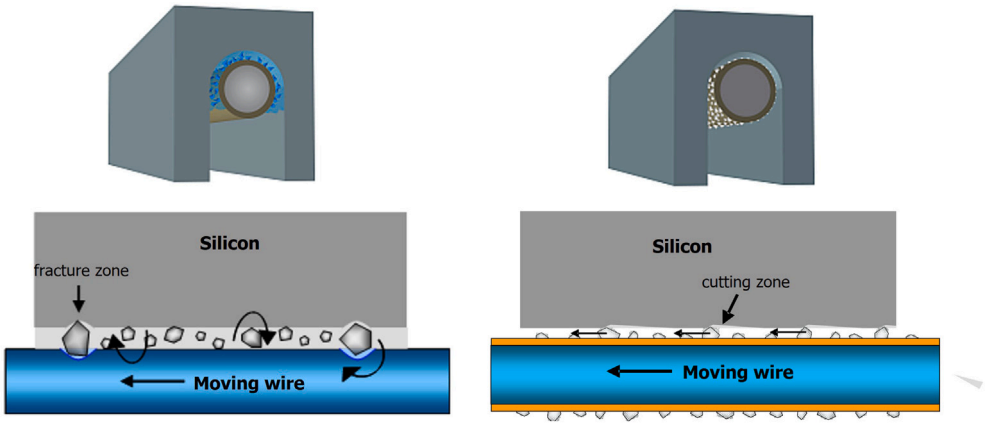


Fig. 3. Slicing process using loose abrasive (left) and fixed abrasive (right). Loose abrasive cuts by rolling, while fixed abrasive does so by sliding.

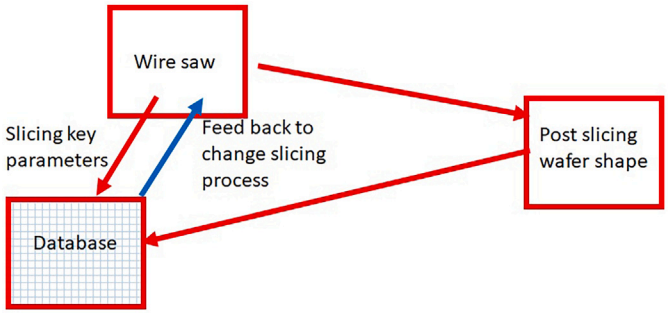


Fig. 4. Closed loop between the saw parameters and the as-cut wafer shape properties.

4. Methodology and experiments

The methodology to generate models capable of predicting the wafer warpage based on saw parameters focuses on the steps described next.

4.1. Data acquisition

The first step is the creation of a database containing the records of a set of measurements taken from the slicing process.

We collected, in total, data from 1098 cuts from a single machine. For every cut, the dataset contains records taken during the cut of a set of measurements (more than 50), such as temperatures, wire speed, wire tension, motor torque, coolant pressure, etc. The recording period is seven seconds,[4] while the total cutting cycle takes between 5 and 10 h, depending on the tool and type of silicon to be sliced. In our case, the cutting time was set to 5 h. In total, we collected about 2650 records per cut. The samples from each cut were stored in a single CSV file. An example of data collected during a cut is shown in Table 1. A detailed explanation of each measured parameter is provided in Table A.12 reported in the Appendix.

The CSV files were then collected into a database, containing a total of about 2.9 M records, which were then used as input for the subsequent steps.

4.2. Pre-processing

Some of the parameters were deemed significant from an engineering point of view, or thought to impact our output of interest (i.e., the warpage), while others were considered to have no or limited influence on them. This data screening was performed based on previous silicon slicing experience, after which the non-impacting parameters were removed from the analysis. Among more than 50 parameters, we found about 20 of them with a potential impact on warpage. So, for each cut, we considered only the time series associated with those 20 parameters, and for the subsequent steps, we restricted the analyses only to those time series. For example, “slurry temperature”, “slurry flow”, and “wire tension” were kept as significant, while “wire breakage and ground signal” or “cooling water valve position” were not considered in the analyses.

Furthermore, we decided to remove the warm-up phase and the wafer’s unloading phase from each time series: the first phase is needed to “thermalize” and stabilize the machine before starting cutting, so its length depends on the previous conditions rather than the slicing configuration *per se*; the latter occurs at the end of the cut and as such it does not contribute to the warpage.

As output data, we considered warpage measurements of sample wafers from each cut. Specifically, for each cut, a certain number of wafers (taken from fixed positions of the ingot) were taken and measured for warpage. The tool used to measure warpage was a single-station system using two non-contact capacitive probes, a standard tool utilized in the semiconductor industry for 200 mm as-cut wafers [2].[5] The tool explores the whole wafer surface during the measurements. More than 2000 records were taken per wafer to obtain the final warpage value.

An example of warpage measurements is shown in Table 2, organized as an output matrix. The first column is simply the wafer ID number, indicating the position of the wafer inside the cut; the sequence is repeated since the shown cut was composed of two ingots. The second column indicates the cut ID. The third column indicates the warpage measurement.

As can be noticed by looking at the data, there is no direct link between rows in the input matrix, which are time-based, thus varying in a direction perpendicular to the ingot axis, and the rows in the output matrix, which are by wafer number (position along the ingot), and vary

[4] In this period, the cutting position change is about 100–120 μm, less than a wire diameter (which is close to 150 μm). This makes measurements accurate enough to evaluate local deformation of the warpage to a level requested by the customers.

[5] As the tool is used 24/7 in production, a calibration check based on a master wafer, generated following the semiconductor industry standards, hence with a certified warpage value, was used to validate the accuracy of the tool three times per day. In terms of accuracy, the calibration readings were within 1 μm of the master warpage value (which is around 10–12 μm).

Table 1
Example of slicing parameters collected during a cut. Note that not all columns are shown, due to space limitations.

Time	ABS_F [mm]	REL_V [mm]	v_Fed [mm/min]	v_Wir [mm/min]	N3201 [N]	N3221 [N]	B4003	B4007	B4001	B4005	B1771	B1773
02:30:19	287.924	25.001	−0.49	0.00	2.66	3.08	17.33	17.26	17.53	17.16	15.90	15.81
02:30:26	287.873	24.948	−0.49	−9.59	24.72	25.14	17.33	17.33	17.53	17.16	15.75	15.72
02:30:33	287.830	24.906	−0.49	−15.00	24.78	25.18	17.33	17.33	17.53	17.16	15.49	15.31
02:30:40	287.759	24.836	−0.49	−15.00	24.83	25.18	17.33	17.36	17.53	17.16	15.23	15.02
02:30:47	287.711	24.788	−0.49	−15.00	24.83	25.18	17.33	17.36	17.53	17.16	15.00	14.78
02:30:54	287.647	24.724	−0.49	−15.00	24.89	25.18	17.33	17.36	17.53	17.16	14.81	14.60
02:31:01	287.574	24.652	−0.49	−15.00	24.94	25.20	17.33	17.36	17.53	17.16	14.66	14.44
02:31:08	287.473	24.550	−0.49	−15.00	25.04	25.34	17.36	17.36	17.53	17.16	14.53	14.29
02:31:15	287.360	24.436	−0.49	−15.00	25.14	25.36	17.36	17.36	17.53	17.16	14.41	14.14
02:31:22	287.309	24.386	−0.49	−15.00	25.20	25.40	17.36	17.36	17.53	17.16	14.30	14.00
02:31:29	287.246	24.322	−0.49	−15.00	25.25	25.42	17.36	17.36	17.53	17.16	14.20	13.87
02:31:36	287.175	24.250	−0.49	−15.00	25.27	25.42	17.36	17.36	17.53	17.16	14.12	13.75
02:31:43	287.099	24.174	−0.49	−15.00	25.30	25.42	17.36	17.36	17.53	17.16	14.05	13.64
02:31:50	287.018	24.093	−0.49	−15.00	25.32	25.42	17.36	17.36	17.53	17.16	14.00	13.54
02:31:57	286.930	24.005	−0.49	−15.00	25.35	25.42	17.36	17.36	17.53	17.16	13.94	13.44
02:32:04	286.848	23.923	−0.49	−15.00	25.36	25.42	17.36	17.36	17.53	17.16	13.89	13.35
02:32:11	286.764	23.839	−0.49	−15.00	25.38	25.42	17.36	17.36	17.53	17.19	13.85	13.27
02:32:18	286.698	23.774	−0.49	−15.00	25.41	25.42	17.36	17.36	17.55	17.19	14.74	14.52
02:32:25	286.799	23.874	−0.49	−15.00	25.42	25.42	17.36	17.36	17.58	17.19	14.74	14.52
02:32:32	286.734	23.811	−0.49	−15.00	25.24	25.24	17.65	17.79	17.58	17.19	14.70	14.52
02:32:46	286.734	23.811	−0.49	−15.00	25.24	25.24	17.65	17.79	17.58	17.19	14.70	14.52
02:32:53	286.685	23.762	−0.49	−15.00	24.70	24.75	17.70	17.84	17.60	17.19	14.70	14.49

Table 2
Example of warpage measurements.

WAFER	CUT ID	warpage [μm]
1	H16_1_002	16.943
2	H16_1_002	9.786
3	H16_1_002	12.223
4	H16_1_002	16.324
5	H16_1_002	14.966
6	H16_1_002	9.821
7	H16_1_002	11.768
8	H16_1_002	11.712
9	H16_1_002	12.374
1	H16_1_002	21.659
2	H16_1_002	21.152
3	H16_1_002	19.170
4	H16_1_002	21.518
5	H16_1_002	18.707
6	H16_1_002	20.874
7	H16_1_002	23.017

in the direction of the ingot axis. As a solution to this issue, we summarized the output data at the cut level, calculating the average (AVE) and the standard deviation (STDEV) of the warpage measurements of the wafers belonging to each cut. Therefore, we derived for each cut two associated output values (AVE and STDEV), hereafter referred to as *targets*, according to common jargon in ML-based regression.

4.3. Exploratory data analysis and feature engineering

In order to effectively use most ML algorithms, we have built a set of derived features from each of the 20 time series considered for each cut; for a sample cut, see Figs. 5 and 6. For feature extraction, we leveraged the `Catch22` Python library [13], generating 20 synthetic time series indicators from each of the input time series, to which we additionally added the average and standard deviation of each input time series. Eventually, we derived a set of $(22 + 2) \cdot 20 = 480$ features for every cut.

Eventually, we ended up building an input matrix made up of 1098 rows (each one corresponding to a separate cut) and 480 columns (one column for each of the `Catch22`-derived features, see Table 3 for a sample of the input matrix).

This way, we could obtain input and output matrices with rows representing cuts, thus being able to associate the set of input data associated with each cut with the two outputs AVE and STDEV of the warpage.

4.4. Assessment of empty and `NAN` values

As a first step in the exploratory data analysis, we evaluated the columns with empty or `NAN` values. These correspond to pairs (cut, time series) for which (part of) the `Catch22`-derived features could not be calculated. We did not focus on the reasons behind these missing calculations, considering the lack of a value as a possible indicator of an anomalous time series.

In order to analyze possible specific patterns between the `NAN` values and the target values, we identified the cuts whose target values were extreme. For this purpose, we first identified families of cuts with high and low target values. These have been defined, respectively, as follows:

- high target cuts: cuts appearing as high outliers in Fig. 7, i.e., cuts whose target value is greater than $Q3 + 1.5 * IQR$;
- low target cuts: as no low outlier appears in Fig. 7, we defined a lower threshold as $0.8 * Q1$. Cuts whose value is lower than that are considered low target cuts.

Based on these definitions, no specific pattern in `NAN` values distribution was observed. Therefore, we dropped all the columns with more than 20% of empty values, keeping 446 remaining features.

4.5. Feature cleaning

We have focused on columns with fewer than three identical values. These columns correspond to `Catch22`-derived features that take one or two unique values among all cuts. This means that, for some specific time series, the calculated features yield the same results, which do not add any information to the analysis. We decided to remove all these features. After this step, we retained 428 cleaned features.

4.6. Train-test split and data imputation

According to ML best practices, we randomly split the whole dataset into a training and a test set (containing respectively 80% and 20% of cuts), filling the remaining `NAN` values with the median calculated over the training set (after some preliminary attempts, we eventually decided to replace the empty values with the median, instead, e.g., of dropping

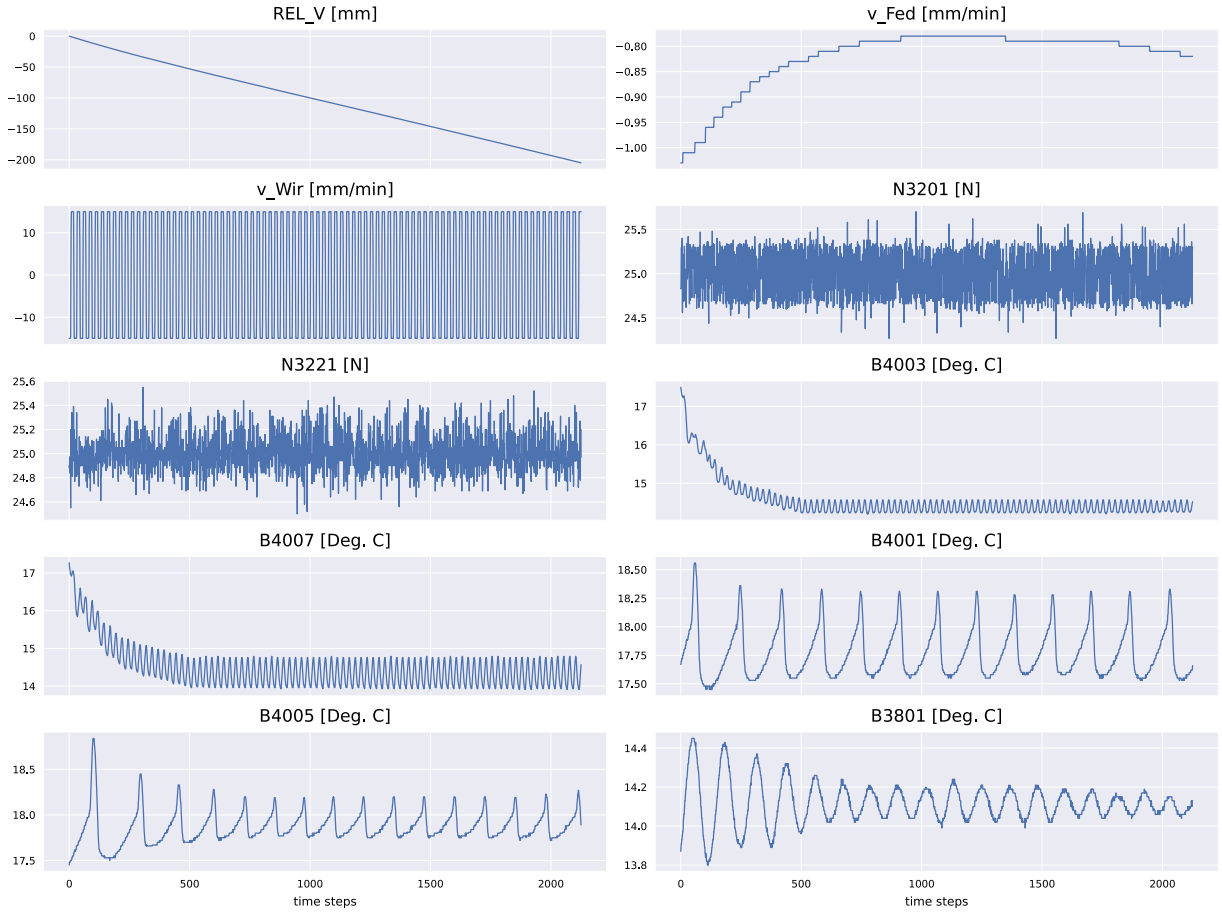


Fig. 5. Time series of selected features (1-10) for a sample cut, after removal of warm-up and final unload steps. The x-axis indicates the displacement of the ingot along the x direction, in mm.

the columns or the cuts; we have observed how this simple imputation strategy, which is common in data preparation, is sufficient for our purposes). There are other techniques to fill up the `NAN` values (see, for example, [14]), yet we considered the median a valuable and easy system for this purpose.

4.7. Feature normalization and correlation analysis

As a last step before evaluation, we have normalized the features and calculated the linear correlation coefficients with both targets (AVE and STDEV). Normalization has been done according to the values calculated on the training set only, to avoid any leakage issues. Tables 4 and 5 show the most correlated features with the AVE and STDEV targets.

4.8. Evaluation

The evaluation phase aims at deriving from the data an empirical relationship between the input features (X_i) and a target value (y_j). This relationship is inferred by using different ML algorithms. The ML task to tackle is a regression one, as we are attempting to predict continuous outputs. As we have two independent targets (AVE and STDEV), we trained independent models for each target.

There are different metrics suitable for regression tasks, such as Root Mean Square Error (RMSE), Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and coefficient of determination (R^2), which analyze the model's predictions from different perspectives. Given the range of target values, for the present analysis, we use MAPE and R^2 since MAPE is normalized with respect to actual target values.

Models have been trained on the training set and tested on the test set to evaluate their generalization capabilities.

We started considering **four widely-adopted traditional ML algorithms** for tabular data, namely:

- Linear Regression (LR)
- Random Forest (RF)
- Light Gradient Boosting Machine (LGBM)
- Multi-Layer Perceptron (MLP)

Detailed results for the best models can be found in Section 5.1.

Ultimately, we decided to include in our evaluation the recently introduced **Tabular Prior-data Fitted Network (TabPFN)** model [5], one of the latest approaches representing a family of models that use In-Context Learning (ICL) of foundational models pretrained on large (mostly, synthetically generated) sets of tabular data to perform classification or regression [15–18]. This model, in particular, has been claimed to outperform state-of-the-art gradient boosting models such as LGBM. While the inference time and memory footprint of this model are larger than those of traditional ML models, it offers various advantages, *in primis*, the fact that it does not require training and that it is essentially parameter-free. Moreover, another advantage is that, with this model, most of the data pre-processing and cleaning steps required for the traditional ML algorithms can be skipped. In detail, we tested the TabPFN on our data with the following pre-processing steps:

1. after having removed the `Catch22`-derived features with more than 20% of empty values (i.e., by leaving 446 columns in the input matrix);

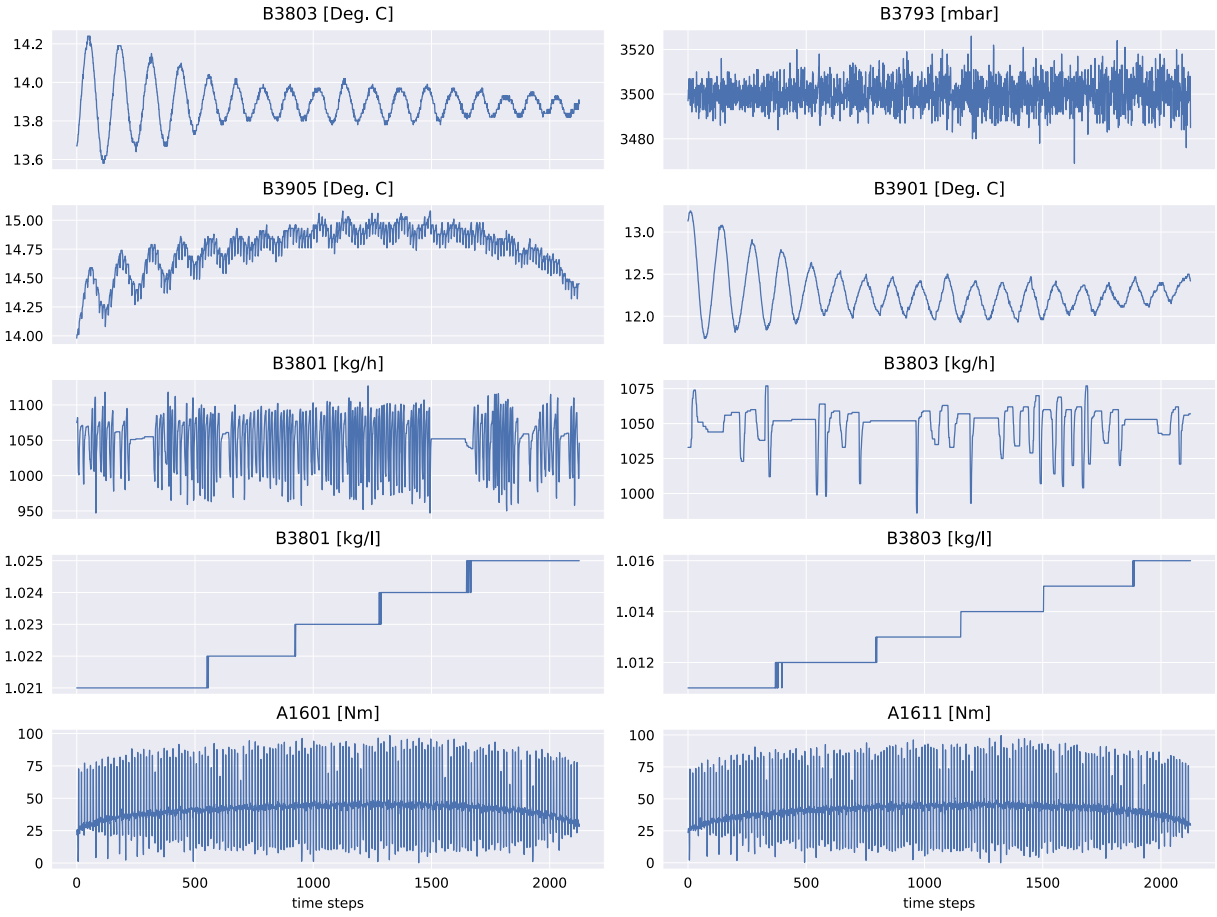


Fig. 6. Time series of selected features (11-20) for a sample cut, after removal of warm-up and final unload steps. The x-axis indicates the displacement of the ingot along the x direction, in mm.

Table 3
Sample of the input data, showing the first cuts and `Catch22`-derived features.

ID	mode_5 [mm]	mode_10 [mm]	stretch_decreasing [mm]	outlier_timing_pos [mm]	outlier_timing_neg [mm]	acf_timescale [mm]	acf_first_min [mm]	centroid_freq [mm]	low_freq_power
H16_1_002	2.00	13.50	1281.0	−0.91	0.56	562.0	2048.0	4746.86	0.0015
H16_1_003	2.00	13.50	1277.0	−0.91	0.56	561.0	2048.0	4751.54	0.0015
H16_1_004	2.00	13.50	1274.0	−0.91	0.56	559.0	2048.0	4755.57	0.0015
H16_1_005	2.00	13.50	1280.0	−0.91	0.56	562.0	2048.0	4748.01	0.0015
H16_1_007	1.97	13.47	1280.0	−0.91	0.56	562.0	2048.0	4747.37	0.0015
H16_2_165	−98.75	−3.13	1124.0	−0.70	0.50	490.0	1618.0	3807.17	0.0015
H16_2_166	−98.76	−3.14	1123.0	−0.97	0.50	490.0	1618.0	3803.20	0.0015
H16_2_167	−98.75	−3.13	1123.0	−0.97	0.50	490.0	1618.0	3806.23	0.0015

2. after having removed the `Catch22`-derived features with less than 3 unique values (i.e, by leaving 428 columns in the input matrix);
3. after having filled the remaining `NAN` values with the median calculated over the training set.

Specifically, we tested a model where only the preprocessing step (1) was applied; another where steps (1) and (2) were applied; and a final one where all steps (1)-(3) were applied. Detailed results for each attempt can be found in Section 5.2.

Table 6 shows a summary of the models tested, along with their main parameters.

Table 7 shows a rough summary of the methodology with some reference numbers and comments. Note, though, that certain property details, e.g., some saw parameters, warpage values, and correlations, are not disclosed.

5. Results and discussion

Here we show the most relevant results obtained during the evaluation phase.

5.1. Results of traditional ML algorithms

As expected, the best-performing ML algorithms for tabular data are tree-based algorithms, such as LGBM and RF, which are known to be particularly effective on regression tasks on such kinds of data [19–21].

For each of the two targets, we first identified the best model, which resulted in LGBM and RF, respectively, for the AVE and STDEV targets. For each target, we then used the best-performing model to identify the most important features (based on both the highest model-specific importance metrics and the highest correlation coefficients). Eventually,

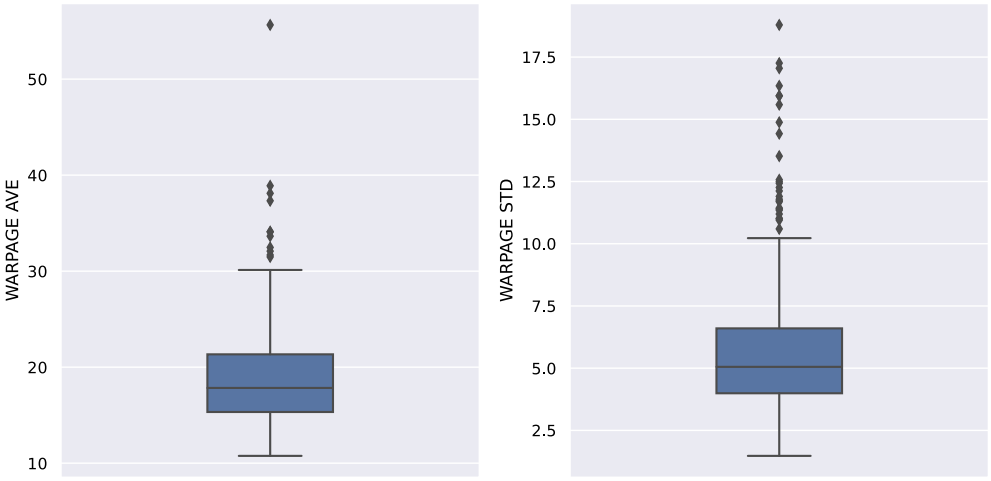


Fig. 7. Distribution of AVE and STDEV warpage values across all the analyzed cuts.

Table 4
Features with the lowest (top) and highest (bottom) correlation with AVE target values.

Feature	Correlation
B3901 [Deg. C]_mean	−0.26
B4001 [Deg. C]_rs_range	−0.23
B4001 [Deg. C]_acf_first_min	−0.22
B3901 [Deg. C]_mode_10	−0.21
A1611 [Nm]_std	0.22
B4001 [Deg. C]_mode_10	0.22
B3905 [Deg. C]_mean	0.23
B3905 [Deg. C]_mode_10	0.23

Table 5
Features with the lowest (top) and highest (bottom) correlation with STDEV target values.

Feature	Correlation
B3901 [Deg. C]_mean	−0.29
B3901 [Deg. C]_mode_5	−0.27
B3901 [Deg. C]_mode_10	−0.26
B3905 [Deg. C]_entropy_pairs	−0.25
B3905 [Deg. C]_mean	0.32
B3905 [Deg. C]_centroid_freq	0.33
B3905 [Deg. C]_std	0.32
B3905 [Deg. C]_mode_10	0.32

Table 6
Summary of the tested ML models with their parameterization.

Model	Details
Linear Regression (LR)	N/A
Random Forest (RF)	- num. estimators = 100 - maximum depth = 4 - minimum samples per leaf = 20 - minimum samples per split = 20 - maximum features per split = 50%
Light Gradient Boosting Machine (LGBM)	- maximum training round = 20,000 - early stopping = 100 rounds
Multi-Layer Perceptron (MLP)	- 2 hidden layers (resp. 50 and 25 neurons) - RELU activation function
TabPFN	N/A

the same best-performing model was retrained on the same input data, this time considering as input features:

Table 7
Summary of the methodology flow.

	Type	Value
Input data	Logging period	Mar. 2023 – Sept. 2024
	Number of cuts	1098
	Number of rows in the dataset	>2.9 M
	Number of significant (for warpage) slicing parameters	20
Output data	Wafer warpage	5–30 µm
Data analysis	Time series indicators	`Catch22`-derived
	Model types	- LR - RF - LGBM - MLP - TabPFN
	Regression metrics	- MAPE - R^2

Table 8
Results of the traditional ML algorithms on the test set for the AVE target (top) and the STDEV target (bottom). For each target, we also report the results of the best-performing model after retraining on the most important features (see the text for details).

Model	MAPE	R^2
LR	≫ 100%	< 0
RF	16.1%	0.218
LGBM	16.1%	0.227
MLP	18.9%	< 0
LGBM(retrained)	15.7%	0.248
RF	28.8%	0.108
LGBM	29.4%	< 0
MLP	3.52%	< 0
RF(retrained)	2.84%	0.120

- the top-20 most important features (evaluated according to the specific selected model); these features were chosen, respectively, using the split gain method for LGBM and the impurity-based feature importance for RF;
- the features with correlation coefficients > 0.15 or < −0.15 (thresholds chosen empirically based on the maximum correlation coefficient found for each target).

The subset of the best features accounts for 80 features for LGBM and 67 features for RF models, out of 428 initial features. This step was

taken to verify if feature selection can help with training. Table 8 shows a summary of the results of all the tested ML models for both targets.

The best model was found to be the LGBM, related to the AVE target, yielding a MAPE equal to 15.7% and an R^2 value of 24.8%. The latter result indicates that about 25% of the variation of the target is explained by the model. The MAPE value, instead, indicates that the model predictions, on average, are about 15.7% accurate. Visually, this can be appreciated in Fig. 8, where the scatterplot between actual and predicted values is reported, for each of the two targets, together with the 1-to-1 reference line (intuitively, the more the points align with the reference line, the more accurate the prediction model).

We then used the best model for each of the two targets to identify the most relevant features. A list of the 20 top-scoring features is provided in Fig. 9, respectively for LGBM (left) and for RF (right).

Finally, we further explored the interpretation of the MAPE metric for the case of the best model, namely LGBM, on the AVE target. We recall that the MAPE is defined as follows:

$$MAPE = \frac{1}{n}\sum_{i=1}^{n} APE_i$$

$$APE_i = \left|\frac{\hat{y}_i - y_i}{y_i}\right| * 100\%$$

where $\hat{y}_i$ and y_i are, respectively, the i-th estimated and actual values, and APE_i is the Absolute Percentage Error (APE) for the i-th value. We investigated the distribution of APE_i across the test data, which can be seen in Fig. 10.

Although, as seen in Table 8, the achieved overall MAPE was about 15.7%, we could observe a specific pattern in the individual values of APE_i: they tend to take lower values as more data are collected. This tendency becomes even clearer if we observe the kernel density estimation (KDE) of the AVE target values of the test set (see Fig. 11). Focusing on the Inter-Quartile Range only (Fig. 10, bottom), we calculated a $MAPE_{|IQR|}$ of about 10.1%. If this observed reduction of APE in the regions with the highest data density is not a pure statistical coincidence, we may expect that increasing the data density by collecting future data could further improve the model performance.

5.2. Results of TabPFN

As discussed earlier, TabPFN leverages ICL to learn complex association rules between input and output. It has been pretrained on millions of synthetic datasets in order to cover a very wide scope of possible rules. In our evaluation, we tested the TabPFN model using the `Catch-22`-derived features after the three different levels of feature pre-processing discussed above. The results for the AVE target are shown in Table 9.

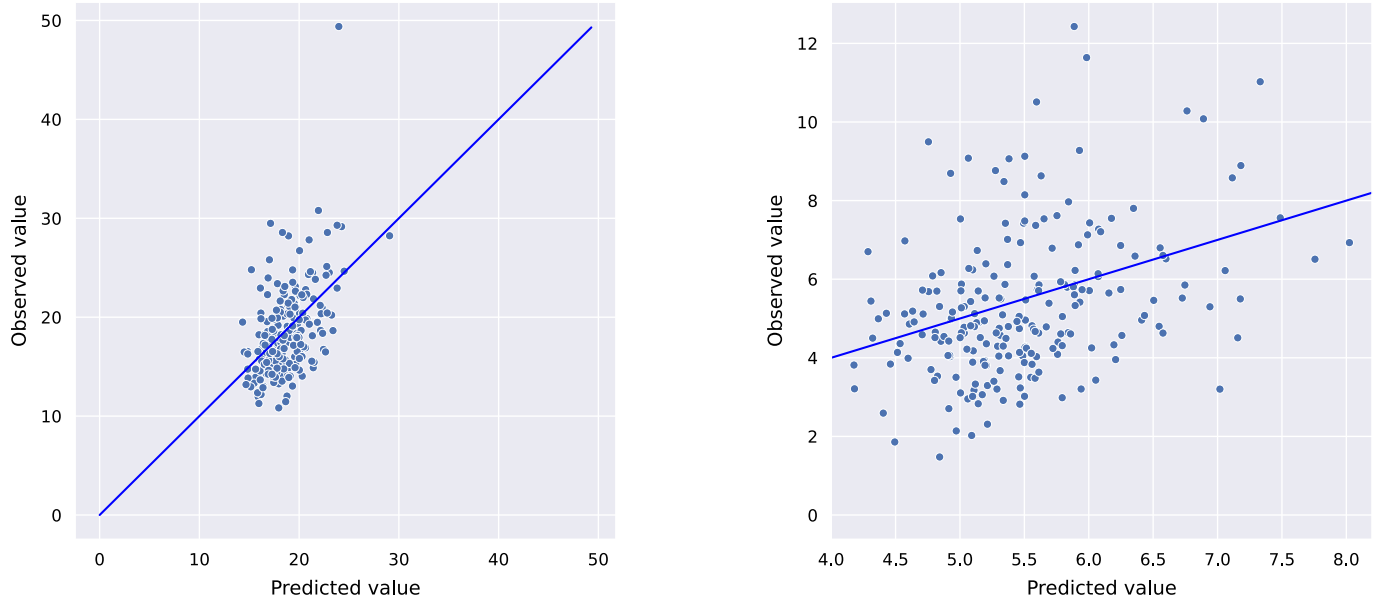


Fig. 8. Observed vs. Predicted values for LGBM on the AVE target (left) and for RF for the STDEV target (right).

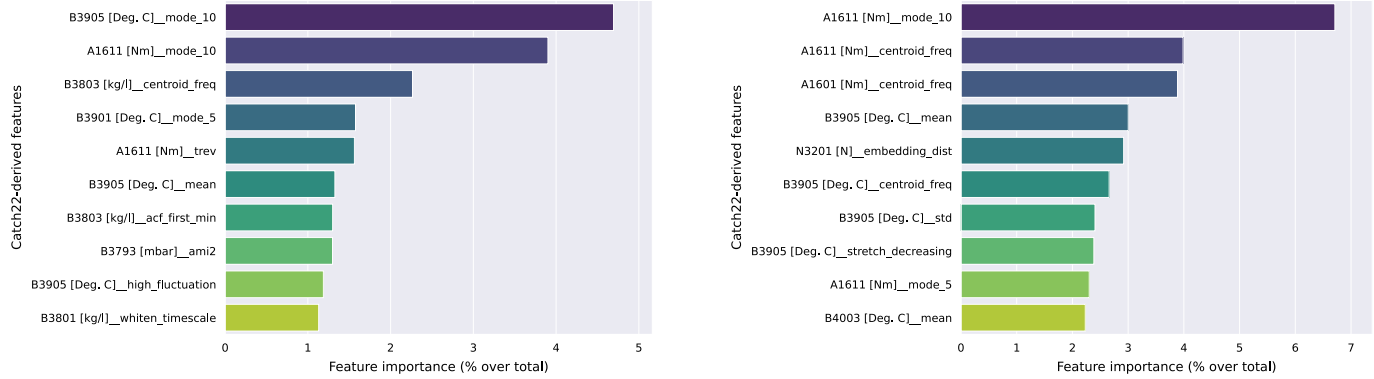


Fig. 9. Feature importance for LGBM model on the AVE target (left) and for RF model for the STDEV target (right).

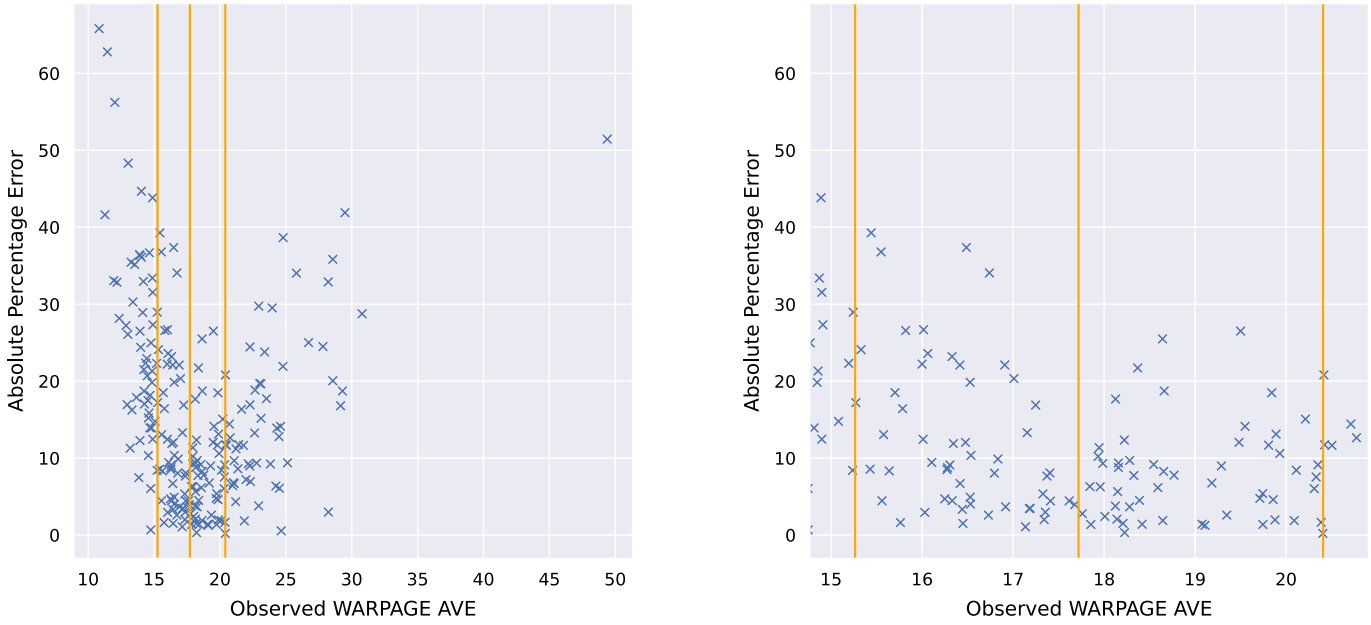


Fig. 10. Absolute percentage error for LGBM on the test set data on the AVE target (on the right, zoom on the inter-quartile range).

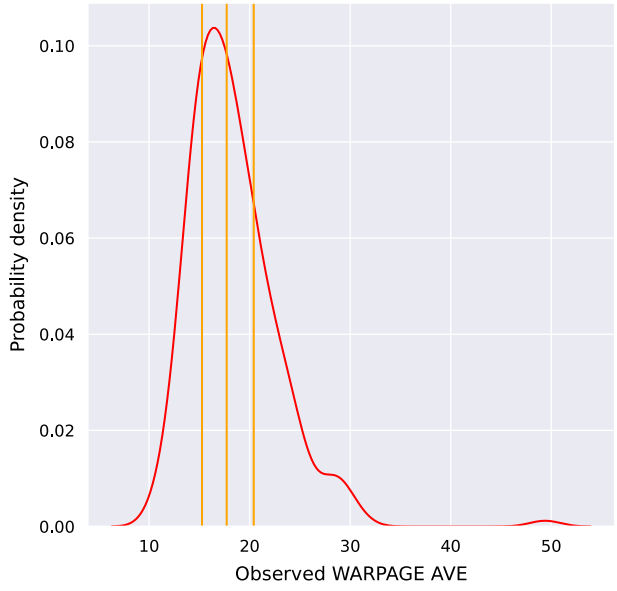


Fig. 11. KDE distribution for LGBM on the test set data on the AVE target, with inter-quartile range superimposed.

Table 9
Results of TabPFN on the test set for the AVE target.

Pre-processing level	MAPE	R^2
Removal of `Catch-22`-derived features with more than 20% of empty values	14.8%	0.358
Removal of `Catch-22`-derived features with less then 3 unique values	14.7%	0.375
Fill of empty values with median	14.7%	0.374

Analogous attempts for the STDEV target all returned negative R^2 values. Therefore, they are not reported in detail and are not discussed in the following.

From the table, we can infer the following main outcomes:

1. TabPFN outperformed the best identified ML algorithm for the AVE target prediction (i.e., LGBM) even when trained on a dataset with very little pre-processing.
2. After removal of features with less than 3 unique values, the R^2 metric achieved the highest value.
3. No improvement was achieved with further pre-processing.

Fig. 12 shows the fitting capability of the TabPFN model visually (note that only a single outlier is observed), in an analogous way to what has been done for the traditional ML models.

At last, we investigated the interpretability of the TabPFN model. This is a crucial aspect as, like any other model based on deep learning, TabPFN behaves as a *black box* model. In other words, it is not trivial to understand how the model response is produced, as the inner model's working is basically hidden. However, it is possible to get an idea of the most relevant features by calculating, for instance, the SHAP (Shapley Additive Explanation) values [22]. This interpretability method is available within the TabPFN extension library.[6] Intuitively, SHAP values represent the contribution of each feature to the response of the model: e.g., a positive SHAP value indicates that, for that data point, that feature is pushing the output toward values greater than the baseline model prediction, the opposite being true for negative SHAP values.

In Fig. 13, we show an example of SHAP values calculated on a random subset of the test dataset for the TabPFN model tested after removal of `Catch-22`-derived features with less than 3 unique values. In particular, the figure shows both the extent and the type of the contribution of the top ranking features to the prediction (positive or negative): each dot represents the SHAP value of a specific feature of a specific cut; the highest the absolute value, the highest the impact of that feature for predicting that point target; the redder the dot, the highest the value of the feature. As a consequence, when reddish dots are mainly placed on positive SHAP values, that feature's correlation with the target is mainly positive (as the feature increases, the target increases); on the other

[6] https://github.com/PriorLabs/tabpfn-extensions

Fig. 12. Actual vs. predicted values for TabPFN on the AVE target.

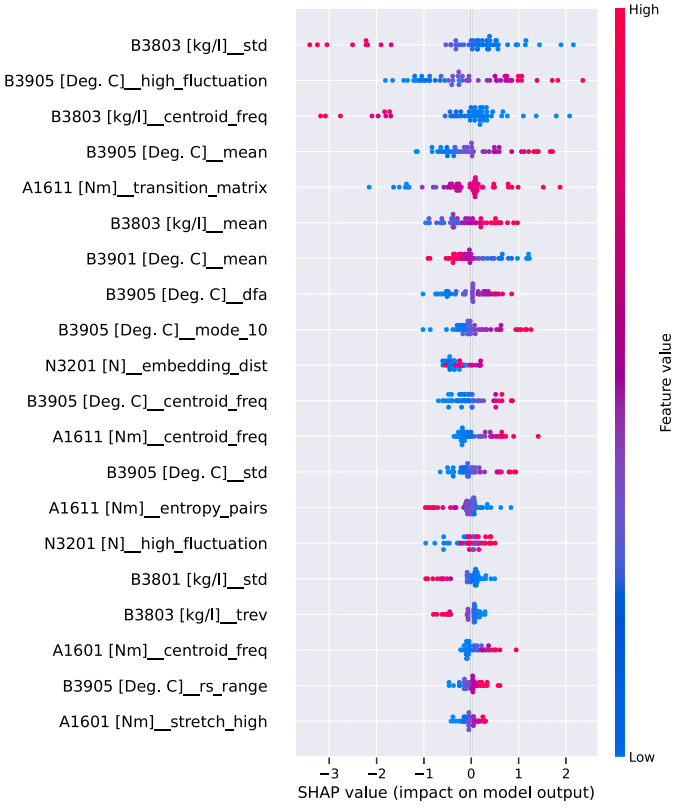


Fig. 13. SHAP values (highest ranking features).

hand, when blueish dots are mainly placed on positive SHAP values, that feature's correlation is mainly negative (as the feature increases, the target decreases).

5.3. Physical interpretation and discussion

During slicing, the relative movement of the DCW and the ingot creates friction and abrasion of silicon, inducing alterations in the wire saw machine: the generated heat diffuses along the silicon ingot, the wire guides supporting the wire, the feed unit, etc. The saw undergoes stresses that induce radial and axial forces. Thus, both heat and mechanical forces tend to deflect the wafers from a perfectly flat plane: this is called warpage.

As shown in Fig. 14, the involved forces can be divided into three groups: vertical radial, horizontal radial, and axial.

Since warpage is essentially an axial deformation, the projection of the resultant of all the forces along the axial direction can provide an indication of what the resulting wafer warpage will be.

As mentioned earlier, using the traditional ML techniques, the parameters with the highest correlation with warpage were found to be the main drive torque (recorded as time series A1601 and A1611) and some temperatures, mainly the coolant ones (B3901, i.e., the temperature after the heat exchanger, and B3905, i.e., the temperature of the coolant after slicing). The analysis using TabPFN highlighted another parameter correlated with warpage: coolant density.

The main drive torque is an indication of the friction (or abrasion force) generated during cutting, which increases the ingot temperature and expansion, as the machine has to apply a torque to the main drive to keep the wire speed at a desired value to continue cutting. An increase in this torque means an abnormal increase in the friction, which indicates that it is more difficult to slice the wafers (see Table 4, where the torque standard deviation is one of the most correlated parameters with warpage average). This may result in a higher wafer deformation (higher warpage).

Friction is also responsible for another phenomenon: diamond is much harder than silicon, but during slicing, the wire coating degrades, and the cutting ability of the new wire (inlet side) is higher than that of the used wire (outlet side). In fact, during cutting, the downward movement of the feed unit generates a wire bow, and the wire wear-out between the inlet and outlet sides induces a difference in the wire deflection. This difference generates an unbalanced situation between the two sides in terms of applied force, yielding an axial force pushing toward the used wire side. This force is one of the reasons why, in general, the warpage of the wafers sliced in the new wire section is lower than that of the ones sliced in the used wire section, since in the first section, this force is counteracting the heat force, while in the other section, they sum up.

Moreover, the increase in temperature due to friction expands the silicon ingot in all directions, including the axial one. The expansion is higher at the two ends of the ingot, compared with the center, and it is responsible for the deflection from a straight line during the wafer generation. This force varies during slicing because of the variation in

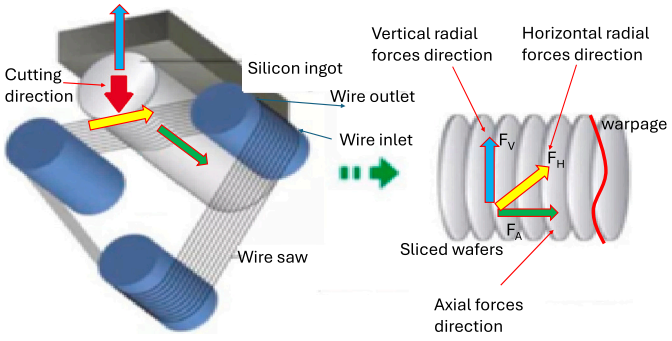


Fig. 14. Sketch of a wire saw with a schematic indication of how warpage is generated due to forces acting on the ingot.

the delta temperature reached by the ingot and the change in the friction force, due to the change in the total silicon surface to be sliced, the ingot being round. More details about the importance of friction and the resulting heat generation are explained and demonstrated in [23]. This is also the confirmation that our models (mainly TabPFN, but also LGBM) link the appropriate parameters to warpage.

As for what concerns the coolant temperatures (B3901 and B3905 in Table 4), these are also strictly connected to the friction, and they can be used as an indication of abnormal behavior during slicing. It is intuitive to understand that if extra friction occurs, more heat is generated, hence the coolant temperature after slicing (B3905) increases, so it needs more cooling to bring it back to the set/desired value. This means the coolant temperature after going through the heat exchanger (B3901) will also depend on the generated friction: the higher the friction, the lower the outcome temperature. In fact, with SHAP analysis, we found a positive correlation between warpage and B3905 (coolant temperature out of the slicing location), and a negative correlation with B3901 (coolant temperature out of the heat exchanger). This dynamic is clearly demonstrated by the wavy behavior of both the temperatures (see Fig. 6), which is due to the PID settings of the saw machine, and it is designed in such a way to keep the coolant inlet (the one used for slicing) temperature as constant as possible, considering the wire moves in a pilgrim mode, and, during the wire direction change, the speed is reduced to zero.

Other monitored temperatures also appeared to have an effect on warpage, such as that of the bearing box (B4001), although with a lower correlation compared to the coolant temperatures. An existing correlation between warpage and bearing box temperature had already been explored in previous work [24], where it was found that it is possible to control warpage during cutting just by changing the bearing box temperature settings and profiles. However, that previous study could be used only to fine-tune the slicing recipe, but not to predict the effects of slicing parameters on warpage.

Finally, the coolant density (B3803 in Fig. 13) also had a high correlation with warpage, as identified by TabPFN. The coolant density is an indication of how much silicon kerf is generated during cutting; accumulation of swarf occurs because of slicing, but if any abnormal accumulation happens, it can result from an abnormal kerf removal due to the variability of diamond grain distribution on the wire or a fluctuation in grain size range, even within the supplier's specification.

Beyond these aspects, other forces contribute to determining the warpage, and the resultant of the axial forces also depends on the ingot position, increasing from the new wire side (inlet) to the used wire side (outlet). An example of such variation is shown in Fig. 15, where the resultant of the axial forces has been determined for three different wire conditions (inlet, central section, and outlet).

Of note, the warpage displays a similar trend as the axial force resultant; in other words, the warpage along the cutting direction correlates with the force resultant along the same direction. Fig. 16 shows an example of a typical warpage profile.

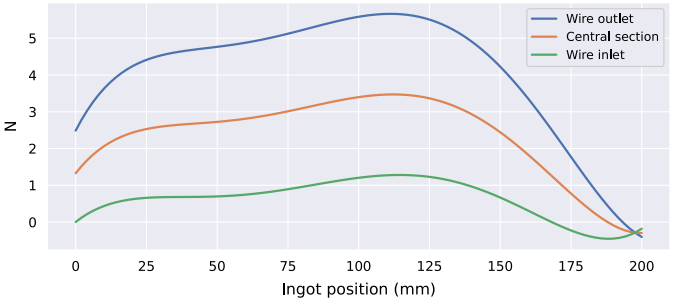


Fig. 15. Axial force resultant applied along the cutting direction at different ingot locations.

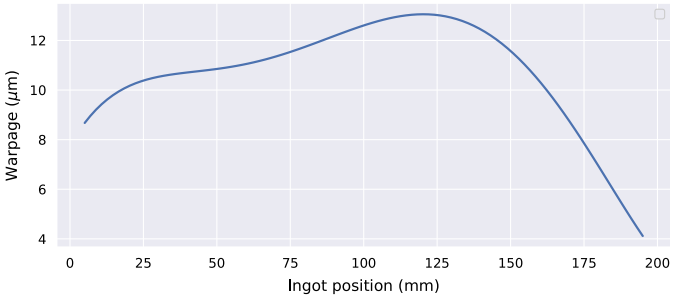


Fig. 16. Warpage cross-section along the cutting direction.

6. Conclusions and future works

Wire sawing is a key process to produce semiconductor silicon wafers. Slicing induces a wafer shape deformation, which is difficult to correct or mitigate. In this work, we applied several ML techniques to help identify the correlation between the saw machine parameters and the warpage. The models generated in this study can be used as a good tool to predict warpage values of as-cut wafers before they are physically measured. Of note, the most efficient model was generated using the recent Tabular Prior-data Fitted Network (TabPFN) [5], which produced effective results even with limited pre-processing on the warpage AVE target, although it exhibited quite unsatisfactory behavior on the STDEV target.

Other models were generated using traditional ML techniques: while LR and MLP did not show any particularly useful predictive ability, others, namely RF and LGBM, performed better, although with a not-so-strong correlation. In particular, the model generated with LGBM showed an R^2 of 24.8% with an MAPE of 15.7%. In contrast, TabPFN showed an ability to predict warpage with an MAPE as low as 14.7% and R^2 of up to 37.5%. This means that about half of the warpage AVG variability is explained by the model. On the other hand, the TabPFN model predicting warpage STDEV returned a negative R^2, indicating an inability to properly model this target variable. One possible reason to explain this can be traced to the approximations made during data preparation, where we had to use summary variables to allow for a match between input and output data.

Overall, the major slicing parameters correlating with warpage, highlighted by the TabPFN model, were found to be the coolant density, its temperature after slicing, the temperature of the coolant after the heat exchanger, and the amount of main drive torque to be applied to the system to overcome the slicing friction. Furthermore, looking at the LGBM results, we found another variable correlating with warpage: the bearing boxes' temperature. SHAP values confirmed a positive correlation between warpage average and coolant density, its temperature after slicing, and main drive torque, while the correlation with the coolant temperature after the heat exchanger is opposite.

To conclude, this article represents a first attempt to connect saw parameters and warpage using an ML approach; our main goal is to understand the main acting mechanisms, assessing the feasibility of ML-based approaches for the problem at hand, with the future goal of approaching the problem using more data and, possibly, more sophisticated models.

For instance, one possible approach to further improve our results could be to consider, as output, the warpage trend along the ingot (instead of the average warpage per cut, as we did in this study); moreover, by increasing the dataset size by collecting more data/cuts, the model performance should further improve. Another possibility would be to use raw data, instead of `Catch22`-derived features, as input to the ML model, and use ML models such as Multi-head CNN-LSTMs [25] or Transformers [26], which have shown promising results in similar time-series-related tasks like remaining useful life prediction. Finally, another way to further improve accuracy would be to go beyond a purely data-driven approach and consider the knowledge about

the physical mechanisms underlying wafer warpage [23] to develop a Physics-Informed Neural Network (PINN) [27]. Such a hybrid approach may not only improve prediction accuracy but also yield results that are more physically interpretable and insightful.

Besides attempting to improve the results, the most appealing future development of this study could be to use ML to modify the slicing recipe and parameters to control the warpage: since the slicing recipe controls the coolant and bearing boxes' temperatures, one could tune and modify their settings based on the model's output. For instance, the main drive torque could be controlled by changing the cutting time (i.e., increasing it to reduce the torque) or modifying the table feed speed profile.

CRediT authorship contribution statement

Carlo Zavattari: Writing – original draft, Methodology, Investigation, Formal analysis, Conceptualization. **Dario Demaria:** Writing – original draft, Visualization, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Fabrizio Bonda:** Methodology, Investigation, Data curation. **Giovanni Iacca:** Writing – review & editing, Supervision, Formal analysis, Conceptualization.

Funding

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Declaration of competing interest

The authors declare the following financial interests/personal relationships that may be considered as potential competing interests:

Carlo Zavattari reports that financial support was provided by MEMC Electronic Materials SpA. Dario Demaria reports that financial support was provided by MEMC Electronic Materials SpA. Fabrizio Bonda reports that financial support was provided by MEMC Electronic Materials SpA. Carlo Zavattari has patent #US Provisional Application Serial No. 63/886,493 pending to MEMC Electronic Materials SpA. Carlo Zavattari has patent #US Patent Application Publication No.2013-0144421A1 issued to MEMC Electronic Materials SpA. Giovanni Iacca has patent #US Provisional Application Serial No. 63/886,493 pending to MEMC Electronic Materials SpA. Fabrizio Bonda has patent #US Provisional Application Serial No. 63/886,493 pending to MEMC Electronic Materials SpA. Dario Demaria has patent #US Provisional Application Serial No. 63/886,493 pending to MEMC Electronic Materials SpA. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A. List of abbreviations used in the paper

Table A.10
Industry- and wafer-related abbreviations.

Abbreviation	Explanation
ESG	Environment, Social and Governance
DCW	Diamond Coated Wire
MRR	Material Removal Rate
SSD	Sub Surface Damage

Table A.11
Statistics- and Machine Learning-related abbreviations.

Abbreviation	Explanation
ANN	Artificial Neural Network
APE	Absolute Percentage Error
AVE	Average Target
CNN	Convolutional Neural Network
ICL	In-Context Learning
IQR	Inter-Quartile Range
KDE	Kernel Density Estimation
LGBM	Light Gradient Boosting Machine
LR	Linear Regression
LSTM	Long-Short Time Memory
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
ML	Machine Learning
MLP	Multi-Layer Perceptron
NAN	Not A Number/empty value
PFN	Prior Fitted Network
R^2	Coefficient of determination
RELU	REctified Linear Unit
RF	Random Forest
RMSE	Root Mean Square Error
SHAP	SHaply Additive ExPlanation
STDEV	Standard Deviation Target
TabPFN	Tabular Prior-data Fitted Network

Table A.12
Wire saw parameters.

Abbreviation	Explanation
ABS_F mm]	feed unit absolute position during slicing
REL_V [mm]	relative feed unit position from (0 mm = cut starting position)
v_FED [mm/min]	table feed speed
v_Wir [mm/min]	wire speed
N3201 [N]	wire tension (used wire side)
N3221 [N]	wire tension (new wire side)
B4003 [Deg. C]	left fixed bearing temperature
B4007 [Deg. C]	right fixed bearing temperature
B4001 [Deg. C]	left movable bearing temperature
B4005 [Deg. C]	right movable bearing temperature
B3801 [Deg. C]	left coolant temperature
B3803 [Deg. C]	right coolant temperature
B3793 [mbar]	coolant pump pressure
B3905 [Deg. C]	coolant temperature (return to tank)
B3901 [Deg. C]	coolant temperature after heat exchanger
B1771 [Deg. C]	left coolant inlet temperature
B1773 [Deg. C]	right coolant inlet temperature
B3801 [kg/h]	left coolant flow
B3803 [kg/h]	right coolant flow
B3801 [kg/l]	left coolant density
B3803 [kg/l]	right coolant density
A1601 [Nm]	left main motor torque
A1611 [Nm]	right main motor torque

Data availability

Data will be made available on request.

References

[1] N.S. Poduje, ADE measurement tehcnology, 1989.
[2] ADE Corporation, Technical publications department, UltraGage 9500/9700, 1995.
[3] A. Li, S. Hu, Y. Zhou, H. Wang, Z. Zhang, W. Ming, Recent advances in precision diamond wire sawing monocrystalline silicon, Micromachines 14 (8) (2023) 1512.
[4] A. Kumar, S.N. Melkote, Diamond wire sawing of solar silicon wafers: a sustainable manufacturing alternative to loose abrasive slurry sawing, Procedia Manuf. 21 (2018) 549–566, 15th Global Conference on Sustainable Manufacturing.
[5] N. Hollmann, S. Müller, L. Purucker, A. Krishnakumar, M. Körfer, S.B. Hoo, et al., Accurate predictions on small data with a tabular foundation model, Nature 637 (8045) (2025) 319–326.

[6] E. Kayabasi, S. Ozturk, E. Celik, H. Kurt, Determination of cutting parameters for silicon wafer with a diamond wire saw using an artificial neural network, Sol. Energy 149 (2017) 285–293.
[7] Y. Zhao, P. Ge, W. Bi, J. Zheng, A representation points based method for object detection on the surface of electroplated diamond wire saw, J. Manuf. Process. 107 (2023) 57–64.
[8] Z. Zhang, F. Yin, G. Huang, C. Cui, Process parameter optimization for diamond wire saw machining of stone, Int. J. Adv. Manuf. Technol. (2025) 1–12.
[9] S.C. Jain, S.S. Rathore, Prediction of cutting performance of diamond wire saw machine in quarrying of marble: a neural network approach, Rock Mech. Rock Eng. 44 (3) (2011) 367–371.
[10] R. Mikaeil, S. Shaffiee Haghshenas, Y. Ozcelik, S. Shaffiee Haghshenas, Development of intelligent systems to predict diamond wire saw performance, J. Soft Comput. In Civ. Eng. 1 (2) (2017) 52–69.
[11] M.G. Fernandez, A. Tokuhiro, K. Welter, Q. Wu, Nuclear energy system's behavior and decision making using machine learning, Nucl. Eng. Des. 324 (2017) 27–34.
[12] A.N. Qadrouh, J.M. Carcione, M. Alajmi, M.M. Alyousif, A tutorial on machine learning with geophysical applications, Boll. Di Geofis. Teor. Ed Appl. 60 (3) (2019).
[13] C.H. Lubba, S.S. Sethi, P. Knaute, S.R. Schultz, B.D. Fulcher, N.S. Jones, Catch22: CAnonical time-series CHaracteristics: selected through highly comparative time-series analysis, Data Min. Knowl. Discov. 33 (6) (2019) 1821–1852.
[14] R.D. Camino, C.A. Hammerschmidt, R. State, Improving missing data imputation with deep generative models, arXiv preprint arXiv:1902.10666, 2019.
[15] N. Hollmann, S. Müller, K. Eggensperger, F. Hutter, TabPFN: a transformer that solves small tabular classification problems in a second, arXiv preprint arXiv:2207.01848, 2022.
[16] D. Rundel, J. Kobialka, C. von Crailsheim, M. Feurer, T. Nagler, D. Rügamer, Interpretable machine learning for TabPFN, in: World Conference on Explainable Artificial Intelligence, Springer, 2024, pp. 465–476.
[17] H.-J. Ye, S.-Y. Liu, W.-L. Chao, A closer look at TabPFN v2: strength, limitation, and extension, arXiv preprint arXiv:2502.17361, 2025.
[18] J. Qu, D. Holzmüller, G. Varoquaux, M. Morvan le, TabICL: a tabular foundation model for in-context learning on large data, arXiv preprint arXiv:2502.05564, 2025.
[19] R. Shwartz-Ziv, A. Armon, Tabular data: deep learning is not all you need, Inf. Fusion 81 (2022) 84–90.
[20] L. Grinsztajn, E. Oyallon, G. Varoquaux, Why do tree-based models still outperform deep learning on typical tabular data? Adv. In Neural Inf. Process. Syst. 35 (2022) 507–520.
[21] S. Uddin, H. Lu, Confirming the statistically significant superiority of tree-based machine learning algorithms over their counterparts for tabular data, PLoS One 19 (4) (2024) e0301541.
[22] S.M. Lundberg, S.-I. Lee, A unified approach to interpreting model predictions, Adv. Neural Inf. Process. Syst. 30 (2017).
[23] C. Zavattari, S. Bhavagat, G. Iacca, F. Bonda, D. Demaria, Systems and methods for modeling and controlling surface profiles of wafers sliced in a wire saw, US Provisional Application Serial No. 63/886, 493, 2025.
[24] C. Zavattari, F. Severico, S.S. Bhagavat, G. Vercelloni, R.R. Vandamme, Systems for controlling temperature of bearings in a wire saw, US Patent Application Publication No. 2013-0144421A1, 2013.
[25] H. Mo, F. Lucca, J. Malacarne, G. Iacca, Multi-head CNN-LSTM with prediction error analysis for remaining useful life prediction, in: Conference of Open Innovations Association, Ieee, 2020, pp. 164–171.
[26] H. Mo, G. Iacca, Evolutionary neural architecture search on transformers for RUL prediction, Mater. Manuf. Process. 38 (15) (2023) 1881–1898.
[27] M. Raissi, P. Perdikaris, G.E. Karniadakis, Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, J. Comput. Phys. 378 (2019) 686–707.