The International Journal of Advanced Manufacturing Technology (2024) 133:5657–5672
https://doi.org/10.1007/s00170-024-14055-z

ORIGINAL ARTICLE

Machine learning-based shear force quality prediction of ultrasonic wire bonds: utilizing process data and machine data without additional sensors

Christoph Buchner[1,2] · Christian T. Seidler[3] · Marco F. Huber[2,4] · Hartmut Eigenbrod[4] · Hans-Georg von Ribbeck[5] · Franz Schlicht[5]

Received: 12 March 2024 / Accepted: 21 June 2024 / Published online: 5 July 2024

Abstract

Ultrasonic wire bonding is a highly automated production process that is used billions of times a year in the electronics and electromobility industries. Due to the complexity of the process and the large number of influencing parameters, there are currently no automated methods that can be used without additional sensors to evaluate the shear force bond quality quantitatively and non-destructively with sufficiently high precision. For this reason, this paper presents a new methodology that uses machine learning to enable quantitative, non-destructive prediction of bond quality without additional sensors. For this purpose, a machine learning algorithm was developed that uses various machine data and process data from existing sensors to quantitatively predict the shear force of the bonded joint. In addition, features are extracted from process time series, such as current, power, and frequency of the ultrasonic generator as well as deformation during bonding. Of the total of 2,090 features considered, the number of features could be reduced to 26 by recursive feature elimination, while maintaining the prediction accuracy. By using optimized deep neural networks, on average, a prediction precision of the regression on the shear force of the source bond of over 89.6% R^2-score and a mean absolute error of 241 cN can be achieved.

Keywords Ultrasonic wire bonding · Quality prediction · Bond quality · Machine learning · Shear force

Abbreviations

DoE	Design of experiment
IC	Integrated circuit
MAE	Mean absolute error
ML	Machine learning
PCA	Principal component analysis
RFE	Recursive feature elimination
RFECV	Recursive feature elimination with cross-validation

Christoph Buchner and Christian T. Seidler contributed equally to this work.

✉ Christoph Buchner
christoph.buchner@strama-mps.de

Christian T. Seidler
christian.seidler@ipa.fraunhofer.de

Marco F. Huber
marco.huber@ieee.org

Hartmut Eigenbrod
hartmut.eigenbrod@ipa.fraunhofer.de

Hans-Georg von Ribbeck
hans-georg.vonribbeck@de.fkdelvotec.com

Franz Schlicht
franz.schlicht@de.fkdelvotec.com

[1] Strama-MPS Maschinenbau GmbH & Co.KG, Ittlinger Straße 195, 94315 Straubing, Germany

[2] Institute of Industrial Manufacturing and Management IFF, University of Stuttgart, Allmandring 35, 70569 Stuttgart, Germany

[3] Fraunhofer Institute for Manufacturing Engineering and Automation IPA, Alte Bahnhofstraße 2, 97422 Schweinfurt, Germany

[4] Fraunhofer Institute for Manufacturing Engineering and Automation IPA, Nobelstraße 12, 70569 Stuttgart, Germany

[5] F & K DELVOTEC Bondtechnik GmbH, Daimlerstraße 5, 85521 Ottobrunn, Germany

1 Introduction

Ultrasonic wire bonding is an important process in the electronics industry for integrated circuit (IC) assembly, in which the IC chip is electrically connected to the substrate [1] and that is used billions of times a year worldwide [2]. Wire bonding is also widely used as a production process in electromobility.

The ultrasonic wire bonding process is a very effective and flexible technology [3]. In addition, the bond has good electrical properties and excellent thermal conductivity [1]. Due to the high degree of automation, the production process with its properties is suitable for series production [4]. Another advantage is the low thermal energy input during the production process.

Ensuring high bond and product quality is a technical challenge in production. Due to the increased demands in power electronics, higher performance, robustness, and flexibility are required in assembly and connection technology. In addition, copper is increasingly being used as a wire material instead of aluminum in highly stressed applications due to its material properties. This change requires higher ultrasonic power and bonding forces in the ultrasonic bonding process. Overall, copper wire bonding poses an even greater challenge to quality than aluminum wire bonding [2].

The currently predominant manual quality control of bonded joints requires a great deal of work and associated costs and can also be prone to errors [5].

The relationship between the influencing parameters and the bonding quality is complicated and non-linear [6]. For this reason, it is expected that the application of machine learning (ML) methods will enhance the quality inspection process.

The method presented in the paper enables automated, sensor-less evaluation of bond quality in real-time during the bonding process. Sensor-less means that no additional sensors are installed on the machine, only the already available sensors are used. This precise quality prediction is based on the calculation of machine algorithms that take into account a wide range of features from different domains. Machine data, quantitative process data, and process time-series of various existing sensors such as deformation of the wire while bonding as well as frequency, active power, and current measured in the bonding machine are used.

The paper is structured as follows. Section 2 gives an overview of the wire bonding process and methods for quality prediction in ultrasonic bonding. Section 3 presents the new methodology based on ML for quality prediction in ultrasonic bonding. The actual implementation of the quality prediction method is described in the Section 4. Section 5 contains the results obtained and the discussion. Section 6 concludes with a summary of our contributions and future research directions.

2 State of the art

2.1 Ultrasonic wire bonding

The basic active principle of an ultrasonic bonding system is achieved by applying an alternating current/voltage signal to a piezo element, the so-called transducer. The high-frequency signals create horizontal vibrations on the transducer. These vibrations are introduced into the bond tool and lead to sinusoidal waves at the horn tip, which stimulate diffusion processes between the wire and substrate material due to a simultaneously applied vertical force [3]. A schematic drawing is shown in Fig. 1.

Figure 2 shows the individual process steps for ultrasonic (wire) bonding. By definition, the process (1) starts with a

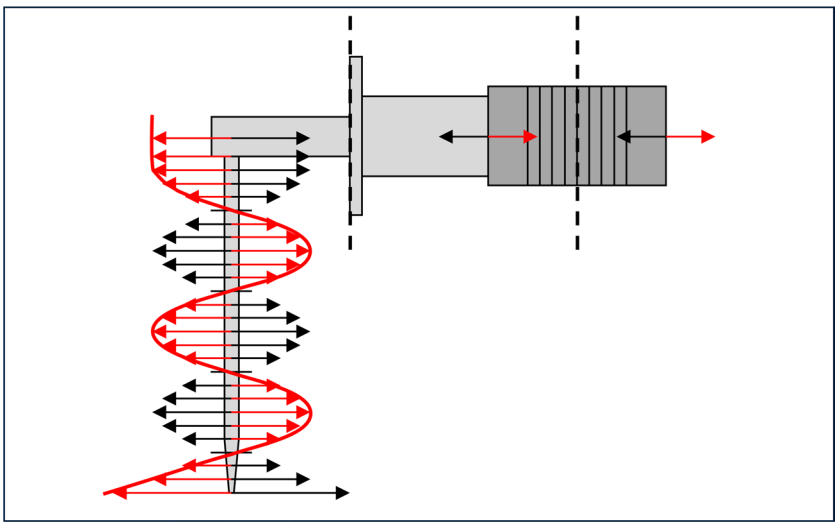

Fig. 1 Propagation of the piezo oscillation at the transducer and at the bond tool

touchdown phase. The substrate surface is searched for with a reduced force. The wire is then pressed onto the substrate surface by the bonding tool using a touchdown force. During this process, after a certain predefined relative displacement between the bond head frame and the bonding tool is reached, the clamp is opened, and the wire is only fixed by the contact between the tool and the substrate. In the next process step (2), a defined bonding force is applied, during which the ultrasonic power is introduced into the wire/substrate via the bond tool. This creates the first wire end as a contact connection. The bonding head then moves along a programmed loop geometry (3) with the clamp open. After the bonding head has traveled a certain distance, the clamp is closed, and the bonding head moves to the target point of the second contact surface (4). In process step (5), a touchdown phase follows, analogous to step (1), in which the wire is fixed to the second target contact surface. This is followed by the application of ultrasonic power to create the bond (6) in the same way as the bond is created on the first contact surface. In step (7), the bonding head moves further in the direction of the wire at a certain height above the substrate in order to set a specific waist length. The further process step (8) is only carried out for thick wire bonding and is omitted for thin wire bonding: The wire that has just been fixed is pre-damaged locally using a knife on the bonding head. This is necessary in the ultrasonic thick wire process as the wire cannot be torn off easily. In the last process step (9), the wire is torn off using a closed clamp. This process is repeated for the next wire once the bonding head has moved to the new contacting surface [2].

2.2 Quality prediction

There are different approaches in the literature for predicting the quality of bonded joints. The methods differ particularly in the type of information sources used. A large number of methods use process data, especially the electrical signals from the ultrasonic generator, for prediction. More recent approaches use image processing methods for evaluation. The literature also describes approaches that use statistical, simulative, or theoretical calculation methods. The various approaches from the literature are described in more detail below.

2.2.1 Process data

Numerous existing approaches from the literature use process data to assess the quality of the bond joints. The methodology proposed by Zhang and Ling [7] uses a time–frequency analysis of the horn vibration to monitor the bond quality. From the frequency time-series, 45 features are extracted that are used to predict the shear force in a three-layer neural network.

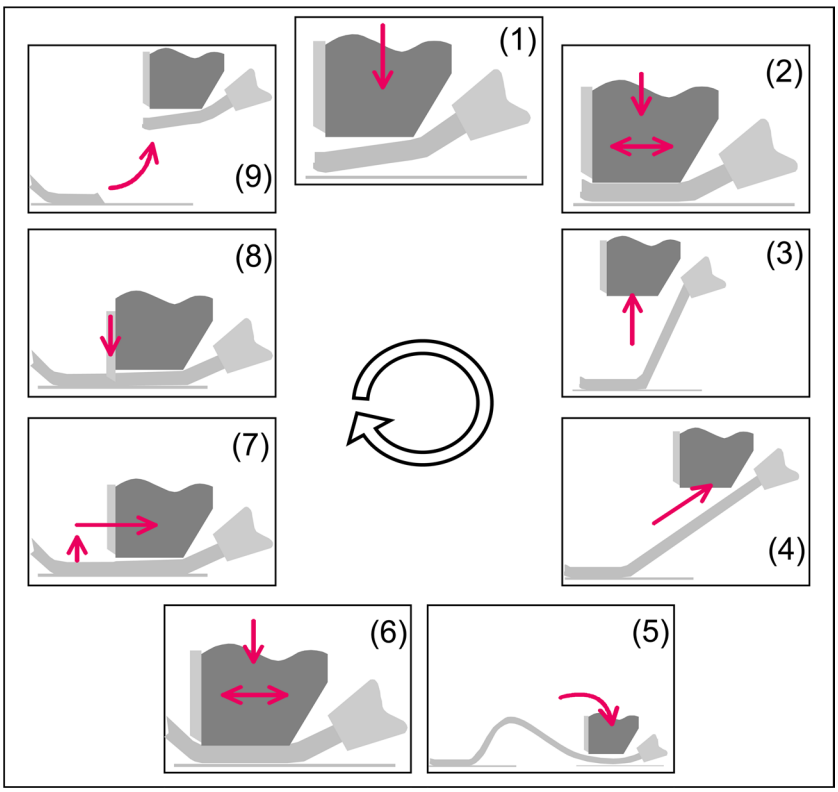

Fig. 2 Process steps in ultrasonic wire bonding

Ling et al. [8] use the impedance time-series of the ultrasonic generator to evaluate the bond quality. This series is calculated from the current and voltage using the Hilbert transformation. A total of 25 features are extracted from the series. As a quality criterion, the shear force is predicted using a neural network with one hidden layer.

Feng et al. [4] use the current and voltage time-series of the ultrasonic generator as a source of information for predicting the shear force of the bond joint. From this, 115 features are selected in a first step. Principal component analysis (PCA) is then used to further reduce the number of features to 13. The approach subsequently uses a neural network with one hidden layer to predict the bond shear force as well.

Arjmand et al. [9] present a non-destructive technique for predicting bond quality by applying a semi-supervised classification algorithm to process the signals obtained from an ultrasonic generator. For this purpose, the current time-series of the ultrasonic generator is used as an influencing variable. Features are calculated from the time-series using a fast Fourier transformation. Then, semi-supervised discriminant analysis is applied. The classification can identify three distinct classes. The quality of the joint is determined using X-ray tomography by assessing the bond area.

In a further method [6], the source bond quality during ball bonding is predicted using a neural network. The study was preceded by data aggregation using an experimental orthogonal design in which nine parameters were varied. The neural network contains one hidden layer. The bond temperature, bond time, bond power, bond force, and size ratio are used as quantitative features. The ball diameter and shear force are calculated as output.

In a similar production process, namely, ultrasonic welding of metals, a slightly more comprehensive approach is described that uses various sensor data for quality assessment [10]. The current, voltage, and frequency of the ultrasonic generator, the shear force applied between the horn, and the machine and the vibration amplitude of the horn, as well as the vertical displacement of the horn are used as features. The number of features is reduced from over 5 million to 170. Different feature reduction methods are compared. By classifying the features into two classes "good" and "bad," a quality prediction accuracy of over 90% is achieved. The regression achieves an average R^2-score of 85% for the prediction of peel strength.

In another approach [1], the position time-series of the bond head of wire bonding systems are evaluated using deep learning in order to deduce a fault diagnosis. Features in the time-domain, time–frequency-domain, and frequency-domain are selected from the measured position time-series. A ML algorithm consisting of a deep neural network and a convolutional neural network then predicts the position errors.

2.2.2 Image data

In more recent methods, image-processing ML solutions are increasingly being used to assess the quality of bonded joints. Wang and Chiu [11] classify the shear force before wire bonding based on 2D images using ML methods. The shear force is evaluated based on six features prior to wire bonding. The results show that a PCA-based random forest can identify bad chips before wire bonding with an accuracy of 97.92%.

A deep learning-based computer vision technique using autoencoders for performing quality inspections of bonded joints is performed on a test specimen, a particle detector design [5].

Further image processing systems for assessing bond quality are described for example in Long et al. [12] and Chen et al. [13].

2.2.3 Others

Other approaches use theoretical calculation models, statistical evaluations, or simulative approaches to assess bond quality.

Gaul's [14] dissertation theoretically calculates the bond quality in ultrasonic wedge/wedge bonding. He extended an already existing model for calculating the interface activation energy. The properties of the substrate, the wire, the bond surface, the bonding machine, and the tool are taken into account. This method makes it possible to predict the resulting shear force of the bond using the calculation model set up to calculate the interface activation energy. From this, the required ultrasonic power can also be predicted that leads to a quality-compliant bond.

Tsai [15] describes a methodology for an adaptive diagnostic system for the control and quality assessment of a copper wire bonding process using gray relativity analysis and a neuro-fuzzy technique.

The study by Yu et al. [16] proposes a fuzzy quality evaluation model for the gold wire bonding process in IC packaging. A process quality index is used as a quality criterion. In addition, statistical methods such as DeMorgan's rules and Boolean inequalities are used to create the upper confidence interval. Finally, a fuzzy radar evaluation diagram is derived to assess the quality of the bond connection.

Huai et al. [17] set up a magnetostatic simulation with the software Ansys Maxwell for bond connections and exported sixteen features that describe magnetic fields. These are then evaluated in an ML-algorithm to assess wire bond lift-offs.

Table 1 summarizes the advantages and disadvantages of using different data origins. The main advantage of process data is that it is usually recorded by the machine itself and does not require any additional sensors.

Table 1 Advantages and disadvantages of different data origins (process data, image data, others)

Data origin	Advantages	Disadvantages
Process data	Data is automatically recorded by the bonder (= no additional sensors needed)	Information limited by sensors within the bonder (= manufacturer)
Image data	Cameras can capture additional characteristics to evaluate the bond quality	Additional sensors (cameras) needed and extra process step after bonding
Others	Highly customizable for individual problem	Large manual effort required and difficult scalability

3 Methodology for quality prediction

This section addresses different aspects of the developed methodology: First, general requirements for a quality prediction system predicting the shear force of source bonds are described. Second, an overview of the methodology is given. Finally, the methodology is systematically explained in detail.

3.1 Requirements for the methodology

The following requirements result from the described objective in Section 1:

a Quantitative automated quality prediction of the ultrasonic bonding in real-time is carried out with high predictive precision.

b Different existing machine data and process data including different sensor time-series are taken into account as features for sensor-less prediction.

3.2 Overview of the methodology

Figure 3 provides an overview of the research framework. The first step is to collect data for later model training, testing and validation. A design of experiment (DoE) is conducted, and machine and process data are recorded. For each bond, the shear force is measured using the shear test, defining the continuous target value for quality prediction. Additionally, the recorded machine and process parameters are reviewed in order to eliminate features that have no practical use for the prediction, like wire number or timestamp. Among the recorded process parameters are four time-series: deformation, ultrasonic power, frequency, and current in the ultrasonic generator. The length of these time-series

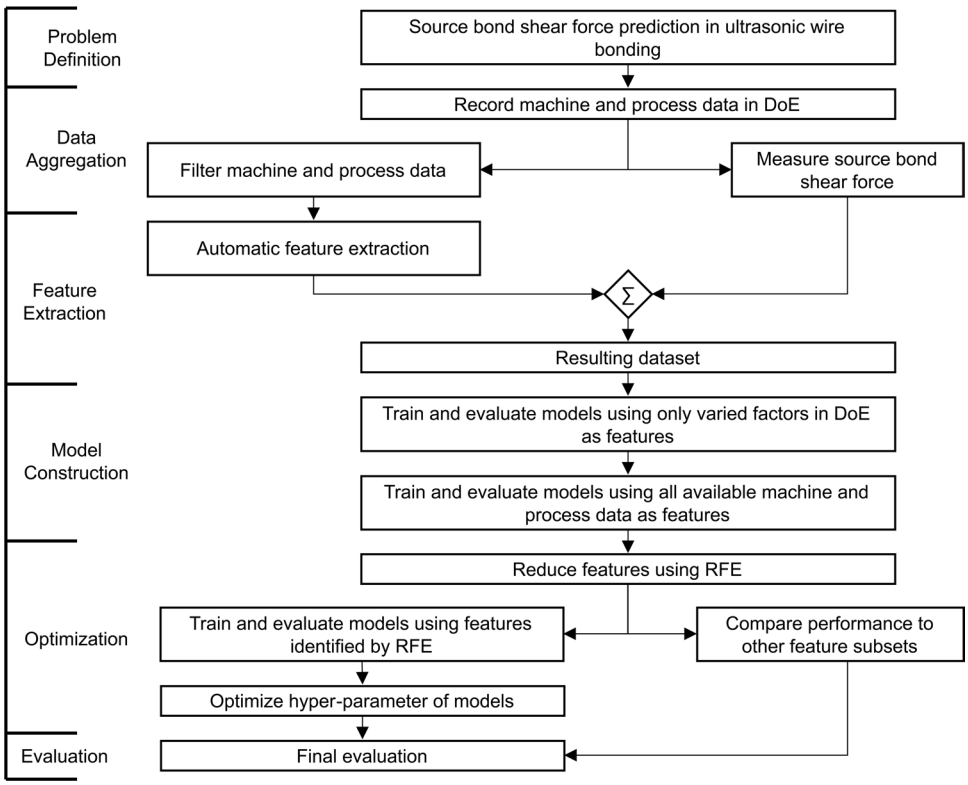

Fig. 3 Developed research framework

vary across different bonds in the DoE, and various features from the time-domain, frequency-domain, as well as time–frequency-domain are automatically extracted from the signals. Combining the target values and the features leads to a tabular dataset. The next step is to construct an ML model for regression. Five different types of ML-algorithms will be trained and evaluated in this step. First, the models are trained with the varied parameters of the DoE only, thereby forming a performance baseline. Then, the models are trained on all extracted features. The next step in the framework is model optimization. In this step, first, the number of features necessary for the regression task is reduced by using recursive feature elimination (RFE). The model performance on the resulting feature subset is then compared to the performance on other feature subsets, e.g., using only features extracted from current or voltage time-series in the ultrasonic generator. Then, the hyper-parameters of the models are tuned, using the feature subset identified by RFE, to further increase prediction quality. Finally, the best identified model is trained and evaluated in detail.

3.3 Detailed description of the methodology

3.3.1 Data aggregation

The first step is to collect data. Similar to approaches by other authors, a DoE is conducted [10]. The advantage of this approach is that different parameters can be varied and their influence on the bond quality can be mapped with a feasible number of experimental trials [18]. A full factorial design allows modeling the relationship between the dependent and independent variables, using the main factors and all interaction terms (between the factors). The number of trials needed for a full factorial design increases exponentially on the number of factors. In practice, interaction terms of order three and higher are often very small. Assuming that interaction terms of order three and higher are very close to zero, a fractional factorial design with resolution V can be used. Here, the main factors are superimposed with interaction terms of order four and the two-way-interaction terms are superimposed with certain three-way-interactions. A fractional factorial design drastically lowers the number of trials needed, e.g., for ten factors with two levels and resolution V from $2^{10} = 1024$ to 128 trials. The factors to vary in the DoE should have a known effect on the bond quality. A bonding program with a process-stable parameter configuration is used as foundation. The variation of the parameters can be selected according to expert assessments within a process window of approx. $\pm 20\%$.

A large number of parameters influence the bond quality. Figure 4 gives an overview over identified influential factors on the bond quality. Even more parameters can influence the shear test results. The factors marked in blue are taken from literature, while the gray italic factors were identified in expert surveys, as well as in production data [1, 4, 6–11, 14, 15, 19]. The parameters can be divided into different groups. The wire, the substrate, the metallization on the substrate, and the bonding machine have a large influence on the bonding quality. Additionally, the values recorded by existing, machine integrated sensors during the bonding process, and the bonding parameters set in the bonding program also have a large influence.

The process design, the process environment, and the quality technology, such as a real-time controller or deformation monitoring, also influence the bond quality, albeit smaller than the previously mentioned groups. The overview is intended to provide a detailed insight into the parameters influencing bond quality. However, due to the complexity of the process, it cannot be claimed to be exhaustive.

For each experimental trial, several hundred different machine and process data are recorded by the bonding machine. Among these are parameters with no relation to the bonding quality, like timestamp of the bond. Thus, through expert surveys, recorded parameters with no relationship to the bonding quality are discarded from the dataset.

The quality of a bond has many different aspects. In general, these aspects can be assessed qualitatively or quantitatively. Common methods for qualitative assessment of bond quality are based on visual inspections of the bonds, e.g., with regard to reproducibility, loop shape, or bond surface design [5]. Various test methods exist for quantitative assessment of the bond connection. For thick wires (upwards of 125 µm), the shear force test is used, while for thin wires (smaller than 75 µm), the pull test leads to a more meaningful result, because it is difficult to adjust the shear chisel to the small bond foot from a process engineering point of view. Another quantitative assessment is the bond foot area, from which an estimation of the connection strength can be calculated. The case of damage which includes lift off or cratering is another quantitative bond characteristic [2]. Finally, the quality of the bonded joint can be measured using special removal tests. These allow an assessment of the life time of the joint, e.g., with temperature cycles or vibration cycles [9]. Figure 5 shows an overview of different bond quality indicators identified from literature and expert surveys.

3.3.2 Feature extraction

Among the automatically recorded machine and process parameters are four time-series. These are the deformation of the bond, the ultrasonic power, as well as the current and frequency in the ultrasonic generator. The bonding time is one of the factors varied in the DoE. The time-series cannot directly be used in most ML-algorithms, as most require a fixed-length features.

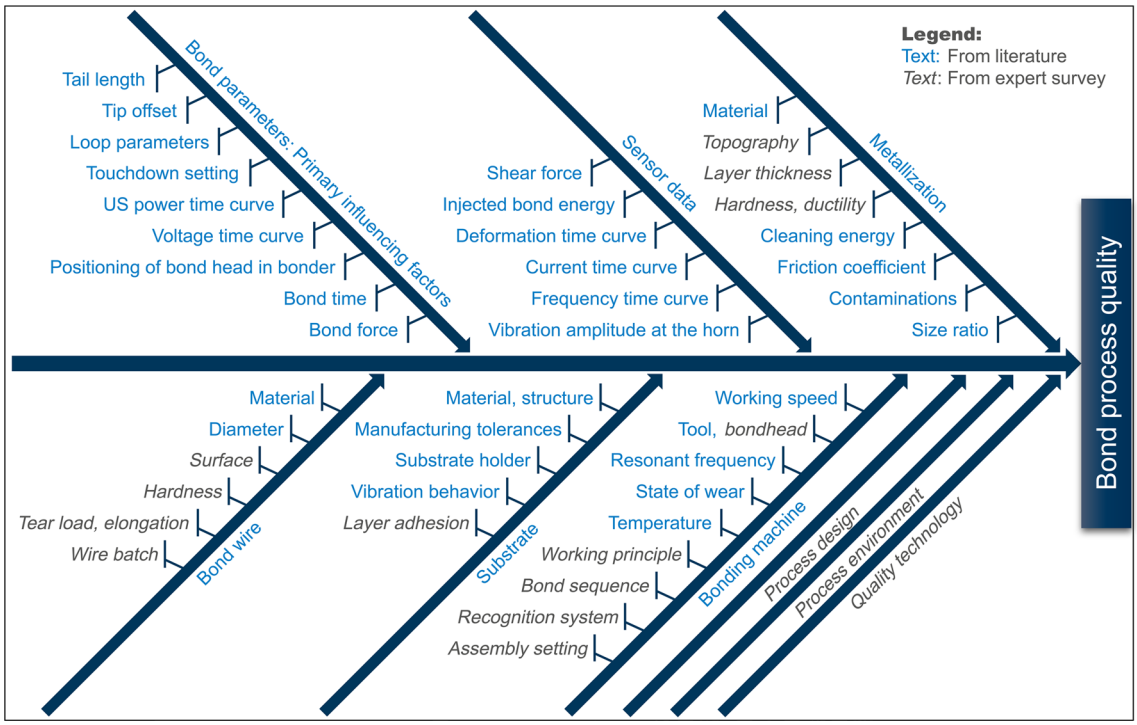

Fig. 4 Overview of parameters influencing the bond quality identified through literature research and expert surveys

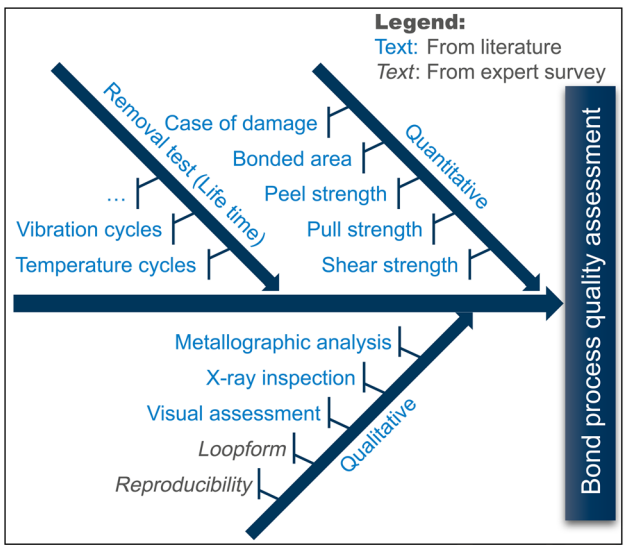

Fig. 5 Overview of quality indicators for the bond quality, identified by literature review and expert surveys

Feature extraction is the process of transforming (in this case) time-series data into quantitative values, e.g., in the time-, frequency- and time–frequency-domain. Multiple python packages exist to extract a large number of features from a time-series, e.g., tsfresh, TSFEL, and pycatch22. The calculated features vary from package to package. An empirical evaluation from Henderson and Fulcher [20] showed that the package tsfresh has an overlap of roughly 76% to other packages. Thus, tsfresh calculates the majority of features implemented in other packages and is selected for feature extraction in this work.

Combining the calculated features from tsfresh with the tabular process and machine data leads to the dataset used in this work and the measured source bond shear force as target.

3.3.3 Model construction

This step consists of two parts. First, we trained and evaluated five regression models using only the ten varied DoE factors as features. This forms a baseline for comparison. The evaluation is performed on a test set, which is unseen during model training. An 80–20 train-test-split is used throughout this examination. The model is evaluated on 20 different train-test splits, to get a robust estimation of the average expected performance. Second, the models are trained and evaluated with all recorded features.

3.3.4 Optimization

The next step is to optimize the ML model in terms of number of features required for an accurate regression result and prediction quality. Since there are more available features than observations, the curse of dimensionality may lead to a reduced prediction quality. Thus, feature selection is applied

first, to reduce the number of features to a value smaller than the number of observations. Several techniques for feature reduction exist, like PCA, feature importance, RFE, or selection processes considering the Akaike information criterion.

In this work, RFE, combined with cross-validation, is used for feature selection, as RFE considers both relevance and interactions of the features simultaneously for a given optimization criteria [21]. RFE is an iterative technique that relies on feature importance. Thus, this work uses random forest as estimator of the feature importance and the mean absolute error as optimization criteria.

Once an optimal feature subset has been identified by RFE, we evaluate the performance of the ML models (as described before) trained on this feature subset. This forms the baseline for the following hyper-parameter tuning, to further increase prediction quality.

Additionally, the performance of the regression model trained on this feature subset is compared to other features subsets, suggested by literature and expert surveys. Because of the differences in measurement setups, we cannot directly compare our regression results with those of other approaches. However, using features extracted from the measured time-series, a comparison of performance of regression models trained on suggested features is possible. These include calculating the system impedance from the recorded current and voltage signals [8, 22], as well as using features from only current and/or voltage signals [4, 9, 23]. It is important to note again that a direct comparison to these approaches is not possible, due to different sensors and bonding machines used. Thus, the comparison shows the differences in model performance when limiting the information to certain domains in the bonding process, within the data available from a specific bonding machine.

3.3.5 Evaluation

Several metrics exist to evaluate a regression model in terms of the reliability of the prediction. The coefficient of determination (R^2-score, short R^2) and the mean absolute error (*MAE*) are commonly used metrics for evaluating the prediction quality of a regression model and are used in this paper. R^2-score describes the relationship between two input variables. R^2 ranges from zero to one. A value of one means that the model is optimal, while a value of zero means that the model is suboptimal. The R^2-score is defined as follows:

$$R^2\left(y,\hat{y}\right) = 1 - \frac{\sum_{i=0}^{n-1}\left(y_i - \hat{y}_i\right)^2}{\sum_{i=0}^{n-1}\left(y_i - \bar{y}\right)^2}, \tag{1}$$

where.

y_i stands for the observed values.

$\hat{y}_i$ stands for the predicted or estimated values.

$\bar{y}$ denotes the mean value of the observed values.

The equation for the mean value of all observed values $\bar{y}$ is given by the following:

$$\bar{y} = \frac{1}{n}\sum_{i=0}^{n-1} y_i, \tag{2}$$

where n is the number of samples. The MAE given by

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|\hat{y}_i - y_i\right| \tag{3}$$

is the mean absolute error.

3.3.6 Utilization in shop floor

Figure 6 shows an overview of the proposed ML framework in productive use. First, the recorded quantitative machine and process data are filtered, to remove irrelevant features. Simultaneously, the pre-defined features, which were identified as relevant during model training, from the four time-series are extracted. Then, both machine and process data, as well as the extracted time-series features, are scaled to a range of (0, 1), as described in Section 4.3. The normalized features are then fed into the trained regression model, which predicts the shear force of a bond that can be used as an estimate for the bond quality.

4 Implementation

This section describes the implementation of the methodology described in the previous section.

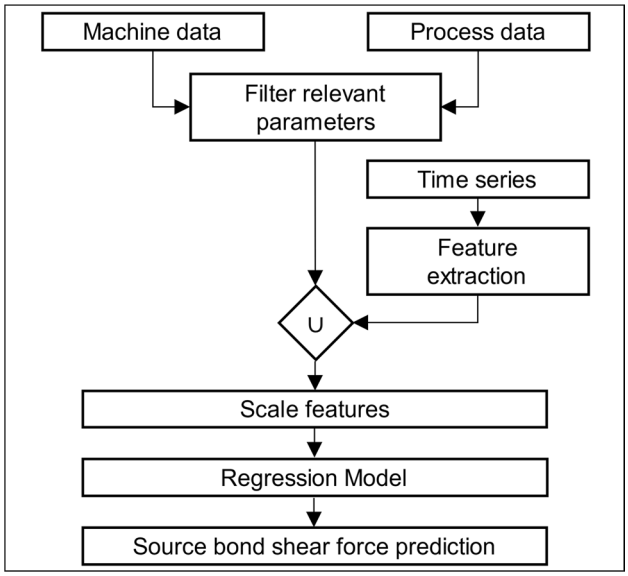

Fig. 6 Diagram of the proposed machine learning framework in production

4.1 Data collection/data gathering

First, data is gathered for the ML system using DoE. Through talks with process experts, ten factors are identified for the DoE. The selected factors include information about the touchdown (factor: 1, 2), the bond force (factor: 3, 4), the induced ultrasonic power (factor: 5–9), and the loop speed (factor: 10). Each factor is varied on two levels, a low and a high level, which deviate by 20% from an initially selected value. This initial value is added three times to the design plan as a center point. Table 2 shows the selected factors and their respective setting for each level. A full factorial design for 10 factors with two levels each would require $2^{10} = 1024$ experimental tries. As the factors originate from different parts of the bonding process, interaction terms between factors have no practical implication. Therefore, a fractional factorial design of resolution V is selected as the foundation of the design plan. The selected fractional factorial design has 128 runs. When adding the initial value of each factor three times as center point, the design consists of 131 runs. To account for process variation, each setting is repeated five times. In total, the resulting trial design used in this study consists of 655 trials. For each trial, the bonding device records several parameters stemming from machine and process data automatically. In total, more than 100 various parameters are available with this approach. The signals are recorded at 6 kHz during bonding and, depending on the length of the signals, downsampled to a maximum of 128 values to reduce the memory space. Using an M17S ultrasonic wire bonder from F & K DELVOTEC with bonding tool WED-31212 (Wolfram carbide), copper wires with a diameter of 380 µm (PowerCu Soft from Heraeus) are bonded to a copper plate with a size of 100 × 60 × 1 mm as part of the DoE at room temperature.

As the selected wire diameter is in the thick wire range, the shear force of the bond is determined exemplary as one of the quantitative quality characteristic. For easy data generation, an automated Xyztec Sigma shear tester is used in addition to the automated bonding machine to produce the bond connections according to the test plan.

Intensive literature studies, expert discussions, and analysis of production data are performed to extract only relevant parameters regarding the bond quality. In addition, the process time-series for current, power, and frequency of the ultrasonic generator and the recorded deformations during bonding are taken into account. Figure 7 shows the recorded time-series for four different factor settings.

Table 2 Factors and level assignments for the DoE used

Factor	Initial value (center point)	Low factor setting	High factor setting
1. Touchdown force in cN	1500	1200	1800
2. Touchdown steps in µm	200	160	240
3. Bond force start in cN	1800	1440	2160
4. Bond force end in cN	3000	2400	3600
5. Ultrasonic init power in %	60	48	72
6. Ultrasonic power in Dig	210	168	252
7. Ultrasonic ramp time 2 in ms	50	40	60
8. Temper time in ms	100	80	120
9. Bonding time in ms	500	400	600
10. Loop velocity in %	100	80	120

4.2 Feature extraction

The ML pipeline is implemented in Python 3.11. The first step in the pipeline is feature extraction. The python package "tsfresh" [24], version 0.20, is used to extract a large number of features from the time-domain, frequency-domain, and time–frequency-domain for every time-series. This way, when adding the scalar machine and process data, 2090 features are available. Of these 2090 features, 102 stem from the machine and process recordings, as well as the varied parameter in the DoE, while the remaining 1988 features are from the four recorded time-series using tsfresh.

4.3 Data preprocessing

For model training and evaluation, the dataset is split into training and test set. From the 655 collected samples, 131 form the test set. This equals 20% of the entire dataset. The remaining 524 samples form the training set. Prior to training the models on the training data, two additional preprocessing steps are applied. First, the dataset is cleaned. In this step, features with a variance of zero are removed. These features have only one value across the entire training dataset and therefore carry no information for the model. Furthermore, features with invalid values across all samples are removed. Secondly, the data is scaled using MinMaxScaler from scikit-learn. Here, every feature is limited to the range (0, 1), thereby eliminating different scales between the features, which could lead to distortion. The estimators for these two steps are trained solely on the training data and then used on the test dataset as well. The models are then trained on the training dataset and evaluated using the test dataset.

4.4 Algorithms used

In this study, five different ML-algorithms are used. The first model is linear regression using the main DoE factors only. The python package "statsmodels" [25], version 0.14.0, is

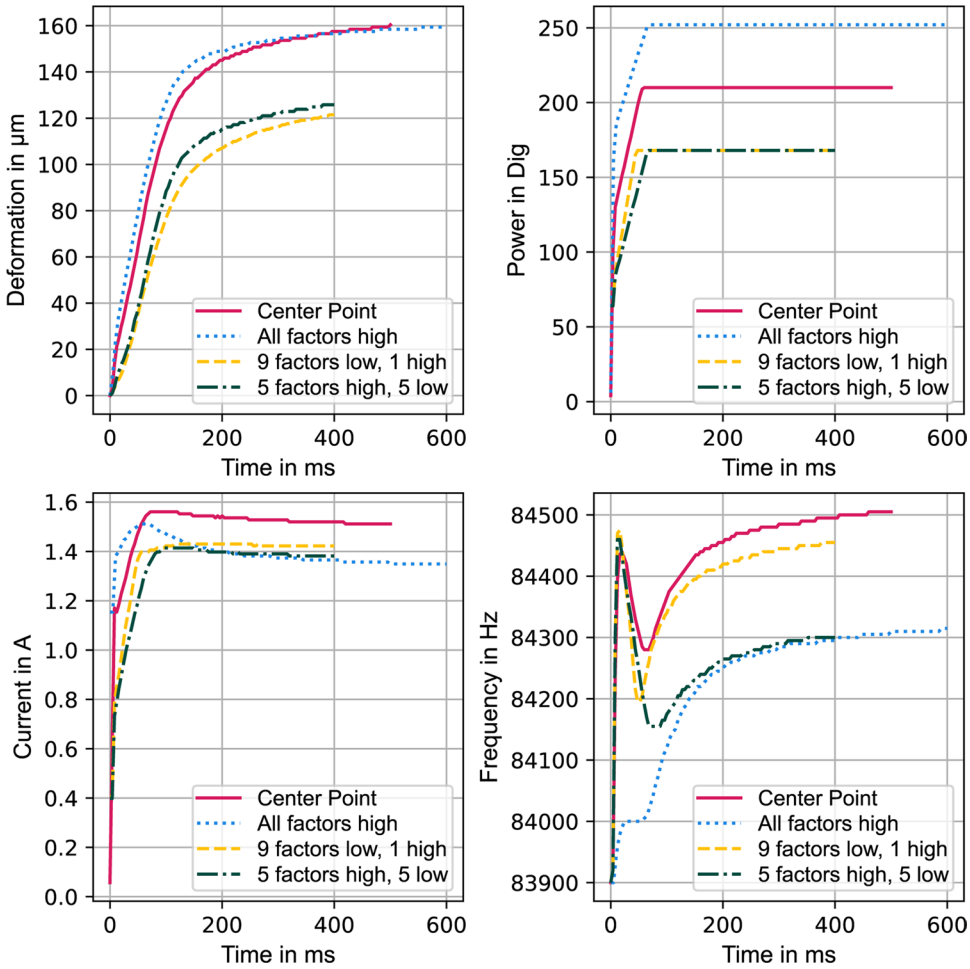

Fig. 7 Examples of recorded time-series for four different factor settings: deformation (top left), power (top right), current (bottom left), and frequency (bottom right)

used for this model. The remaining four models are implemented using the package "scikit-learn" [26], version 1.3.0. These are linear support vector regression (linear SVR), random forest regression (random forest), AdaBoost regression (AdaBoost), and multi-layer perceptron regression (MLP). Table 3 shows an overview of the models and their hyper-parameters used for this work. The hyper-parameters are set based on experience by the authors, as well as recommendations by the scikit-learn library. When possible, random_state is set to zero to ensure reproducibility.

Table 3 Hyper-parameter settings of the models used in the examination

Model	Parameter	Setting
Linear regression	intercept	true
Linear SVR	dual	false
	loss	squared_epsilon_insensitive
	max_iter	1000
Random forest	n_estimators	500
	criterion	absolute_error
	max_depth	10
	max_features	sqrt
AdaBoost	n_estimators	25
	loss	exponential
	learning_rate	2.0
MLP	hidden_layer_sizes	(100, 50)
	batch_size	16
	learning_rate	adaptive
	learning_rate_init	0.01
	max_iter	500

5 Results and discussion

This section presents the results for the implementation described in the previous section.

5.1 Performance results on DoE parameters

First, the models from Section 4 are trained using only the varied factors from the DoE. Table 4 shows the resulting

Table 4 Performance on DoE-parameters, averaged across 20 different train-test splits. The best model is presented in bold

Model	R^2-score	*MAE* in cN
Linear regression	**0.774 ± 0.029**	**355 ± 21**
Linear SVR	**0.775 ± 0.028**	**355 ± 20**
Random forest	0.760 ± 0.030	361 ± 22
AdaBoost	0.754 ± 0.034	380 ± 28
MLP	0.723 ± 0.054	385 ± 47

Table 5 Performance on all measured parameters, averaged across 20 different train-test splits. The best model is presented in bold

Model	R^2-score	*MAE* in cN
Linear regression	0.415 ± 0.267	494 ± 78
Linear SVR	0.784 ± 0.182	319 ± 32
Random forest	0.821 ± 0.024	316 ± 21
AdaBoost	0.815 ± 0.022	324 ± 20
MLP	**0.863 ± 0.026**	**279 ± 30**

R^2-scores and *MAE* values for the test set. The models are trained 20 times, each time with a different train-test-split. Thus, the results contain the mean and standard deviation across these 20 splits. To ensure comparability, the 20 splits are identical across all models and evaluations in the following subsections. The table shows that linear regression and linear SVR perform best, both explaining slightly more than 77% variation and yielding a *MAE* slightly below 360 cN.

Additionally, random forest and AdaBoost show promising results as well, with a R^2-score above 75%. Unexpectedly, the MLP is only on rank five. This is most likely due to the selected network architecture not yet optimized for the given problem.

Overall, all models show similar R^2-score, with differences only within the second decimal. The similarity is likely caused by the fact that the features used (the varied DoE factors) only convey a limited amount of information regarding the shear force, which all models almost completely utilize. This suggests that additional features are needed for a more accurate shear force prediction.

5.2 Performance results on all features

The model performance is evaluated, when using all available features. These include machine and process information, as well as the varied DoE-parameters, as described in Section 4. Table 5 shows the performance results for the five models, averaged across 20 different train-test-splits. The table shows that the MLP model performs the best, with an explained variance of 86.3% and a *MAE* of 279 cN.

The performance for all models except linear regression has increased by using all possible features, when compared to the results from the previous section. The performance for linear regression has dropped sharply to 41% explained variance and a *MAE* of 495 cN. Additionally, linear regression has a large standard deviation. These poor results are due to overfitting on the 2090 features present in the training set. The model shows a very good performance on the training set, but fails to generalize, therefore, yielding a poor performance on the test set. This is likely caused by the high dimensionality of the model, with 2090 parameters. For linear SVR the gain in explained variance is minimal, with only around 1 percentage point, but a large increase in standard deviation (from 2 to 18%) is observed. However, the *MAE* is significantly lower using all measured parameters. The large standard deviation in R^2 stems from a single split, where the linear SVR has an R^2 of 0.000 on the test set. Removing this split lowers the standard deviation to 0.026, which is in range of the performance of other algorithms. The largest increase in performance is visible for the MLP. Here R^2 increased by 14 percentage points from 0.72 to 0.86 and *MAE* decreased by 106 cN from 385 to 279 cN. Comparing the best model in the previous experiment (linear SVR) with the best model of this experiment (MLP), an increase of 9 percentage points in R^2 and a decrease of 66 cN in *MAE* is visible. Thus, using all measured parameters, a significant increase in performance is observed. This underlines the hypothesis that machine and process data gathered during the bonding process can lead to a more accurate prediction of bond shear force.

5.3 Feature selection using recursive feature elimination

5.3.1 Results of recursive feature elimination

By using recursive feature elimination with cross-validation (RFECV), it is possible to reduce the number of features from 2090 to a smaller number, thereby lowering model complexity needed for a prediction. The dataset is again split into train and test dataset. Then, using a random forest regression model with 500 estimators, a maximum depth of 10, a maximum number of features per split determined by the square root, and the absolute error as optimization criteria per tree, RFECV is applied on the training data. For each train-test-split, RFECV is applied using fivefold cross-validation. The selected optimization criteria for RFECV is *MAE*. This process is repeated five times, each time with a different initial train-test-split. Table 6 depicts the optimal number of features determined by the RFE-algorithm, as well as the resulting model performance on the test set, for the five different train-test-splits. The table shows that for all evaluated train-test-splits, less than 180 features are needed for minimizing the absolute error in the regression. The best results are achieved for the fourth train-test-split. Here, the

Table 6 Results of RFE on different train-test-splits using random forest regression as estimator in combination with RFE. The best model is presented in bold

Train-test-split no	Optimal number of features identified by RFE	Resulting R^2-score	Resulting *MAE* in cN
1	158	0.837	336
2	171	0.853	312
3	38	0.800	327
4	26	**0.868**	**258**
5	145	0.824	287

Table 7 Performance on the 26 features identified using RFECV, averaged over 20 different train-test-splits. The best model is presented in bold

Model	R^2-score	*MAE* in cN
Linear regression	$\mathbf{0.860 \pm 0.015}$	$\mathbf{281 \pm 14}$
Linear SVR	$\mathbf{0.863 \pm 0.015}$	$\mathbf{277 \pm 13}$
Random forest	0.853 ± 0.021	285 ± 20
AdaBoost	0.829 ± 0.025	310 ± 25
MLP	0.856 ± 0.018	284 ± 15

random forest explained 86% of the variance present in the test dataset and achieved an average error of ~ 260 cN, while only needing 26 features. Therefore, these 26 identified features are selected for further analysis. An overview of the selected 26 features is shown in Table 10 (see Appendix).

5.3.2 Performance results on reduced features

Table 7 shows the results for the models used in Section 5.2 on the selected 26 features, averaged across 20 train-test-splits. The results show an increase in performance for both R^2-score and *MAE* for all models except MLP, which shows a similar performance as before. Comparing the best performing model from Section 5.2 (MLP) with the performance of the best models on the 26 identified features (linear regression and linear SVR), an equal performance is observed. The standard deviation is around 2 times smaller, for both R^2 and *MAE*. This suggests that the subset identified by the RFE contains most of the information needed for the source bond shear force prediction. The MLP model is among the top three models, when comparing the results depicted in Table 7, with a performance equal to linear SVR and linear regression. The next step (see Section 5.4) is to tune the hyper-parameters of the models, which should lead to a further increase in model performance.

Similar to Table 4, all models show a similar R^2-score, with differences only in the second decimal. As the number of features is rather small, all models are able to predict the shear force more accurately. Thus, the similarity is likely caused by the fact that the hyper-parameters are not tuned yet.

5.3.3 Comparison to other feature subsets

As mentioned in Section 3, different feature subsets have been proposed by other authors for shear force prediction. A common approach is to use features extracted from the current or voltage signals measured in the transducer during the bonding process. In our experiment, both voltage and current are recorded during bonding. Among the 2090 features in our approach, some stem from these time-series. Therefore, it becomes possible to compare the 26 features identified by RFE with features solely from current and/or voltage.

Table 8 shows a comparison of the performance for different models for the 26 features from RFE, the features using voltage data only (467 features), using current data only (532 features) and both current and voltage data (999 features). For reference, the performance using only the varied DoE parameters is shown in the last column. The table shows that solely using features from voltage or current yields to results worse than using only the varied DoE parameters. When using both current and voltage data simultaneously, the best model shows a performance similar to the varied DoE parameters model only.

However, the best model for the 26 features identified by the RFE is significantly better than any of the other subset-model combinations, with an R^2 above 86% and an *MAE* smaller 280 cN. This suggests that the 26 features identified by the RFE, which stem from both machine and process data, contain more information regarding the shear force of the source bond than solely using current or voltage information. Additionally, the results show that all models perform better using only the 26 RFE features than on any other feature subset, further indicating that RFE selected the most valuable features.

5.4 Performance results after tuning of hyper-parameters

Lastly, the hyper-parameters of the models are tuned to further increase model accuracy. A grid search is conducted for all algorithms across their most relevant hyper-parameters. The examined parameter grids, along with the identified optimal model hyper-parameters, are depicted in Table 11 (see Appendix). The 26 features identified by RFE in Section 5.3 are used in this examination. Similarly, fivefold cross validation is used during grid search. Both R^2-score and *MAE* are selected as optimization metrics during grid search. Using the tuned hyper-parameters, the models are re-trained and evaluated 20 times, using different 80–20 train-test-splits. The averaged results and the standard deviation across these

Table 8 Comparison of model performance for different feature subsets, averaged over 20 different train-test-splits. The first column shows the performance using the 26 features identified by RFE. The second column shows the performance using the features from the voltage measured in the transducer only. This is followed by current features (third column) and current and voltages features (forth column). The last column shows performance using only the varied DoE parameters for reference. The best model for each feature subset is italicized. The best overall model is presented in bold. The models are (1) linear regression, (2) linear SVR, (3) random forest, (4) AdaBoost, and (5) MLP

	26 features from RFE		Voltage features only		Current features only		Current and voltage features		Varied DoE parameters only	
Model	R^2-score	*MAE* in cN	R^2-score	*MAE* in cN	R^2-score	*MAE* in cN	R^2-score	*MAE* in cN	R^2-score	*MAE* in cN
1	0.860 ± 0.015	281 ± 14	0.703 ± 0.027	404 ± 20	0.000 ± 0.000	4593 ± 1210	0.000 ± 0.000	4853 ± 1700	0.774 ± 0.029	355 ± 21
2	***0.863 ± 0.015***	***277 ± 13***	*0.704 ± 0.027*	404 ± 20	0.683 ± 0.065	394 ± 13	0.705 ± 0.060	388 ± 17	*0.775 ± 0.028*	*355 ± 20*
3	0.853 ± 0.021	285 ± 20	0.701 ± 0.026	*402 ± 21*	*0.747 ± 0.040*	*363 ± 22*	*0.770 ± 0.031*	*353 ± 23*	0.760 ± 0.030	361 ± 22
4	0.829 ± 0.025	310 ± 25	0.668 ± 0.036	440 ± 21	0.726 ± 0.039	382 ± 24	0.740 ± 0.032	377 ± 21	0.754 ± 0.034	380 ± 28
5	0.856 ± 0.018	284 ± 15	0.687 ± 0.046	418 ± 39	0.715 ± 0.034	392 ± 26	0.742 ± 0.029	379 ± 32	0.723 ± 0.054	385 ± 47

Table 9 Performance on the 26 features identified using RFECV after tuning the model hyper-parameters, averaged across 20 different train-test-splits. For reference, the non-tuned linear regression model is included. The best model is presented in bold

Model	R^2-score	*MAE* in cN
Linear regression (not tuned)	0.860 ± 0.015	281 ± 14
Linear SVR	0.863 ± 0.015	277 ± 13
Random forest	0.863 ± 0.021	275 ± 20
AdaBoost	0.875 ± 0.018	264 ± 18
MLP	**0.896 ± 0.018**	**241 ± 16**

20 splits are shown in Table 9. For comparison, the non-tuned linear regression is also depicted. The table shows that MLP yields the best performance after hyper-parameter tuning, with an explained variance of almost 90% and a *MAE* of 241 cN. Comparing the tuned MLP model to the best non-tuned model in Table 7 (linear SVR) shows an increase in R^2-score of three percentage points and a decrease in *MAE* of 36 cN. Furthermore, comparing each tuned model with its non-tuned counterpart in Table 7 shows that no increase in performance is gained for the linear SVR by tuning the hyper-parameters. This indicates that the potential is already reached prior to tuning the hyper-parameters for the regression problem at hand. The biggest performance increase is observed for AdaBoost. Here, R^2 increased by six percentage points, and *MAE* decreased by 43 cN by tuning the hyper-parameters.

Figure 8 shows the predictions versus actual values for one train-test-split of the tuned MLP model. The blue crosses depict the performance on the training data, while the red dots show performance on the test set. The black dashed line shows the ideal performance (no errors between predictions and actual values). The figure shows that the predictions generally lie close to the dashed line. This model has an *MAE* of 232 cN and a standard deviation of 221 cN for the predictions. The model is trained for 250 epochs, using a batch size of eight and the lbfgs solver. The learning rate is set to "adaptive," with an initial learning rate of 0.001, and the model contains three hidden layers with 800, 800, and 50 neurons, respectively.

5.4.1 Real-time capability

Lastly, the run-time for using the trained model in production (see Fig. 6) is measured on an Intel i7-8565 CPU with 16 GB RAM. The measurement is repeated ten times to account for variation in background processes. On average, the process of predicting the shear force, described in Fig. 6, is executed in 10.75 ms, with a standard deviation of 1.23 ms. The bonding time in the DoE varied between 400 and 600 ms. Thus, it can be concluded that our approach is real-time capable, as the model prediction is available within a fraction of the actual bonding time.

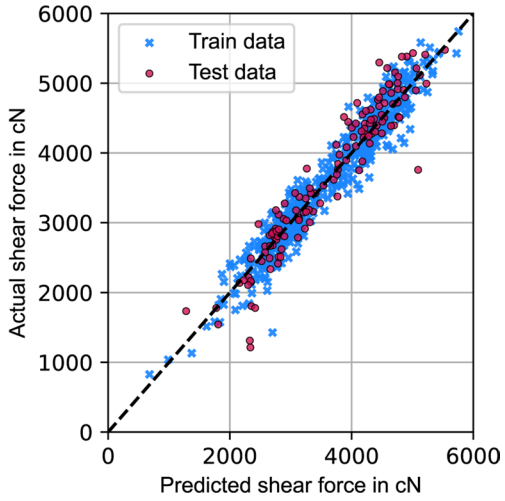

Fig. 8 Predicted vs. actual shear force for one train-test-split of the tuned MLP model. The points indicated by blue crosses show performance on training data, while red dots indicate performance on test data. The black dashed line marks ideal performance (error = 0)

6 Conclusions

This examination showed that the source bond shear force can be predicted with high precision using a neural network, using only parameters that are automatically recorded during bonding by the utilized bonding machine. Furthermore, the results show that features from both the recorded time-series and the machine data contain useful information on the bond quality. After fine-tuning the models, our real-time capable, sensorless approach achieves high prediction quality in ultrasonic copper wire bonding, achieving an average *MAE* of 241 cN and 89.6% R^2-score across 20 different train-test splits. The approach can be applied to other materials and/or processes by performing the mentioned steps again. Alternatively, algorithms from few-shot learning, especially meta-learning, might be used to decrease the manual effort needed to transfer the model to other parameter combinations. In the future, we aim to enhance our method by investigating how it can be efficiently transferred to different materials and bonding settings, using generalization techniques like meta-learning.

The approach only estimates the quality of the source bond. However, the destination bond can also fail. Thus, future research could expand the methodology to predict the shear force of both source and destination bond. By carrying out the steps described above, a similarly good prediction can be expected for the destination bond, as the processes are transferable. In this study, the data set is relatively small. A further increase in model robustness and prediction quality is expected from a larger dataset.

The shear force prediction forms the basis for an automated real-time bond quality estimation. By setting a lower specification limit for the shear force, the proposed model can be extended to return a class label regarding the bond quality. Future research should focus on the transferability of the approach for parameter combinations outside the investigated DoE in this examination. In addition, it should be evaluated how good the prediction is when disturbance variables such as contamination of the substrate surface occur.

Appendix

Table 10 Overview of 26 optimal features identified by RFE and their origin

No	Feature	Origin
1	Y-position of bond	Machine data
2	Time value at which 50% deformation is reached	Process data
3	Peak-to-peak voltage in ultrasonic generator	Machine/process data
4	Electric power in ultrasonic generator	Machine/process data
5	0.1 quantile of voltage	Extracted from time-series
6	Voltage CWT-coefficients	Extracted from time-series
7–14	Voltage FFT-coefficient	Extracted from time-series
15	Linear trend of voltage	Extracted from time-series
16–17	Deformation CWT-coefficients	Extracted from time-series
18	Deformation FFT-coefficient	Extracted from time-series
19	Current sum of reoccurring values	Extracted from time-series
20	Current CWT-coefficients	Extracted from time-series
21–22	Current aggregated linear trend	Extracted from time-series
23–25	Frequency FFT-coefficient	Extracted from time-series
26	Frequency aggregated linear trend	Extracted from time-series

Table 11 Overview of searched parameter grids in hyper-parameter tuning. The optimal hyper-parameters within this grid are displayed in the third column

Model	Parameter grid used in hyper-parameter tuning	Optimal model parameters
Linear SVR	C: [0.1, 0.5, 1, 2, 10] loss: [epsilon_insensitive, squared_epsilon] dual: [auto, false, true] max_iter: [100, 1000, 10,000, 100,000]	C = 2 loss = squared_epsilon dual = true max_iter = 1000
Random forest	n_estimators: [10, 100, 1000] criterion: [squared_error, absolute_error, friedman_mse, poission] max_depth: [none, 5, 10, 20, 50, 100] min_samples_split: [1, 2, 3, 4, 5, 0.1, 0.2] min_samples_leaf: [1, 2, 5] max_features: [1.0, sqrt, log2, none]	n_estimators = 100 criterion = poisson max_depth = 20 min_samples_split = 2 min_samples_leaf = 1 max_features = none
AdaBoost	estimator: [DecisionTreeRegressor(), RandomForestRegressor()] n_estimators: [5, 10, 25, 50, 100] learning_rate: [0.5, 1, 1.5, 2, 5] loss: [linear, square, exponential]	estimator = RandomForestRegressor() n_estimators = 100 learning_rate = 2 loss = exponential
MLP	hidden_layer_sizes: [(100, 50), (100, 100, 50), (200, 100, 50), (26, 12, 4), (800, 200, 50), (200, 200, 50), (800, 400, 50), (800, 800, 50)] solver: [lbfgs, adam] batch_size: [4, 8, 16, 32, 64, 128] learning_rate: [adaptive] learning_rate_init: [0.00001, 0.0001, 0.001, 0.01] max_iter: [100, 250, 500]	hidden_layer_sizes = (800, 800, 50) solver = lbfgs batch_size = 8 learning_rate = adaptive learning_rate_init = 0.001 max_iter = 250

Funding Open Access funding enabled and organized by Projekt DEAL. This work was funded by the Bavarian State Ministry for Economic Affairs, Regional Development and Energy as part of the Bavarian Joint Funding Program (BayVFP)-Digitalization funding line-Information and communication technology funding area under grant DIK0460.

Data availability Not applicable.

Code availability Not applicable.

Declarations

Consent for publication All authors have provided their consent for publication of this paper.

Conflict of interest The authors declare no competing interests.

References

1. Kao S-X, Chien C-F (2023) Deep learning based positioning error fault diagnosis of wire bonding equipment and an empirical study for IC packaging. IEEE Trans Semicond Manuf. https://doi.org/10.1109/TSM.2023.3243775
2. Kästle C (2018) Qualifizierung der Kupfer-Drahtbondtechnologie für integrierte Leistungsmodule in harschen Umgebungsbedingungen. FAU University Press. https://doi.org/10.25593/978-3-96147-146-1
3. Harman GG (2010) Wire bonding in microelectronics, 3. ed. [fully updated]. McGraw-Hill, New York, NY
4. Feng W, Meng Q, Xie Y et al (2011) Wire bonding quality monitoring via refining process of electrical signal from ultrasonic generator. Mech Syst Signal Process 25:884–900. https://doi.org/10.1016/j.ymssp.2010.09.010
5. Akchurin N, Damgov J, Dugad S et al. (2022) Deep learning applications for quality control in particle detector construction. arXiv preprint arXiv:2203.08969. https://doi.org/10.48550/arXiv.2203.08969
6. Guan XD, Chen G, Liang WL (2012) The optimization of wire bonding parameters and quality prediction model based on neural network. Advanced Mater Res 542:976–980. https://doi.org/10.4028/www.scientific.net/AMR.542-543.976
7. Zhang D, Ling S-F (2003) Monitoring wire bonding via time-frequency analysis of horn vibration. IEEE Trans Electron Packag Manuf 26:216–220. https://doi.org/10.1109/TEPM.2003.820800
8. Ling S-F, Zhang D, Yi S et al (2006) Real-time quality evaluation of wire bonding using input impedance. IEEE Trans Electron Packag Manuf 29:280–284. https://doi.org/10.1109/TEPM.2006.887400
9. Arjmand E, Agyakwa PA, Corfield MR et al (2016) Predicting lifetime of thick al wire bonds using signals obtained from ultrasonic generator. IEEE Trans Components, Packag Manuf Technol 6:814–821. https://doi.org/10.1109/TCPMT.2016.2543001
10. Lindamood L, Mohr L, Moghaddas A et al (2021) Investigation of monitoring methods for ultrasonic metal welding. Sensors Smart Struct Technol Civil, Mech Aerospace Syst 11591:62–70. https://doi.org/10.1117/12.2583121
11. Wang C-S, Chiu J-E (2022) Shear force classification before wire bonding based on probe mark 2-D images using machine learning

methods. IEEE Trans Semicond Manuf 35:210–219. https://doi.org/10.1109/TSM.2022.3162401
12. Long Z, Zhou X, Zhang X et al (2019) Recognition and classification of wire bonding joint via image feature and SVM model. IEEE Trans Components, Packag Manuf Technol 9:998–1006. https://doi.org/10.1109/TCPMT.2019.2904282
13. Chen J, Zhang Z, Wu F (2021) A data-driven method for enhancing the image-based automatic inspection of IC wire bonding defects. Int J Prod Res 59:4779–4793. https://doi.org/10.1080/00207543.2020.1821928
14. Gaul H (2009) Berechnung der Verbindungsqualität beim Ultraschall-Wedge-Wedge-Bonden. Dissertation, Technische Universität Berlin (Prüfungsjahr 08.05.2009)
15. Tsai T-N (2013) An adaptive diagnosis system for copper wire bonding process control and quality assessment in integrated circuit assembly. Int J Comput Integr Manuf 26:513–526. https://doi.org/10.1080/0951192X.2012.731614
16. Yu C-M, Lai K-K, Chen K-S et al (2020) Process-quality evaluation for wire bonding with multiple gold wires. IEEE Access 8:106075–106082. https://doi.org/10.1109/ACCESS.2020.2998463
17. Huai H, Chidanandappa N, Wilde J (2023) Application of machine learning to recognize wire bond lift-off in power electronics manufacturing. In: 2023 24th International Conference on Thermal, Mechanical and Multi-Physics Simulation and Experiments in Microelectronics and Microsystems (EuroSimE). IEEE, pp 1–6. https://doi.org/10.1109/EuroSimE56861.2023.10100782
18. Siebertz K, van Bebber D, Hochkirchen T (2017) Statistische Versuchsplanung: design of experiments (DoE), 2. Auflage. VDI-Buch. Springer Vieweg, Berlin, Heidelberg. https://doi.org/10.1007/978-3-662-55743-3
19. Chang T-C, Chen K-S (2018) Testing process quality of wire bonding with multiple gold wires from viewpoint of producers. Int J Prod Res 57:5400–5413. https://doi.org/10.1080/00207543.2018.1524164
20. Henderson T, Fulcher BD (2021) An empirical evaluation of time-series feature sets. In: 2021 International Conference on Data Mining Workshops (ICDMW). IEEE, pp 1032–1038. https://doi.org/10.1109/ICDMW53433.2021.00134
21. Chen R-C, Manongga WE, Dewi C (2022) Recursive feature elimination for improving learning points on hand-sign recognition. Future Internet 14:352. https://doi.org/10.3390/fi14120352
22. Feng W, Zhang Y, Meng Q (2012) Monitoring of the ultrasonic wire bonding quality based on system impedance. In: 2012 International Conference on Computer Science and Service System. IEEE, pp 2013–2016. https://doi.org/10.1109/CSSS.2012.501
23. Feng W, Chen X, Wang C et al (2021) Application research on the time–frequency analysis method in the quality detection of ultrasonic wire bonding. Int J Distrib Sens Netw 17:155014772110183. https://doi.org/10.1177/15501477211018346
24. Christ M, Braun N, Neuffer J et al (2018) Time series feature extraction on basis of scalable hypothesis tests (tsfresh – a Python package). Neurocomputing 307:72–77. https://doi.org/10.1016/j.neucom.2018.03.067
25. Seabold S, Perktold J (2010) Statsmodels: econometric and statistical modeling with python. In: Proceedings of the 9th Python in Science Conference, 57, 10–25080. https://doi.org/10.25080/Majora-92bf1922-011
26. Pedregosa F, Varoquaux G, Gramfort A et al. (2011) Scikit-learn: machine learning in Python. Journal of Machine Learning Research 12:2825–2830. https://scikit-learn.org/stable/about.html#citing-scikit-learn. Accessed 26 Jun 2024