Tribology International 200 (2024) 110067

Contents lists available at ScienceDirect

Tribology International

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

Full Length Article

On the use of machine learning for predicting femtosecond laser grooves in tribological applications

Luis Moles [a,c,*], Iñigo Llavori [b], Andrea Aginagalde [b], Goretti Echegaray [c], David Bruneel [d], Fernando Boto [e], Alaitz Zabala [b]

[a] TECNALIA, Basque Research and Technology Alliance (BRTA), Paseo Mikeletegi 2, Donostia, 20009, Guipuzcoa, Spain
[b] Mondragon Unibertsitatea, Faculty of Engineering, Mechanics and Industrial Production, Loramendi 4, Mondragon, 20500, Guipuzcoa, Spain
[c] Department of Computer Sciences and Artificial Intelligence, University of the Basque Country (UPV/EHU), Paseo Manuel Lardizabal 1, Donostia, 20018, Guipuzcoa, Spain
[d] LASEA, Liege Science Park, Rue Louis Plescia 31, Seraing, 4102, Belgium
[e] Faculty of Engineering, University of Deusto, Mundaitz Kalea 50, Donostia, 20012, Guipuzcoa, Spain

ARTICLE INFO

Keywords:
Femtosecond laser
Inverse modelling
Machine learning
Stamping
Surface texturing

ABSTRACT

Femtosecond laser surface texturing is gaining increased interest for optimizing tribological behaviour. However, the laser surface texturing parameter selection is often conducted through time-consuming and inefficient trial-and-error processes.

Although machine learning emerges as an interesting option, multitude of models exists, and determining the most suitable one for predicting femtosecond laser textures remains uncertain. Furthermore, the absence of open-source implementations and the expertise required for their utilization hinders their adoption within the tribology community.

In this study, two novel inverse modelling approaches for the optimal prediction of femtosecond laser parameters are proposed, based on the results of a comparison between six different machine learning models conducted within this research. The entire development relies on open-source tools, and the models employed are shared, with the aim of democratizing these techniques and facilitating their adoption by non-expert users within the tribology community.

1. Introduction

The problems associated with tribology are responsible for the 23% of the world's energy consumption, and it is estimated that 40% of this energy can be reduced within 15 years by implementing existing knowledge and techniques to reduce friction and wear [1]. Consequently, significant research efforts are directed towards strategies aimed at optimizing tribological behaviour. Surface texturing has garnered increasing interest within the tribological research community and can be defined as the incorporation of specific surface features (grooves, discrete dimples...), which generate well-defined patterns [2]. The benefits of textured surfaces can be summarized as follows: (i) they act as hydrodynamic micro-bearings that support the applied load, (ii) they behave as reservoirs under the limit and mixed lubrication regime (preventing starvation and seizure), and (iii) trap the wear debris and contaminants (reducing 3rd body abrasion) [3]. All this contributes improving tribological properties, but the geometry, size, depth, density, and cross-section of the features must be carefully selected and manufactured [4].

Laser Surface texturing (LST) is the preferred choice for surface texturing due to the benefits it offers compared to other current techniques: processing speed, high efficiency, environmentally friendly nature, high precision, and the capability of manufacturing complex textures [2,5]. Specifically, one of the lasers that gained more and more attention in recent years is the femtosecond (fs) laser, since it offers increased processing precision, minimal thermal damage to the processed area, compatibility with a wide range of materials, and high processing quality, eliminating the need of any mechanical or chemical post-processing after texturing [6].

The micro and nanostructured surfaces produced by ultrafast laser irradiation provide beneficial properties in terms of friction, adhesion, optical absorption, and hydrophobicity.

In laser surface texturing, several parameters play a crucial role in achieving the desired results. Here are the key parameters involved in each category:

- Beam parameters: wavelength, pulse frequency, pulse duration, pulse energy, beam and beam shape.

* Corresponding author at: TECNALIA, Basque Research and Technology Alliance (BRTA), Paseo Mikeletegi 2, Donostia, 20009, Guipuzcoa, Spain.
E-mail address: luis.moles@tecnalia.com (L. Moles).

https://doi.org/10.1016/j.triboint.2024.110067
Received 24 May 2024; Received in revised form 20 July 2024; Accepted 2 August 2024
Available online 8 August 2024

- Focusing parameters: focal length and focal spot size.
- Scanning head parameters: scanning speed, scans overlapping, line pitch and number of layers.
- Material and surface properties: optical properties, reflectivity, thermal conductivity, threshold for ablation, composition and roughness.

Searching the optimum parameter combination becomes a tedious activity in laser processing, since multitude potential process parameter combinations exist, and trial and error procedures are very time-consuming [7].

To improve the parameter searching process, is very important to know the effect of each of these parameters on the LST. That is why many researchers like Benton et al. [8], have analysed the effect of some laser processing parameters like the power, laser defocusing distance, scanning speed, material thermal conductivity, specific heat, and the effect of the convective heat transfer on the width and depth of the machined channer. Campanelli et al. [9] analysed the effect of three main process parameters (average laser power, scanning speed and frequency) on shape geometry and roughness of parts fabricated by laser ablation, calculated for the single considered geometrical entities, and on surface roughness irradiating a Ti6Al4V titanium alloy with a nanosecond laser. It was demonstrated that the wall shape angles, surface roughness and an error index (ER%) can be predicted changing the above mentioned parameters. Pou-Álvarez et al. [10] studied the role of pulse length on topography and corrosion properties of AZ31 magnesium alloy with nanosecond, picosecond and femtosecond lasers, while Ezhilmaran et al. [11] studied the influence of laser wavelengths on surface morphology of dimples and other dimple parameters on film thickness, observing an improvement in tribology characteristics with texturing.

Accordingly, predictive analytics [12] and numerical models [13] have been developed to simulate the laser ablated areas. Those models, however, can be computationally expensive and require input material properties that might not be readily obtainable, thus being not practical for industrial processing facing different material alloys or coated solutions for example.

Therefore, tools such as machine learning or artificial intelligence are an option to consider for laser process optimization to obtain the appropriate laser parameters [14,15]. These methods allow to foresee the values of the laser parameters that will be used to create a certain texture in a certain material without knowing all the material properties needed for the analytical forecast. Desai and Shaikh [16] studied the effect of variation of cutting power and speed on the laser textured patterns depth. The depth values taken experimentally, were compared with the ones obtained with the prediction of semi-analytical model, multi-gene genetic programming (MGGP) model and artificial neural network (ANN) model. Power, as opposed to speed, has been observed to have a linear effect on texture depth. All the prediction models, had good accuracy in the depth prediction, but the one that made the best prediction in the validation points was ANN.

Although there has been done many works analysing the effect of the laser parameters on the process, there is still work to do on the parameter prescription. In addition, it is not clear which ML model is the most suitable for prescripting process parameters of a femtosecond laser. Moreover, ML models usually require extensive expert knowledge to achieve a robust performance (data cleaning, feature selection, hyperparameter optimization), which make them complex to use for non-expert users. Therefore, this study has two main objectives. On the one hand, it will focus on comparing different ML models to predict the depth of the cavity with certain parameters of the laser and vice versa, improving the yield of the machine and reducing overhead costs. On the other hand, it intends to provide a tool for non-expert users to use ML models for the prediction of process parameters.

The central focus of this paper is depth prediction, driven by the pivotal role that lubrication film thickness plays in tribology applications [17].

The remainder of the article is organized as follows. An introduction to Machine Learning is presented in Section 2. All of the methods used are explained in more detail in Section 3. Section 4 presents the results of the Machine Learning strategy for depth prediction models, and 5 shows the conclusions and possible lines of future work.

2. Background

Machine Learning can be described as an Artificial Intelligence field that enables to find patterns and make predictions from various data through the use of different algorithms. Machine Learning is usually divided into two main groups: supervised and unsupervised.

Supervised learning refers to the problems where a set of labelled input–output examples is used to estimate a function that maps a given input to an output. Within supervised learning, a distinction can be made between classification problems, where the predicted output is categorical, and regression problems, where the output to be predicted is a numerical value [18].

On the other hand, problems where the output of the given data is unknown are referred to as unsupervised learning problems. The objective of unsupervised learning is to group similar data into different classes [19].

In this study, a supervised regression problem is addressed. The objective is to predict the depth of different typical shapes used for tribological applications: lines and dimples. The input variables for the regression model are some of the laser parameters (see Table 1 for the lines and Table 2 for the dimples).

Every Machine Learning approach requires several steps, starting from the collection of initial data to the evaluation of the generated Machine Learning model (called ML methodology).

As shown in Fig. 1, a typical Machine Learning methodology consist of five steps. The initial step of a Machine Learning project is data cleaning, which involves techniques focused on removing any duplicate or mislabelled data. Feature engineering focuses on selecting the most important variables from raw data in order to create an accurate predictive model. After this step, it is necessary to choose the most suitable Machine Learning algorithm from the available options. This is known as the model selection phase. Once the model is selected, a hyper-parameter optimization phase is required to avoid overfitting and improve predictions by finding the best configuration for the selected model. Finally, to ensure the generalization of the trained model to unseen data, and evaluate model performance, a validation with a test dataset is necessary in the model evaluation step.

Fig. 1. Methodology of a Machine Learning approach.

Machine Learning approaches offer several advantages. They enable the easy identification of trends and patterns in data and can handle higher dimensional data spaces [20].

There are several Machine Learning models. In this study, six different models are compared: Decision Trees (DT), Random Forest (RF), Gaussian Process (GP), Support Vector Machines (SVM), K-nearest Neighbours (KNN) and Artificial Neural Networks (ANN). In addition, a genetic algorithm (GA) is implemented for creating an inverse model capable of prescribing input parameters for a given depth.

Decision Trees: Decision tree based models (DT) [21] incrementally develop a tree by generating different decision rules (branches of the tree). This splits the entire data-set (root node) into smaller subsets (decision nodes), aiming to minimize the prediction error, until the model becomes confident enough to make a prediction (leaf node). In regression tasks, the resulting model returns the mean of all the points falling in the corresponding leaf node as the prediction.

The correct selection of the variable that will generate the split in each level of the tree minimizes the prediction error. The most commonly used criteria are entropy (1) and gini index (2) for classification problems, and the mean squared error for regression problems (3).

$$E(S) = \sum_{i=1}^{c} -p_i \log_2 p_i \tag{1}$$

$$Gini = 1 - \sum_{i=1}^{c} p_i^2 \tag{2}$$

where c is the number of classes of the variable and p_i represents the probability of occurrence of the ith class.

$$MSE = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \tag{3}$$

where n represents the number of data points, y_i the actual target value for the ith data point and $\hat{y}_i$ is the predicted value for the ith data point.

Random Forest: The Random Forest algorithm (RF) [22] builds multiple decision trees and merges them together in order to get a more accurate prediction. This technique is mainly based on the concepts of bagging and the random selection of attributes.

Bagging [23] combines several instances of estimators that have been built on random samples from the original training set and aggregates the individual predictions to obtain a single prediction. The process involves the following steps:

- The original data set is divided into different subsets. These subsets form uniform samples with the same size as the training set but may not contain all the individuals, as some of them are repeated in the samples.
- A predictive model is created with each of the subsets, resulting in different models.
- Finally, a single predictive model is built by averaging all the generated models, reducing the variance compared to the individual models.

Gaussian Process: Gaussian Process (GP) [24] is a statistical technique for modelling relationships between inputs and outputs. It is based on the idea that the relationship between the inputs and outputs is not deterministic, but rather a random function that follows a particular pattern. This pattern is described by a covariance function that determines how similar pairs of input values are expected to be.

The GP assumes that any two points on the output function have a joint Gaussian distribution. This means that the distribution of possible functions that fit the data is a normal distribution. The mean of this distribution represents the expected output value at a particular input point, while the variance represents the uncertainty around that prediction.

GP regression is used to estimate the mean and variance of the distribution of possible functions that fit the data. This is achieved by finding the best-fitting function that passes through the observed data points while considering the expected similarity of input values based on the covariance function.

The covariance function used in GP regression is typically a squared exponential function, which describes how similar pairs of inputs are expected to be. The characteristic length-scale of the process, denoted by "l", determines how quickly the covariance between two input points decays with distance.

Support Vector Machines: Support Vector Machines (SVM) [25] is a machine learning algorithm used for both classification and regression tasks. In regression problems, the aim is to predict a continuous output variable given some input features.

SVM for regression works by finding the best line (or hyperplane) that separates the data points, such that the distance between the data points and the line is maximized. This distance is measured by a margin, which is the distance between the data points closest to the line. The SVM algorithm aims to maximize this margin while minimizing the prediction error.

In regression SVM, the algorithm finds a line that passes through as many data points as possible while trying to minimize the error. This line is called a "support vector regression" line as it is defined by the data points closest to it, which are called "support vectors".

The SVM algorithm also uses a "kernel function" to transform the input data into a higher dimensional space. This allows the algorithm to find a linear boundary in the higher dimensional space that matches a nonlinear boundary in the original input space.

K-Nearest Neighbours: KNN is a non-parametric, supervised learning algorithm that uses proximity to make predictions of an individual data point [26]. K is the number of neighbours taken into account to calculate the prediction. To identify the nearest neighbours of a data point, the distance between them needs to be calculated. There are different distance metrics that can be used, depending on the nature of the data. Some examples of distances are Euclidean distance, Manhattan distance or Hamming distance.

The correct choice of the K value is crucial for the algorithm's performance [27] and depends on the input data. A low value of K could lead to a model with high variance and low bias, while greater values could have the opposite effect.

Artificial Neural Networks: Artificial Neural Networks (ANN) [28] are computational models inspired by the human brain's structure and function, designed to recognize patterns and solve complex problems. They consist of layers of interconnected nodes (neurons), where each node processes input data and passes the result to the next layer. A typical neural network includes an input layer, multiple hidden layers, and an output layer. Each connection between neurons has an associated weight, which is adjusted during training to minimize prediction errors. Activation functions, such as Sigmoid, Tanh, and ReLU, introduce non-linearity, enabling the network to learn complex patterns.

Training a neural network involves a process called backpropagation [29], where errors from the output layer are propagated backward through the network to update the weights, using optimization algorithms like Stochastic Gradient Descent (SGD) or Adam. This iterative process continues until the model achieves satisfactory performance.

Genetic Algorithms: Genetic algorithms (GA) [30] reflects the process of natural selection where the fittest individuals are selected for reproduction in order to produce the offspring of the next generation. By simulating the natural evolutionary process, they can efficiently explore and exploit the search space to find high-quality solutions, and they are highly used in optimization problems. The genetic algorithm process starts with a randomly generated population of candidate solutions. Each candidate solution, known as an individual, is evaluated using a fitness function that measures how well it solves the problem at hand. The algorithm then iterates through a process that mimics natural evolutionary processes.

In the initial step, the fittest individuals are selected based on their fitness scores. These selected individuals become parents, and pairs of them are combined to create offspring. This recombination process, known as crossover, involves swapping segments of the parents' genetic material at randomly chosen points, resulting in new individuals that share traits from both parents. To introduce genetic diversity and prevent the algorithm from converging prematurely to suboptimal solutions, a probability of mutation is applied to the offspring's genetic material. This mutation process involves randomly altering some genes in the offspring. The new generation of offspring then replaces the old generation, and the fitness of the new individuals is evaluated. This process of selection, crossover, mutation, and replacement is repeated over many generations. As the algorithm progresses, the population of solutions evolves, ideally leading to increasingly better solutions and it continues until a stopping criterion is met, such as a predefined number of generations or achieving a satisfactory fitness level.

3. Materials and methods

3.1. Materials and specimens

Use cases considering both bare and coated materials at which tribological behaviour is important have been considered in the study. On the one hand, an aluminium stamping die material used in the automotive industry (1.2379/X153CrMoV12) was selected as texturing strategy is recognized as potentially interesting for deep drawing operations [31]. On the other hand, PVD (physical vapour deposition) TiN and TiCN coatings applied on biomedical titanium grade IV substrates were included as TiN and TiCN demonstrated good tribological behaviour reducing the polymeric wear for orthopaedic applications [32,33] and different laser-generated patterns on different materials have been analysed in this field [34,35].

3.2. Laser surface texturing

For laser processing, a commercial Yb:YAG femtosecond laser amplifier system was used (Satsuma HP2, Amplitude Systémes: t = 280 fs pulse duration, l = 1030 nm wavelength). The samples were mounted on a motorized x–y–z linear translation stage and placed perpendicular to the incident laser beam. The beam was shaped by a LS-Shape module, achieving a Gaussian-like beam profile with a radius of w0 (1/e2) = 10.64 μm, whereas the incident laser beam was controlled by the LS-Scan module, allowing the good positioning and speed precision in the laser processing. Both modules, LS-Scan and LS-Shape, were developed by Lasea. Apart from the laser machine, it has been used the KYLA™ software to control the machine and all the parameters relating the laser processing.

Two different texture shapes have been generated, lines and dimples, very used textures to achieve tribological improvements [36,37].

For line shape processing, they were machined in one millimetre long with 10 passes maintaining the same laser processing direction.

Due to the importance of obtaining a flat dimple bottom to optimize tribological functionality [38], a layered cross hatched pattern strategy has been followed. Fig. 2 shows the details of the strategy, with a layer sequences that rotates 90° on each processing layer (hatch pitch = 4 μm).

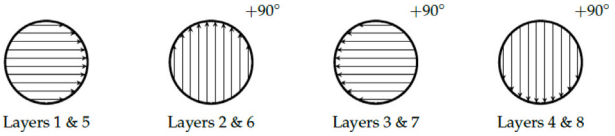


Fig. 2. Schematic representation of the employed scanning strategy.

A full factorial design of experiment (DoE) approach was followed to generate all the experimental data required for the machine learning model input. Three laser process parameters at four levels were selected for each morphology. Among all the femtosecond laser process parameters, laser fluence, scanning speed and pulse repetition rate are considered critical [39]. Laser fluence is defined as the energy of the laser pulse applied over the area and it has been recognized as a key parameter in the ablated depth [40] which is related to the fact that a minimum fluence value (material's threshold) has to be surpassed to remove material from the surface. Therefore, pulse energy is one of the most influential parameters for material ablation [41]. On the other hand, repetition rate and scanning speed are directly related to the surface quality and processing time [42]. Accordingly, these three parameters (Pulse repetition rate, pulse energy and scanning speed) were selected as process parameter variables for line machining, fixing the layers to 10 in order to minimize the experimental effort. Preliminary experiments were performed to determine a range for the parameters, selected based on the criteria of maximizing material removal rate (minimizing process time) and maintaining good surface quality (see Table 1).

For the dimple machining case, the parameters and ranges needed adjustment to ensure optimal surface and shape quality, while minimizing process time and achieving similar line depth ranges.

Table 2 summarizes the variables and levels used for dimples DoE construction. First, the scanning speed needed to be decreased, since the dimple shape may become irregular or asymmetric at high speeds due to difficulties in maintaining the accuracy of the synchronization loop between the laser and the scanning system [43]. Because of the increment in pulse overlap when scanning an area over a line (since pulses must overlap in both dimensions, length and width), the effective number of pulses is higher when processing dimples compared to lines, which is further increased due to the speed reduction required for good dimple quality. The strategy to obtain similar depth ranges obtained in lines was to match the total deposited energy per area, as it influences the ablated volume [40]. This was achieved by fixing the pulse repetition rate to the minimum value (50 kHz) and reducing the pulse energy values for dimple machining. As previously mentioned, a layered cross hatched pattern strategy has been followed to ensure a flat dimple bottom cross section, which is related to the obtained Surface quality. Accordingly, the layer number has been considered as process variable for dimple processing. Following this strategy, similar depth ranges where obtained in both dimples and lines (1–18 μm).

Table 1
Variables and levels used to build the lines DoE.

Input parameters	Unit	Level 1	Level 2	Level 3	Level 4
Pulse repetition rate	kHz	50	83.33	125	166.66
Pulse energy	μJ	5.14	7.34	9.79	12.41
Scanning speed	mm/s	100	200	300	400

Table 2
Variables and levels used to build the dimples DoE.

Input parameters	Unit	Level 1	Level 2	Level 3	Level 4
Layers	–	2	4	6	8
Pulse energy	μJ	1.57	2.98	4.7	5.84
Scanning speed	mm/s	75	100	125	150

Rest of the laser processing parameters were fixed to the values shown in Table 3 for both cases. The wavelength was fixed in the laser system, and circular polarization was selected to ensure more uniform ablation in both directions. The beam size was chosen to ensure the required minimum motive ablation and adequate focus distance, considering its effect on sensitivity to waviness or flatness errors on the surface. A Hatch pitch of 4 μm was determined to have a half effective beam radious overlap, ensuring a good dimple definition.

Three repetitions of each configuration have been generated in order to have a greater reliability. When generating Machine Learning models, the repetitions have been considered as independent configurations, so the model is able to learn the existing variance between the same parameter values. In a full factorial design within DoE, the total number of experimental trials is calculated using the formula $N = L^k$, where N is the number of trials, L is the number of levels for each factor, and k is the number of factors (process parameters). Accordingly, a total of 64 experiments were repeated three times, making a total of 192 experiments used for the training of the models.

The test set used for evaluating the predictive performance of the Machine Learning model comprised 16 additional data points, each representing specific combinations of values for the selected variables. Notably, the input values of the test points were chosen to fall between the middle values of each level used in the DoE. The thought behind this selection have been to measure the model's capability to generalize to unseen data effectively. By choosing intermediate values, it is ensured that the validation set provides a fair test of the model's interpolation capabilities within the defined feature space, which is crucial for assessing the robustness and reliability of the model. Experiments

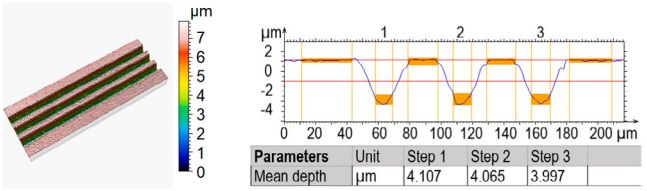
Table 3
Fixed parameters to create the experimental input data.

Process parameters	Unit	Level 1
Polarization	–	Circular
Wavelength	nm	1030
Number of layers	–	10 (lines DoE)
Pulse Repetition Rate	kHz	50 (dimples DoE)
Beam diameter	µm	21.28
Hatch pitch	µm	4 (dimples DoE)

from the test set are also repeated three times, so, the mean of the three repetitions is computed in order to compare the predicted values against the real ones.

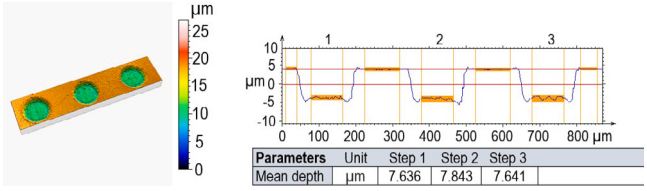
3.3. Texture characterization

A non-contact 3D optical profiler (Sensofar S-NEOX, confocal technique) was used with an objective of 20xEPI (lateral resolution = 0.91 µm, vertical resolution: 20 nm) to measure the depth of the laser engraved dimples and lines. Following the 3D areal measurements (700×525 µm^2), 2D profiles were extracted to characterize the depth in SensoMap Premium 7 metrology software (see Figs. 3 and 4).



(a) (b)

Fig. 3. (a) Detail of three femtosecond textured lines. (b) Cross section of the lines to measure the depth based on the average of the three heights.



(a) (b)

Fig. 4. (a) Detail of three femtosecond textured dimples. (b) Cross section of the dimples to measure the depth based on the average of the three heights.

3.4. Predictive model methodology

The followed methodology covers both the prediction of the depth and the laser parameters using ML, as can be seen in Fig. 5.

Firstly, different ML models are compared in order to select the one that performs better in the prediction of the depth. For each of the models, an optimization of hyperparameters is performed, so the best performance of each model is achieved. After it, the performance of all the models is tested using the R^2Score metric, and the best performing model is selected to predict the depth of the different cavities. Finally, that model is used to create an inverse model, which, given an input depth, it is able to prescript the corresponding parameters of the laser. Performance of the inverse model is measured by computing the absolute difference between the values of the prescripted input parameters and the real ones.

3.4.1. Selected models

Six different models are selected to perform the comparison: Decision Trees (DT), Random Forest (RF), Gaussian Process (GP), Support Vector Machines (SVM), K-nearest Neighbours (KNN) and Artificial Neural Networks (ANN). Each selected model covers a specific family of Machine Learning algorithms, which makes the comparison more significant: tree based models, ensemble models, statistical models, linear models, instance based models and neural network models. In addition, they are selected due to the good performance in different tribological applications and in problems with small datasets [44–46]. An explanation of each model is given in Section 2.

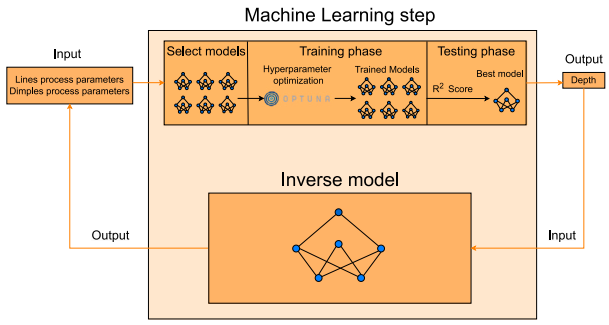


Fig. 5. Proposed Machine Learning pipeline.

3.4.2. Hyperparameter optimization

Every Machine Learning model has a set of hyperparameters that need to be tuned before testing. A hyperparameter is a parameter set before the learning process and remains constant during the entire training. These parameters control the behaviour of the learning algorithm and influence how the model is trained and the final model's performance. Hyperparameters are not learned from the data, and they need to be carefully selected and tuned to achieve the best possible performance of the model. In this work, for optimizing the hyperparameters of each model, an AutoML tool named Optuna [47] is used, which automatically performs the optimization.

A description of the hyper-parameters tuned for each model is given in Table 4.

In order to perform a fair comparison between Machine Learning methods and prevent overfitting, each model was trained using 3-Fold cross-validation. The cross-validation technique [48] divides the training dataset into k random subsets, using $k-1$ subsets for training the model, and one subset for validating it. This process is repeated k times. The selection of $k = 3$ allowed us to have folds substantial enough to be representative of the overall dataset, which is crucial for training effective models. Afterward, the test dataset mentioned in Section 3.2 is used to compare the models and evaluate their performance.

3.4.3. Evaluation metrics

The metric used to compare the performance of each model is the R^2 score. It is used in the context of a statistical model whose primary purpose is to predict future results or test hypotheses. This coefficient determines the quality of the model in replicating the results and the proportion of the variation in the results that the model can explain. Its value is usually between 0 and 1, although it can take a negative value if the model is worse than the one that only provides the mean of the dependent variable. The closer its value is to 1, the more accurate the model is. It is computed as

$$R^2Score = 1 - \frac{\sum \left(y_i - \hat{y}_i\right)^2}{\sum \left(y_i - \bar{y}\right)^2} \tag{4}$$

where y_i is the real target value for the ith data point, $\hat{y}_i$ is the predicted value for the ith data point, and $\bar{y}$ is the mean of all the observed target values y_i in the dataset.

Models' performance have been further tested using two different statistical tests. The Friedman test is used to find significant differences between the models among different datasets. It is a non-parametric test

Table 4
Hyper-parameters description.

Model	Parameters	Search Space	Description
DT	max_depth	[1–100]	Maximum depth of the tree.
	max_features	[1–3]	Maximum number of features taken into account when searching for the best split.
	min_samples_leaf	[2–50]	Minimum number of samples required to be at a leaf node.
	min_samples_split	[2–50]	Minimum number of samples required to split an internal node.
RF	n_estimators	[100–800]	Number of trees in the forest.
	max_depth	[1–100]	Maximum depth of the trees.
	max_features	[1–3]	Maximum number of features taken into account when searching for the best split.
	min_samples_leaf	[2–50]	Minimum number of samples required to be at a leaf node.
	min_samples_split	[2–50]	Minimum number of samples required to split an internal node.
GP	alpha	[1e−5–1e2]	Parameter for preventing a potential numerical issue during fitting, ensuring that the calculated values form a positive definite matrix.
SVM	gamma	[1e−3–1000]	Kernel coefficient.
	C	[1e−3–1e3]	Regularization parameter.
	epsilon	[0.001–1.0]	Specifies the value within which no penalty is associated in the training loss function with points predicted within a distance epsilon from the actual value.
KNN	n_neighbours	[1–10]	Number of neighbours required for each sample.
	weights	[uniform, distance]	Weight function used in prediction.
	p	[1,2]	Parameter for selecting the distance used (p = 1 for manhattan distance and p = 2 for euclidean distance).
ANN	hidden_layer_sizes	(50,), (100,), (50, 50), (100, 100)	Number of hidden layers and number of neurons in each one.
	activation	[relu, tanh]	Activation function.
	solver	[adam, sgd]	Optimizer.
	alpha	[1e−5–1e−1]	Strength of the L2 regularization term.
	learning_rate	[constant, adaptive]	Learning rate schedule for weight updates.
	batch_size	[2,4,8,16,32,64,128]	Size of minibatches.
	max_iter	[100,200,500,1000]	Number of epochs.

based on the average ranked performances (R_j) of the models on each dataset, and it aims to test the null hypothesis that all the models have equal performance [49]. The Friedman statistic is computed as

$$Q = \frac{12D}{K(K+1)} \sum_{j=1}^{K} \left(R_j - \frac{K+1}{2} \right)^2 \tag{5}$$

where D is the number of datasets, K is the number of algorithms, and $R_j = \frac{1}{D} \sum_{i=1}^{D} r_i^j$ is the average rank of the algorithm j [50].

If the p-value is below a chosen significance level (i.e., 0.05), then the null hypothesis is rejected, which means that there is enough evidence to conclude that there are differences in performance.

When this happens, the Nemenyi post-hoc statistical test is used, which proves that the performance of various algorithms is significantly different if their average ranks differ by at least the critical difference (CD) [49]:

$$CD = q_{\alpha,K} \sqrt{\frac{K(K+1)}{6D}} \tag{6}$$

where $q_{\alpha,K}$ represents a critical value table for the corresponding significance level (α) and the number of models (K), and D is the number of data-sets.

After the test, the results can be studied with the diagrams proposed by Demšar [49]. These diagrams show the mean ranked performances of the algorithms and the critical difference, in a way that the lower the ranking of the algorithm, the better it is.

3.4.4. Inverse model

In machine learning, an inverse model refers to a type of model that aims to propose optimal input variables based on observed output variables. Unlike traditional forward models that relate inputs to outputs, inverse models work in the reverse direction. They use observed output data to make prescriptions about the corresponding inputs that led to those outputs. Different inverse modelling techniques exist, such as optimization algorithms or Bayesian inference, which enable a deeper understanding of complex systems and facilitate decision-making processes.

The initial inverse modelling approach proposed in this paper involves using a trained predictive model to estimate input parameters based on observed output variables. As stated in previous sections, the predictive model is based on the comparison of five different Machine Learning algorithms. To establish the inverse model, the trained predictive model is utilized to predict the output depth based on input parameters. For a given target output depth, the inverse model proposes an input parameter combinations sampled from the original DoE dataset. For each input combination, the trained predictive model predicts the output depth, and the absolute difference between the predicted depth and the target depth is calculated. The inverse model then selects the input combination that results in the smallest absolute difference, thus identifying the best match to achieve the desired output depth.

This approach offers simplicity and efficiency, as it does not rely on explicit optimization techniques or complex mathematical procedures. It allows for a straightforward estimation of input parameters based on the observed output, providing insights into the relationships between the inputs and outputs of the system under consideration. The performance of this approach is strongly related to the robustness of the initially trained predictive model and the number of input combinations given to the inverse model.

Given the simplicity of the model explained above, a more sophisticated evolutionary strategy is also proposed. The intention is to better exploit the predictive model and give more accurate parameters. The proposed algorithm is a Genetic Algorithm [51,52] which is widely used in this type of tasks. In this case the following parameters are used:

- Population size: 8 individuals
- Number of parents for crossing: 4
- Probability of crossover: 0.9
- Mutation probability: 0.9

The implemented cost function minimizes the difference of the model prediction with the required depth. As a stopping criterion, 500 iterations of the algorithm are fixed.

These parameters have been established by trial and error by measuring the best performance from the initial random population.

Table 5
Hyper-parameter tuning results.

Model	Parameters	Steel lines	Steel dimples	TiN lines	TiN dimples	TiCN lines	TiCN dimples
DT	max_depth	64	38	98	8	98	38
	max_features	2	3	3	3	3	3
	min_samples_leaf	2	2	2	2	2	2
	min_samples_split	2	3	8	2	8	3
RF	n_estimators	259	159	733	389	190	161
	max_depth	20	53	51	19	76	66
	max_features	3	2	2	3	2	2
	min_samples_leaf	2	2	2	2	3	2
	min_samples_split	2	5	2	4	2	3
GP	alpha	3.94e−2	4.9e−3	1.09e−2	6.62e−3	3.5e−3	5.9e−3
SVM	gamma	2.34e−3	2.7e−1	9.79	27.07	1.72e−1	2.25e−2
	C	75.6	129.12	251.88	73.36	968.53	401.26
	epsilon	7.2e−2	1.2e−1	1.5e−1	5.3e−2	1.09e−1	9.5e−2
KNN	n_neighbours	2	2	2	2	2	2
	weights	uniform	uniform	uniform	uniform	uniform	uniform
	p	2	2	2	2	2	2
ANN	hidden_layer_sizes	(100,100)	(100,100)	(50,50)	(100,100)	(100,100)	(100,100)
	activation	relu	tanh	relu	relu	relu	relu
	solver	adam	adam	adam	sgd	adam	sgd
	alpha	1.39e−5	2.29e−4	2.08e−3	1.05e−2	3.9e−3	4.19e−3
	learning_rate	adaptive	constant	constant	adaptive	constant	adaptive
	batch_size	4	4	2	4	32	2
	max_iter	1000	1000	500	1000	1000	500

4. Results and discussion

As mentioned in previous sections, several ML models are compared in this study to find the best model for predicting the depth in various laser machined cavities. This section presents the results obtained during the training and prediction process of the ML models. Table 5 shows the results of the hyper-parameter tuning of each model.

Some significant differences can be observed in the hyper-parameter values of the same model across different data-sets (e.g., max_depth in DT model). The explanation lies in the fact that very different values of hyper-parameters resulted in very similar performance during the hyper-parameter tuning process. After tuning the hyper-parameters and training the model, in order to compare the performance of all the models across different data-sets, the R^2 score is computed on the test-set. The values of the test set data points are shown in Table 6, and the results of the models are presented in Table 7.

In this comparison, it is evident that all techniques achieve better performance in predicting dimples' depth than lines' depth, with R^2 values of 0.99 for steel, TiN and TiCN dimples, values around 0.89 for steel lines, 0.85 for TiN lines, and 0.79 for TiCN lines, respectively.

Random Forest and Decision Trees approaches are the ones that present the worst results in terms of R^2 score. On the other hand, Gaussian Process and SVM present good performance in dimples cases, but their results worsen in the prediction of lines. However ANN model outperforms the rest of the methods across every data-set, becoming the most suitable and best performing method for predicting lines' and dimples' depth.

To confirm this, visual representations of the results obtained for each prediction are provided in Fig. 6. Two graphics are shown for each data-set, displaying the differences between actual and predicted values in an explainable manner, which helps visualize the accuracy of the model's performance.

The bar-chart shows the absolute difference between the predicted depth and the actual one. Representing these differences as bars on the chart allows us to visualize how much the predictions deviate from the actual values. A smaller absolute difference, closer to zero, indicates more accurate predictions, suggesting that the model has a better fit to the data.

The scatter plot faces the real values against the predicted ones. The closer the points are to the regression line, the better the prediction of the depth is. The scatter plot also shows the 95% confidence interval of the predictions, represented by the shadowed area. This interval indicates the probability of the real regression line being within that area. In other words, as the area increases, the level of uncertainty in the predictions tends to rise.

Table 6
Laser parameters from the 16 test data points of steel line case.

Pulse repetition rate (kHz)	Pulse energy (μJ) theoric	SS (mm/s)	Mean depth
100	5.49	360	2.39
62.5	11.52	198	3.14
55.55	5.84	285	1.80
62.5	6.20	218	2.51
71.43	7.34	383	1.76
55.55	12.41	366	1.45
50	11.96	131	3.64
62.5	10.65	291	2.04
50	5.14	396	0.995
100	5.84	209	4.007
83.33	5.49	234	2.995
100	12.41	359	2.79
55.55	10.65	218	2.426
55.55	7.73	132	3.725
55.55	8.54	357	1.44
166.66	9.37	320	5.11

Table 7
R^2 score of depth prediction (best values for each case are highlighted in bold).

Material	RF	GP	SVM	DT	KNN	ANN
Steel lines	0.72	0.84	0.85	0.71	0.72	**0.89**
Steel dimples	0.80	**0.99**	**0.99**	0.78	0.90	**0.99**
TiN Lines	0.74	0.84	**0.85**	0.72	0.75	**0.85**
TiN dimples	0.82	**0.99**	**0.99**	0.80	0.91	**0.99**
TiCN lines	0.6	0.74	0.75	0.73	0.69	**0.79**
TiCN dimples	0.79	**0.99**	**0.99**	0.77	0.89	**0.99**

By analysing the plots for the steel line data-set, the results observed in Table 7 are confirmed. While the plots corresponding to Gaussian Process and Support Vector Machines show smaller absolute differences between the predicted and the real values than Decision Trees, Random Forest and KNN, Artificial Neural Network model shows the least differences and the least uncertainty in the predictions, outperforming the rest of the methods.

In order to study the robustness of the models against the existing variance of each instance, graphs shown in Fig. 7 are provided. Red dots represent the predicted values, while blue lines represent the minimum and maximum values of the experiments for each instance.

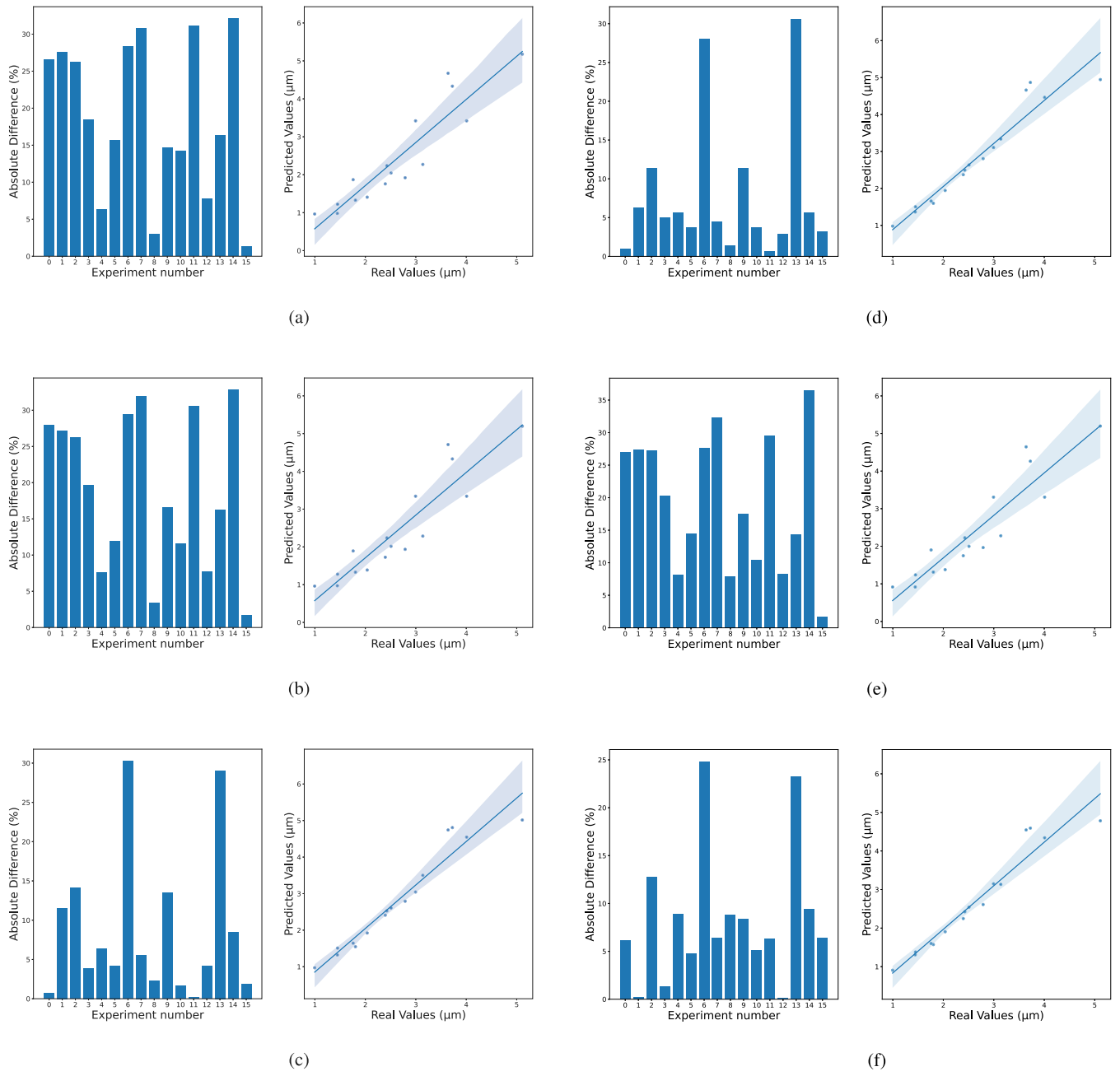


Fig. 6. Results of each model for the Steel Lines case: (a) RF, (b) DT, (c) GP, (d) SVM, (e) KNN (f) ANN. The shadowed area represents the 95% confidence interval of the predictions.

The prediction being inside the blue range indicates that the model has learned from the intrinsic variances of the experiments.

These graphs show that Gaussian Processes, Support Vector Machines and Artificial Neural Network models are more robust to noise or data fluctuations as they are able to capture the variance of the individual instances better than other tested models. This robustness is evident as the predictions of these models (represented by red dots) are generally closer to the blue lines representing the experimental variance range. Among them, ANN presents slightly better results than the other ones, suggesting that the ANN model is particularly effective in applications where data variability is significant.

To reinforce the results the Friedman statistical test was carried out with a significance level of 0.05, and it reported a p-value of 1.05e−4 (lower than the significance level). Accordingly, the null hypothesis can be rejected, assuming that the algorithms perform significantly differently. After this, the Nemenyi test is used to find the algorithm that performs better overall.

The results of this test are shown in Fig. 8. The diagram consists of a series of vertical lines, with each line representing a single model or algorithm being compared. The models are ranked based on their average performance on each data set, with the best-performing model on the left and the worst-performing one on the right.

The horizontal black lines in the diagram represent the critical difference, which is the minimum difference in the average performance that must exist between two models to be considered statistically significant. If two models are linked by a horizontal line, it means that their difference in performance is not statistically significant. On the other hand, if two models are not linked by a horizontal line, it means that their difference in performance is statistically significant, and one model performs better than the other. From this diagram, we can conclude the following:

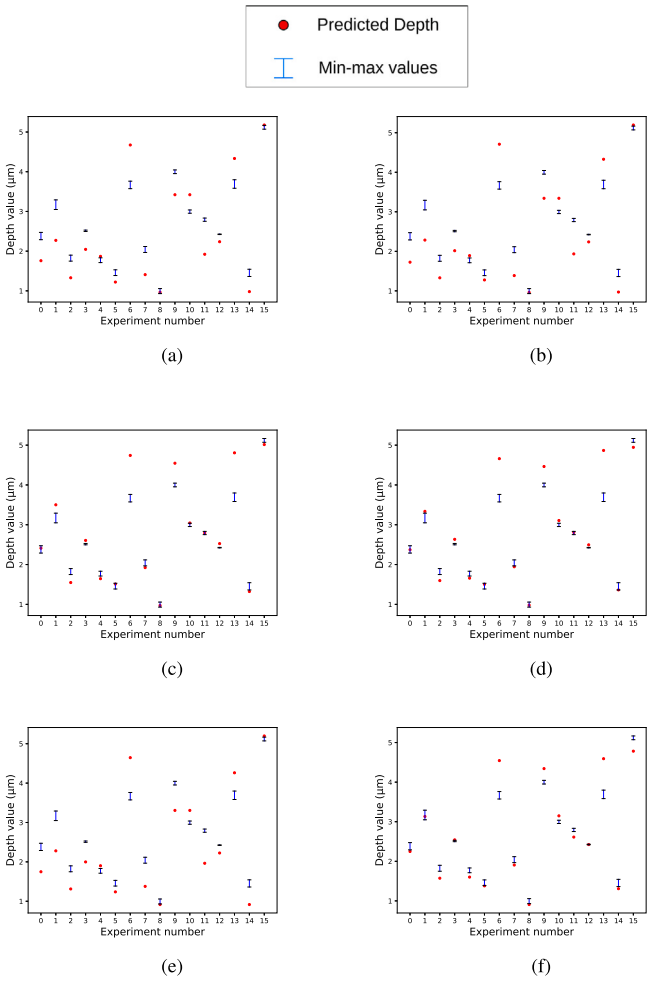


Fig. 7. Differences between predicted depth and min-max depth of each instance for steel lines models: (a) RF, (b) DT, (c) GP, (d) SVM, (e) KNN, (f) ANN.

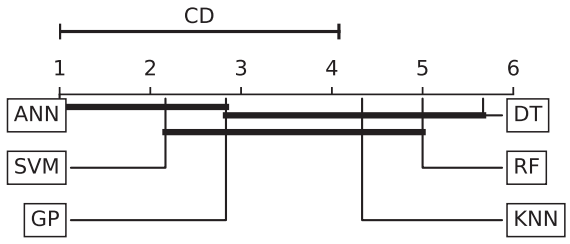


Fig. 8. Nemenyi test diagram.

- According to the statistical tests, ANN has the best average performance in the used data-sets, and its performance is statistically significant compared to RF, DT and KNN algorithms. This matches the results of the R^2 score shown in Table 7.
- Although ANN is the best performing algorithm overall, its difference with SVM and GP is not statistically significant.

According to the analysis of the results and the statistical tests, Artificial Neural Network is the best performing model in the prediction of the depth. Therefore, this model is used in the inverse model approaches presented in Section 3.4.4.

As mentioned in Section 3.4.4, two inverse modelling approaches for prescripting input parameters based on a given target depth are proposed in this study. The test data instances have been used to check the good performance of both strategies. The first approach is a simple model that selects the input parameters from the training dataset whose predicted depth is closest to the target depth.

Results of the first approach show that the biggest absolute difference between the desired and real depths is around 7%, and the mean among the 16 test data points is around 4%. These low differences confirm the capability of this method to propose parameters combinations that achieves depth values that are near to the desired one. However, as this method does not generate new input combinations but relies solely on existing data points, often results in errors when prescribing the input parameters.

To address this limitation, we developed a more advanced inverse model using a genetic algorithm. This algorithm optimizes the input parameters to achieve the target depth with higher accuracy. The genetic algorithm iteratively evolves a population of input parameter sets, utilizing operations such as selection, crossover, and mutation to effectively explore the parameter space.

The analysis conducted for the inverse model with the evolutionary strategy is given below. For each instance, the genetic algorithm is run 50 times and the run that has a mean absolute error is recorded. This error is calculated with the normalized differences between the real parameters and the estimates of the evolutionary inverse model. This way of validating the model is due to the fact that different combinations of laser parameters offer the same depth, due to the characteristics of the test set. Therefore, for the genetic algorithm there are a multitude of local minima, i.e., possible parameter configurations for a given depth. However, this validation strategy serves to demonstrate that in addition to finding a very similar depth, for that depth the algorithm manages to find the parameters of the test analysed.

The Table 8 shows the results of this strategy for the inverse problem where the laser parameters are prescribed on a depth basis. The error committed is always less than 5.4%, with an average error for all parameters of 4.8%. It should be added that the solutions obtained always achieve very similar depths to the required one (differences close to 0). In the comparison of both inverse model approaches, the evolutionary approach demonstrates a significant improvement over the first approach being able to reduce the error of every prescribed parameter. This substantial decrease in error shows the effectiveness of the evolutionary approach in finding more accurate input parameters to achieve a given depth.

Table 8
Results of evolutionary inverse model.

Parameter	Mean absolute error
Pulse repetition rate	5.3%
Pulse energy	3.6%
Theoric SS	5.4%
Depth	1.93e−5%

5. Conclusions and future work

In this work, a Machine Learning methodology for predicting the depth of the cavity and for proposing the optimal parameters of a femtosecond laser for a desired depth is proposed. Specifically, six different Machine Learning algorithms are compared for predicting the depth, and the best one is used to create two inverse model approaches that proposes the best input parameter combination of the laser.

Conclusions reached through the comparison of the models and the analysis of the results are presented below.

1. Artificial Neural Network model achieves the best results in terms of R^2 score, and this is supported by different statistical tests. Therefore, we can conclude that, for this data-sets, ANN model is the best for predicting the depth of lines and dimples.
2. All of the approaches have better performance in predicting dimples' depth than lines' depth. However, further analysis is needed to draw a conclusion about this insight.

3. The results obtained show that Machine Learning is an effective approach for predicting femtosecond laser textures.
4. The first inverse modelling approach successfully proposes input parameters that result in depth values close to the given target depth, with a mean absolute error margin of around 4%. However, the prescribed input parameters differ significantly from the actual tested input parameters. This discrepancy arises because different sets of input parameters can produce similar depth values, leading to substantial variations in the proposed parameters.
5. The evolutionary based inverse model provides a very reliable laser parameters estimation, considerably improving the first inverse model approach results, by reducing the error between the given depth and the actual depth to values near 0%. In addition, is also able to propose similar input parameter values to the real ones, with an average error of 4.8%. The practical use of this model is to provide a depth where the output will give a feasible combination of parameters.
6. The study highlighted the need for a tool that allows non-Machine Learning expert users to utilize ML models for predicting process parameters. Addressing this necessity, all the code will be published on https://github.com/luismoles/texturized_laser.

Finally some lines to be followed in future work are identified, and are enumerated below.

1. An experimental validation of the inverse model approaches proposed in this paper will be carried out.
2. Based on the results, all of the approaches demonstrate better performance in predicting dimples' depth compared to lines' depth. The underlying reasons for this discrepancy and the development of methods to improve the depth predictions for lines will be investigated.
3. As stated in the conclusions, the evolutionary based inverse model approach obtains successful results in prescripting input parameter combinations. However, the application could be improved by setting an input parameter and making a more targeted search. The latter will be investigated in future work.
4. An online framework will be provided, allowing non-expert ML users to reproduce the proposed pipeline.

CRediT authorship contribution statement

Luis Moles: Writing – review & editing, Writing – original draft, Validation, Software, Formal analysis. **Iñigo Llavori:** Writing – review & editing, Supervision, Investigation, Funding acquisition. **Andrea Aginagalde:** Writing – review & editing, Supervision, Investigation, Data curation. **Goretti Echegaray:** Writing – review & editing, Supervision, Investigation. **David Bruneel:** Writing – review & editing, Supervision, Resources. **Fernando Boto:** Writing – review & editing, Supervision, Methodology, Investigation. **Alaitz Zabala:** Writing – review & editing, Supervision, Methodology, Investigation, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The data and code used in this research, will be made available after publication.

Declaration of Generative AI and AI-assisted technologies in the writing process

Statement: During the preparation of this work the authors used GPT4.0 in order to improve readability and language. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

Acknowledgements

The authors gratefully acknowledge the financial support given by the Basque Government (Eusko Jaurlaritza) under "Programa de apoyo a la investigación colaborativa en áreas estratégicas" (Project BISUM: Ref. KK-2021/00089) programs. The authors would also like to acknowledgement the technical support provided by Unai Lizarribar and Dr Alain Andrés Fernández.

References

[1] Holmberg K, Erdemir A. The impact of tribology on energy use and CO2 emission globally and in combustion engine and electric cars. Tribol Int 2019;135:389–96.
[2] Mao B, Siddaiah A, Liao Y, Menezes PL. Laser surface texturing and related techniques for enhancing tribological performance of engineering materials: A review. J Manuf Process 2020;53:153–73.
[3] Grützmacher PG, Profito FJ, Rosenkranz A. Multi-scale surface texturing in tribology—Current knowledge and future perspectives. Lubricants 2019;7(11):95.
[4] Gropper D, Wang L, Harvey TJ. Hydrodynamic lubrication of textured surfaces: A review of modeling techniques and key findings. Tribol Int 2016;94:509–29.
[5] Ibatan T, Uddin M, Chowdhury M. Recent development on surface texturing in enhancing tribological performance of bearing sliders. Surf Coat Technol 2015;272:102–20.
[6] Bonse J, Kirner SV, Griepentrog M, Spaltmann D, Krüger J. Femtosecond laser texturing of surfaces for tribological applications. Materials 2018;11(5):801.
[7] Orazi L, Cuccolini G, Fortunato A, Tani G. An automated procedure for material removal rate prediction in laser surface micromanufacturing. Int J Adv Manuf Technol 2010;46:163–71.
[8] Benton M, Hossan MR, Konari PR, Gamagedara S. Effect of process parameters and material properties on laser micromachining of microchannels. Micromachines 2019;10(2):123.
[9] Campanelli S, Lavecchia F, Contuzzi N, Percoco G. Analysis of shape geometry and roughness of Ti6Al4V parts fabricated by nanosecond laser ablation. Micromachines 2018;9(7):324. http://dx.doi.org/10.3390/mi9070324, URL http://www.mdpi.com/2072-666X/9/7/324.
[10] Pou-Álvarez P, Riveiro A, Nóvoa X, Fernández-Arias M, Del Val J, Comesaña R, Boutinguiza M, Lusquiños F, Pou J. Nanosecond, picosecond and femtosecond laser surface treatment of magnesium alloy: Role of pulse length. Surf Coat Technol 2021;427:127802.
[11] Ezhilmaran V, Vasa N, Vijayaraghavan L. Investigation on generation of laser assisted dimples on piston ring surface and influence of dimple parameters on friction. Surf Coat Technol 2018;335:314–26.
[12] Nolte S, Momma C, Jacobs H, Tünnermann A, Chichkov BN, Wellegehausen B, Welling H. Ablation of metals by ultrashort laser pulses. J Opt Soc Am B 1997;14(10):2716–22.
[13] Xian J, Wang X, Fu X, Zhang Z, Liu L, Kang M. A simple model to predict machined depth and surface profile for picosecond laser surface texturing. Appl Sci 2018;8(11):2111.
[14] Zhang Z, Liu S, Zhang Y, Wang C, Zhang S, Yang Z, Xu W. Optimization of low-power femtosecond laser trepan drilling by machine learning and a high-throughput multi-objective genetic algorithm. Opt Laser Technol 2022;148:107688.
[15] Wang C, Zhang Z, Jing X, Yang Z, Xu W. Optimization of multistage femtosecond laser drilling process using machine learning coupled with molecular dynamics. Opt Laser Technol 2022;156:108442.
[16] Desai CK, Shaikh A. Prediction of depth of cut for single-pass laser micromilling process using semi-analytical, ANN and GP approaches. Int J Adv Manuf Technol 2012;60(9–12):865–82. http://dx.doi.org/10.1007/s00170-011-3677-8, URL http://link.springer.com/10.1007/s00170-011-3677-8.
[17] Wang Z, Ye R, Xiang J. The performance of textured surface in friction reducing: A review. Tribol Int 2023;177:108010.
[18] Mahesh B. Machine learning algorithms-a review. Int J Sci Res (IJSR). [Internet] 2020;9:381–6.
[19] El Naqa I, Murphy MJ. What is machine learning? In: Machine learning in radiation oncology. Springer; 2015, p. 3–11.
[20] Khanzode KCA, Sarode RD. Advantages and disadvantages of artificial intelligence and machine learning: A literature review. Int J Libr Inf Sci (IJLIS) 2020;9(1):3.

[21] Loh W-Y. Classification and regression trees. Wiley Interdiscip Rev: Data Min Knowl Discov 2011;1(1):14–23.
[22] Breiman L. Random forests. Mach Learn 2001;45(1):5–32.
[23] Breiman L. Bagging predictors. Mach Learn 1996;24(2):123–40.
[24] Schulz E, Speekenbrink M, Krause A. A tutorial on Gaussian process regression: Modelling, exploring, and exploiting functions. J Math Psych 2018;85:1–16.
[25] Smola AJ, Schölkopf B. A tutorial on support vector regression. Stat Comput 2004;14:199–222.
[26] Song Y, Liang J, Lu J, Zhao X. An efficient instance selection algorithm for k nearest neighbor regression. Neurocomputing 2017;251:26–34.
[27] Zhang S, Li X, Zong M, Zhu X, Cheng D. Learning k for knn classification. ACM Trans Intell Syst Technol 2017;8(3):1–19.
[28] Rojas R. Neural networks: a systematic introduction. Springer Science & Business Media; 2013.
[29] Rumelhart DE, Hinton GE, Williams RJ. Learning representations by back-propagating errors. Nature 1986;323(6088):533–6.
[30] Sivanandam S, Deepa S, Sivanandam S, Deepa S. Genetic algorithms. Springer; 2008.
[31] Hazrati J, Stein P, Kramer P, Van Den Boogaard A. Tool texturing for deep drawing applications. In: IOP conference series: materials science and engineering. Vol. 418, IOP Publishing; 2018, 012095.
[32] Serro A, Completo C, Colaço R, Dos Santos F, Da Silva CL, Cabral J, Araújo H, Pires E, Saramago B. A comparative study of titanium nitrides, TiN, TiNbN and TiCN, as coatings for biomedical applications. Surf Coat Technol 2009;203(24):3701–7.
[33] Chyr A, Qiu M, Speltz JW, Jacobsen RL, Sanders AP, Raeymaekers B. A patterned microtexture to reduce friction and increase longevity of prosthetic hip joints. Wear 2014;315(1–2):51–7.
[34] Borjali A, Monson K, Raeymaekers B. Friction between a polyethylene pin and a microtextured CoCrMo disc, and its correlation to polyethylene wear, as a function of sliding velocity and contact pressure, in the context of metal-on-polyethylene prosthetic hip implants. Tribol Int 2018;127:568–74.
[35] Drescher P, Oldorf P, Dreier T, Peters R, Seitz H. Modification of joint prosthesis surfaces by ultrashort pulse laser treatment for improved joint lubrication. Curr Direct Biomed Eng 2019;5(1):57–60.
[36] Xu Y, Zheng Q, Abuflaha R, Olson D, Furlong O, You T, Zhang Q, Hu X, Tysoe WT. Influence of dimple shape on tribofilm formation and tribological properties of textured surfaces under full and starved lubrication. Tribol Int 2019;136:267–75.
[37] Conradi M, Kocijan A, Klobčar D, Podgornik B. Tribological response of laser-textured Ti6Al4V alloy under dry conditions and lubricated with Hank's solution. Tribol Int 2021;160:107049.
[38] Nanbu T, Ren N, Yasuda Y, Zhu D, Wang QJ. Micro-textures in concentrated conformal-contact lubrication: effects of texture bottom shape and surface relative motion. Tribol Lett 2008;29:241–52.
[39] Bharatish A, Soundarapandian S. Influence of femtosecond laser parameters and environment on surface texture characteristics of metals and non-metals–state of the art. Lasers Manuf Mater Process 2018;5:143–67.
[40] Lopez J, Faucon M, Devillard R, Zaouter Y, Honninger C, Mottay E, Kling R. Parameters of influence in surface ablation and texturing of metals using high-power ultrafast laser. J Laser Micro Nanoeng 2015;10(1):1.
[41] Liu S, Zhang Z, Yang Z, Wang C. Femtosecond laser-induced evolution of surface micro-structure in depth direction of nickel-based alloy. Appl Sci 2022;12(17):8464.
[42] Primus T, Novák M, Zeman P, Holešovskỳ F. Femtosecond laser processing of advanced technical materials. 2023.
[43] Moskal D, Martan J, Honner M. Scanning strategies in laser surface texturing: A review. Micromachines 2023;14(6):1241.
[44] Hasan MS, Kordijazi A, Rohatgi PK, Nosonovsky M. Triboinformatic modeling of dry friction and wear of aluminum base alloys using machine learning algorithms. Tribol Int 2021;161:107065.
[45] Hasan MS, Kordijazi A, Rohatgi PK, Nosonovsky M. Triboinformatics approach for friction and wear prediction of al-graphite composites using machine learning methods. J Tribol 2022;144(1):011701.
[46] Timur M, Aydin F. Anticipating the friction coefficient of friction materials used in automobiles by means of machine learning without using a test instrument. Turk J Electr Eng Comput Sci 2013;21(5):1440–54.
[47] Akiba T, Sano S, Yanase T, Ohta T, Koyama M. Optuna: A next-generation hyperparameter optimization framework. In: Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining. 2019, p. 2623–31.
[48] Refaeilzadeh P, Tang L, Liu H. Cross-validation.. Ency Database Syst 2009;5:532–8.
[49] Demšar J. Statistical comparisons of classifiers over multiple data sets. J Mach Learn Res 2006;7:1–30.
[50] Gutiérrez-Gómez L, Petry F, Khadraoui D. A comparison framework of machine learning algorithms for mixed-type variables datasets: a case study on tire-performances prediction. IEEE Access 2020;8:214902–14.
[51] Deb K. Introduction to genetic algorithms for engineering optimization. 2004, URL https://api.semanticscholar.org/CorpusID:59911104.
[52] Fraser AS. Simulation of genetic systems by automatic digital computers II. effects of linkage on rates of advance under selection. Aust J Biol Sci 1957;10:492–500, URL https://api.semanticscholar.org/CorpusID:54799366.