

## Article

# Application of Machine Learning in Predicting Quality Parameters in Metal Material Extrusion (MEX/M)

Karim Asami , Maxim Kuehne , Tim Röver  and Claus Emmelmann

Institute of Laser and System Technologies, Hamburg University of Technology, Harburger Schloßstraße 28, 21079 Hamburg, Germany; maxim.kuehne@tuhh.de (M.K.); tim.roever@tuhh.de (T.R.); c.emmelmann@tuhh.de (C.E.)

\* Correspondence: karim.asami@tuhh.de

**Abstract:** Additive manufacturing processes such as the material extrusion of metals (MEX/M) enable the production of complex and functional parts that are not feasible to create through traditional manufacturing methods. However, achieving high-quality MEX/M parts requires significant experimental and financial investments for suitable parameter development. In response, this study explores the application of machine learning (ML) to predict the surface roughness and density in MEX/M components. The various models are trained with experimental data using input parameters such as layer thickness, print velocity, infill, overhang angle, and sinter profile enabling precise predictions of surface roughness and density. The various ML models demonstrate an accuracy of up to 97% after training. In conclusion, this research showcases the potential of ML in enhancing the efficiency in control over component quality during the design phase, addressing challenges in metallic additive manufacturing, and facilitating exact control and optimization of the MEX/M process, especially for complex geometrical structures.

**Keywords:** additive manufacturing (AM); material extrusion of metals (MEX/M); machine learning (ML); process development; AISI stainless steel 1.4404/316L; design for additive manufacturing (DfAM)



Academic Editors: T. DebRoy,  
Thomas Niendorf, Dayalan  
Gunasegaram, Paulo J. Bártolo and  
Yang Du

Received: 10 March 2025

Revised: 13 April 2025

Accepted: 27 April 2025

Published: 30 April 2025

**Citation:** Asami, K.; Kuehne, M.; Röver, T.; Emmelmann, C. Application of Machine Learning in Predicting Quality Parameters in Metal Material Extrusion (MEX/M). *Metals* **2025**, *15*, 505. <https://doi.org/10.3390/met15050505>

**Copyright:** © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Additive manufacturing (AM) [1] technologies allow, due to layerwise production, the manufacture of complex and intricate geometrical components with reduced material waste and energy consumption. Until now, the laser-beam-based powder bed fusion of metals (PBF-LB/M) has been the most used AM technology in several application industries such as the aerospace, automotive, and medical environments. Therefore, a large amount of data for different PBF-LB/M machines exists, and also approaches to machine learning (ML) prediction promise powerful support in predicting suitable and successful process parameter combinations for different materials [2–6]. The objective of the study in [4] is to utilize ML models to predict the density of AlSi10Mg parts based on various process parameters. To achieve this, 54 experimental data points, as referenced in [7], were used to train and test multilayer perceptron (MLP) models. Among the models evaluated, the MLP was found to be the most accurate and robust prediction model [2]. In addition, two MLP models were trained and tested using density results from simulation data developed in study [4]. However, this did not lead to an improvement in prediction accuracy. Without the inclusion of simulation data, the root mean square error (RMSE) of the MLP model was 0.91%, demonstrating the potential of the MLP algorithm even with a limited amount of experimental data. When incorporating the simulation data, the RMSE increased by 21%,

resulting in less accurate predictions. This increase is likely attributable to computational errors associated with the solver used in the simulation. Another study [5] evaluated different ML models for the density prediction of a stainless steel 316L PBF-LB/M specimen with a difference of less than 4% between the experimental results and the ML-based prediction. Different ML models, such as support vector regression (SVR), random forest (RFR), and ridge regression, have been developed to estimate surface roughness in fused deposition modeling (FDM) with high accuracy [8]. Machine learning is also utilized in the design of AM parts. In [9], maximum-stress predictions were made using different regression models, allowing for the efficient use of lattice structures. In this paper, gradient boost regression and random forest regression were the models with the smallest RMSEs of 67.31 MPa and 77.07 MPa for compression and bending specimens, respectively, with  $R^2$  scores of 0.91 and 0.88. These results indicate relatively high performance in terms of predicting mechanical properties and reducing computational simulation time. Additionally, machine learning models are being developed to predict the compressive strength of additively manufactured components, such as PEEK spinal fusion cages, and to improve the geometric accuracy of AM parts through online shape deviation inspection and compensation [10,11]. ML in AM is further used for the prediction of defects in terms of thickness and length. With Gaussian process regression, MLP, RFR, and support vector machine (SVM) models, it was possible to demonstrate significant advantages in processing time and performance as a non-destructive test [12]. These examples show the potential integration of ML in different fields of AM technologies and design processes. It can be used for the prediction of process parameters to achieve successful and desired production, monitor the manufacturing process, and reduce simulation effort for achieving the desired component characteristics. Nevertheless, ML in AM faces several limitations such as data scarcity and quality, variability in product quality, computational challenges, and material and process limitations. The novelty of AM results in a lack of extensive training data, which is crucial for developing robust ML models. Additionally, the quality and standardization of available data are often inadequate, affecting the reliability of predictions [13,14]. AM processes are highly complex and much less established than other manufacturing processes such as machining or casting. Consequently, AM processes are often still lacking in product quality, which poses a significant challenge for ML applications. Such variability complicates the development of predictive models that can generalize well across different conditions [15,16]. The complexity of AM processes requires significant computational resources for ML model training and optimization. This requirement can be a barrier, especially for small-scale operations [13]. The limited library of materials and the presence of processing defects further restrict the applicability of ML in AM. These challenges can hinder the accurate prediction of material performance and process outcomes [17]. Addressing these limitations requires advancements in data collection, standardization, and computational techniques. There are several successful case studies that demonstrate overcoming the limitations of machine learning in additive manufacturing. In [18], machine learning algorithms, such as random forests and support vector machines, are employed to optimize process parameters like laser power and scanning speed. By monitoring the process and predicting the optimal combination of parameters, the variability in printed part quality can be reduced, enhancing reliability in applications such as the aerospace and medical fields. Integrating machine learning with digital twin technology allows for real-time monitoring and control of the AM process. A case study demonstrated the integration's effectiveness in defect detection, achieving high precision and reliability in manufacturing processes [19]. In [20], a hybrid machine learning algorithm was developed to recommend design features during the conceptual phase of AM. This approach was validated through a case study involving the design of R/C car components, proving useful for inexperienced designers by providing

feasible design solutions. Furthermore, the synthetic parameters of  $Ni_3TeO_6$  multiferroic materials were experimentally trained and tested with the RFR model using a small dataset of nine data points. The data showed no correlation, and yet a prediction accuracy of 87% was achieved [21]. A final approach of RFR, gradient boosting, and XGBoost ML models was used for the prediction of the corrosion rate of zinc-based alloys with copper, lithium, magnesium, and silver [22]. Here, XGBoost showed the highest prediction accuracy and the highest  $R^2$  value.

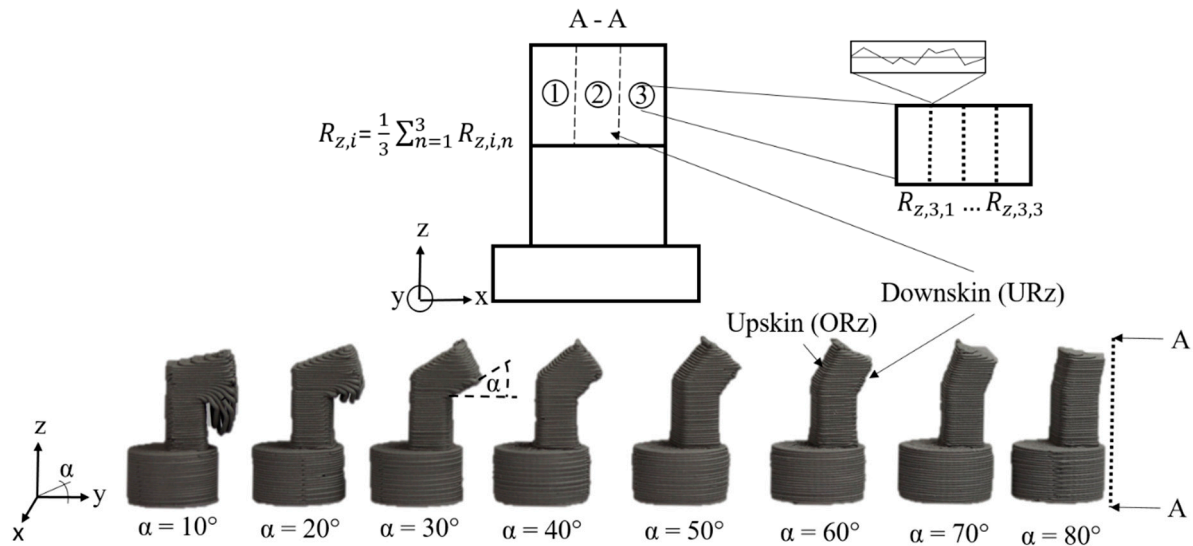
The material extrusion of metals (MEX/M, ISO/ASTM 52900) has gained attraction in recent years due to its status as a low-cost AM alternative. The MEX/M process consists of the steps shaping, debinding, and sintering, which are very similar to the metal injection molding process (MIM). In the shaping process, a commercial 3D printer is used for the AM part instead of a mold. The initial intention of this process chain—where the feedstock for the shaping step can be, for example, filament or pellets depending on the extrusion mechanism—was to combine the knowledge of common 3D printing with the MIM process. Nonetheless, the MIM standard of component characteristics is not achievable with MEX/M technology. According to [23], more experimental parameter studies and sintering routines must be developed to achieve suitable material characteristics. In addition, the numerous influencing factors of the multistep process chain make it very difficult and require substantial effort for experimental evaluation. To overcome this hurdle, this paper evaluates an approach using machine learning models with a small batch of experimental data to predict further parameter settings for the MEX/M process. In this way, it is possible to identify suitable process parameters for component design without extensive experimental effort. Up to now, ML has been used in AM technologies for several reasons, either on the experimental side or on the simulation and design side. Furthermore, ML has been applied for quality assurance and predicting process parameter settings. So far, no predictions for the MEX/M process have been made, only for single-stage AM processes, such as the laser-beam-based powder bed fusion (PBF-LB/M) process. Surface roughness and density are key values for meeting the MIM standard of the combined technology and have been set as the target values to be predicted. The choice of the appropriate ML model is challenging, as several ML models are capable of predicting specific parameters, as discussed. Therefore, different models have been evaluated and compared, with a focus on models that can be trained with a small amount of data while achieving sufficiently good prediction.

## 2. Experimental Approach and ML Methodology

The aim of this study is to develop a predictive model that is able to estimate the surface roughness as well as the sintered density depending on the used process parameters. This provides insights into quality control and process optimization for MEX/M and allows for faster parameter development.

### 2.1. Specimen Manufacture and Measurements

For the data preparation, different overhang specimens (see Figure 1) have been manufactured on a Renkforce RF2000 (Conrad Electronic SE, Hirschau, Germany) 3D printer with a 0.4 mm nozzle using AISI 316L filament (PT&A GmbH, Dresden, Germany) with a 2.85 mm diameter. The surface roughness of the specimens, which are called green parts after the shaping process, have been measured on the downskin (URz) and upskin (ORz) areas of the overhang parts using a Keyence VHX-S600E (Osaka, Japan). Three areas of each side have been measured three times, and the average of the values has been taken as the average surface roughness of the downskin and upskin areas.

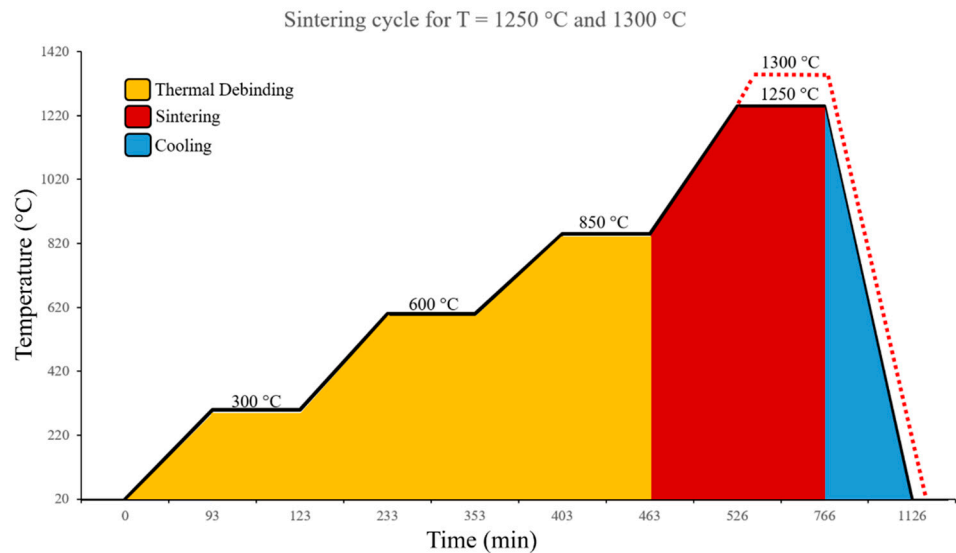


**Figure 1.** Overhang specimen (green parts) and roughness measuring procedure. Adapted from Ref. [24].

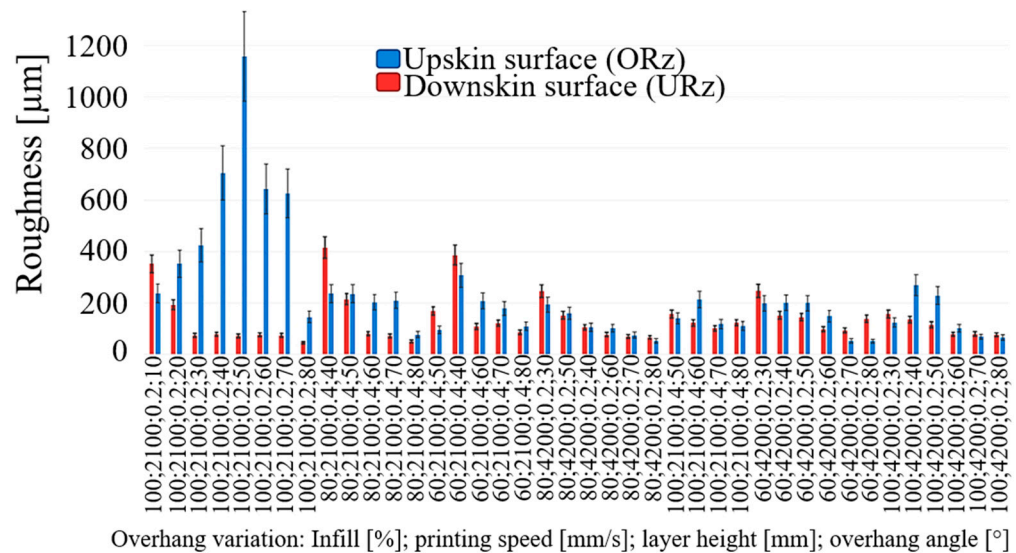
The starting powder composition is shown in [25]. The exact binder composition and particle size distribution of the metal powder are not provided by the manufacturer. The solvent debinding step is performed in 500 mL of acetone for 12 h, heated at 38 °C, as recommended by the filament provider, to achieve a 5% mass loss and open the pores for the thermal debinding step. An SEM image of the debinded 316L filament is provided in [26]. The particle size is in the range of 2 µm to 15 µm. An ExSO90 sinter oven (Aim3D GmbH, Rostock, Germany) was used for the thermal debinding and sintering step of the specimen manufacturing. The thermal debinding and sintering cycles are performed in a combined single step according to the recommendation of the feedstock provider, with 99.9% argon used as the atmosphere for both process steps at a flow rate of 1 L/min. It was found that this setup achieved a sintered density of 97.4% for the 316L specimen, which is higher than the metal injection molding (MIM) standard of 95% density [26]. Figure 2 illustrates the thermal debinding and sintering cycle for the specimens. Two different maximum temperatures were chosen for the printed process parameters. The holding temperatures of 300 °C, 600 °C, and 850 °C were selected to burn out the binder from the samples. The heating rate between each holding temperature is 5 K/min.

The measurements of the different parameter combinations are depicted on the x-axis in Figure 3. On the y-axis, the different roughness measurements of the upskin and downskin surfaces are illustrated. The identification of individual samples was based on the set process parameters. The first number represents the infill percentage [%], the second number denotes the printing speed in millimeters per second [mm/s], the third number corresponds to the layer height in millimeters [mm], and the last number indicates the overhang angle in degrees [°]. For example, the combination 100;2100;0.2;10 signifies that the sample was printed with an infill of 100%, a printing speed of 2100 mm/s, and a layer height of 0.2 mm. The overhang angle of the sample is 10°. As expected, the mean surface roughness for the upskin and downskin areas is higher for lower overhang angles, independent of the printing speed and the percentage infill. A layer height of 0.4 mm causes higher surface roughness for the downskin and upskin areas. The roughness measurements are in the typical range for roughness in the MEX/M process [25].

Furthermore, combinations of additional overhang angles have been printed, debinded, and sintered using the illustrated cycles (Figure 2). Table 1 shows the additional variations of the specimen parameters. The process parameters that were not varied are kept constant according to [25].



**Figure 2.** Thermal debinding and sintering cycle for maximum sinter temperature of 1250 °C and 1300 °C. Adapted from Ref. [26].



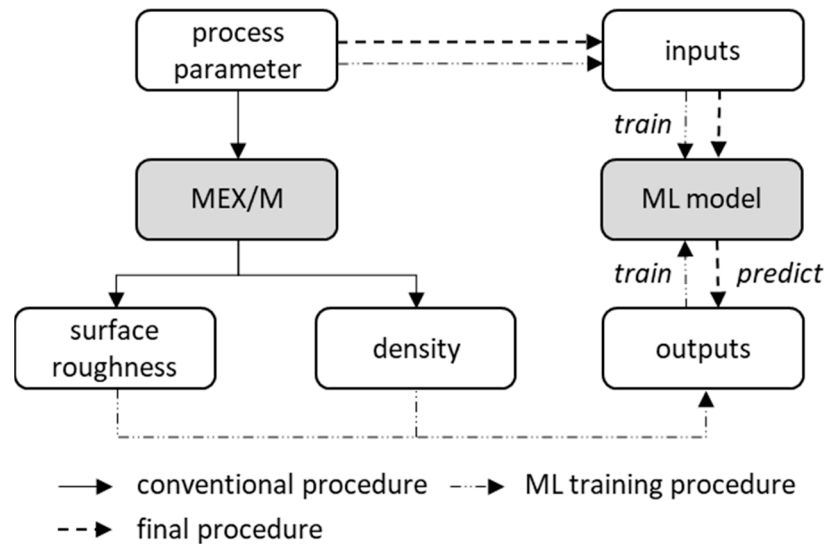
**Figure 3.** Average surface roughness of the downskin and upskin areas for each process parameter combination of the green parts.

**Table 1.** Parameter variation of the experimental data.

Variation	Infill [%]	Printing Speed [mm/s]	Layer Height [mm]	Overhang Angle [°]	Sintering Temperature
Green part: 40 combinations	60, 80, 100	2100, 4200	0.2, 0.4	10, 20, ..., 80	-
Sinter part: 48 combinations	50, 100	2100, 4200	0.2, 0.4	10, 20, ..., 80	1250, 1300

### 2.2. ML Development and Evaluation

The applied methodology and relations between process and parameters are shown in Figure 4. The left side shows the current procedure, where process parameters are selected, a MEX/M process is performed, and the target parameters are investigated. The right side shows the target methodology, where the process parameters are directly fed into the ML model to predict the resulting surface roughness and density. This procedure enables an initial assessment of the part quality without the part actually having to be manufactured.



**Figure 4.** Methodological scheme of the experimental approach and the ML prediction.

The inputs and outputs of the ML model are derived from the process. The process parameters of the MEX/M process are used as inputs and the part properties as outputs. The considered parameters are summarized in Table 2.

**Table 2.** Input and output parameter for the investigated ML models.

Parameter	Symbol	Description	Unit	Model
Inputs				
Infill	fill	Percentage of the material fill level	%	URz, ORz, density
Print speed	v	Speed of printing head during the process	mm/min	
Layer thickness	l	Height of the individual print layers	mm	
Overhang angle	alpha	Overhang angle of unsupported areas of the printed part	°	
Sinter temperature	st	temperature used when sintering the material	°C	
Outputs				
Downskin surface roughness	URz	Measured average bottom inclined wall roughness of the component surface	µm	URz
Upskin surface roughness	ORz	Measured average top inclined wall roughness of the component surface	µm	ORz
Density	rho	Part density before sintering	%	density

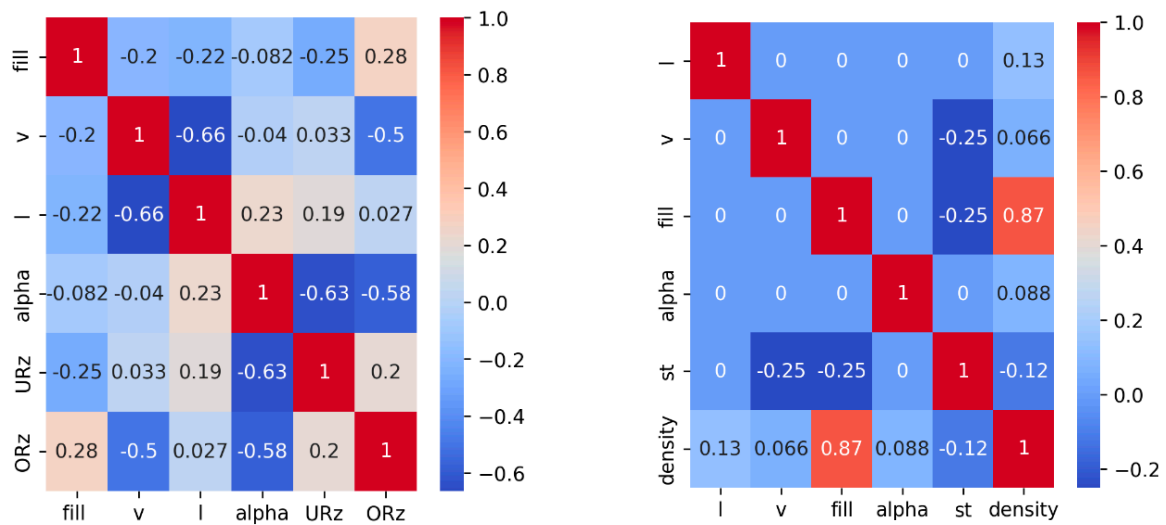
For data processing and training of the models, Python (v3.8.19) was used with the scikit-learn library (v1.3.0) [27]. These environments were selected to ensure compatibility with the latest machine learning methods as well as robust and efficient processing of the data. The developed suitable ML models and the experimental data are provided in [28].

### 3. ML Model Setup and Training

The dataset generated for the investigation of surface roughness consists of 40 data points, while the dataset for density comprises 48 data points. The preprocessing of the data consists of four steps: cleaning the data by removing outliers and making sure no values are missing, investigating correlations between features and labels and removing

strongly correlated features if necessary, splitting the data into training and test sets, and finally scaling the input data to ensure similar magnitudes of feature values.

When cleaning the data, no missing values and only slight outliers were detected. As the dataset is already very small, it was decided not to remove any further data points as long as the outliers are not attributable to measurement errors. Subsequently, a correlation analysis was performed, including the calculation of the correlation matrix and the Spearman coefficients [29,30]. The plots in Figure 5 show the correlation matrices for the parameters. The calculated coefficients are summarized in Table 3. The results show that URz correlates significantly negatively with angle ( $-0.63$ ), while ORz also shows a strong negative correlation with print speed ( $-0.5$ ) and angle ( $-0.58$ ). Density shows a strong positive correlation with infill ( $0.87$ ).



**Figure 5.** Correlation matrices for the surface roughness (left) and density (right).

**Table 3.** Spearman coefficients for the downskin surface roughness, upskin surface roughness and density.

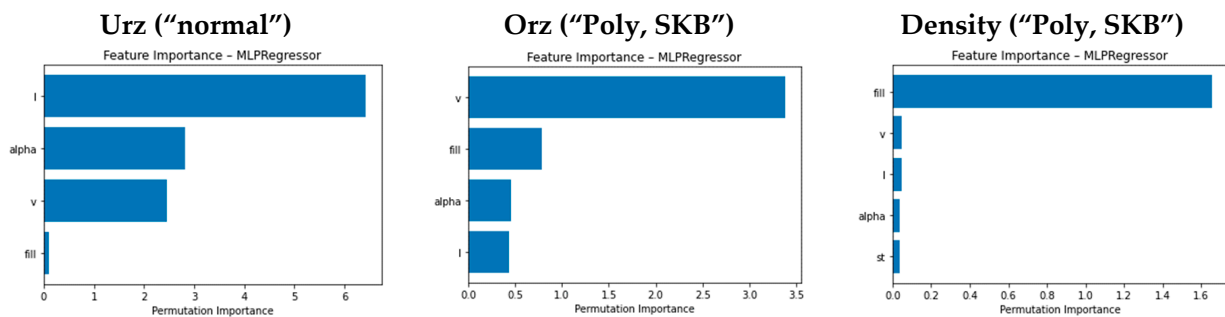
	URz	ORz	Density
fill	$-0.25$	$0.28$	$0.87$
v	$0.03$	$-0.5$	$0.07$
l	$0.19$	$0.03$	$0.13$
alpha	$-0.63$	$-0.58$	$0.09$
st	-	-	$-0.12$

For processing in ML algorithms, the data is split into a training set and a test dataset in a ratio of 75% to 25%. A stratified shuffle is applied based on the respective output parameter in order to take the distribution of the target variables into account. This ensures that, despite the small amount of data, results over the entire interval of the target parameter are always included in the training and test datasets. In addition, polynomial features are calculated up to the second degree in order to capture interaction effects [31]. When training the models, it is investigated whether the inclusion of polynomial features offers added value. A feature selection is carried out using SelectKBest. To standardize the feature values, either MinMax or Standard scaling is used to normalize the number spaces to a comparable range, which is important for the accuracy of some ML algorithms [27,32].

The Spearman correlation coefficient is used in the first step to identify nonlinear monotonic correlations between the features and the target variable. Furthermore, SelectKBest is used in the ML pipeline to eliminate irrelevant or redundant features and shorten

the computing time. In this method, the features are ranked based on their correlation to the target variable based on a score function  $f_{\text{regression}}$ , and the best  $K$  features are selected [27]. With GridSearchCV, different values for  $K$  are tested.

Permutation Importance is used to quantify the influence of individual features on the model by randomly permuting their values and measuring the change in model performance (see Figure 6). This made it possible to identify  $l$  as the most important feature for URz,  $v$  as the most important feature for ORz, and  $fill$  as the most important feature for density. For ORz and density, there is also a high Spearman correlation (see Figure 5) between the identified features and the target variable. The features are therefore not only statistically significant, but also meaningfully linked to the target variable in terms of content. The relationship between  $l$  value and URz is not directly apparent from the Spearman coefficient, which indicates a more complex correlation between the parameters. According to permutation importance, the feature  $alpha$ , which is more highly correlated with URz, is the second most important feature for training and therefore just as relevant for model training.

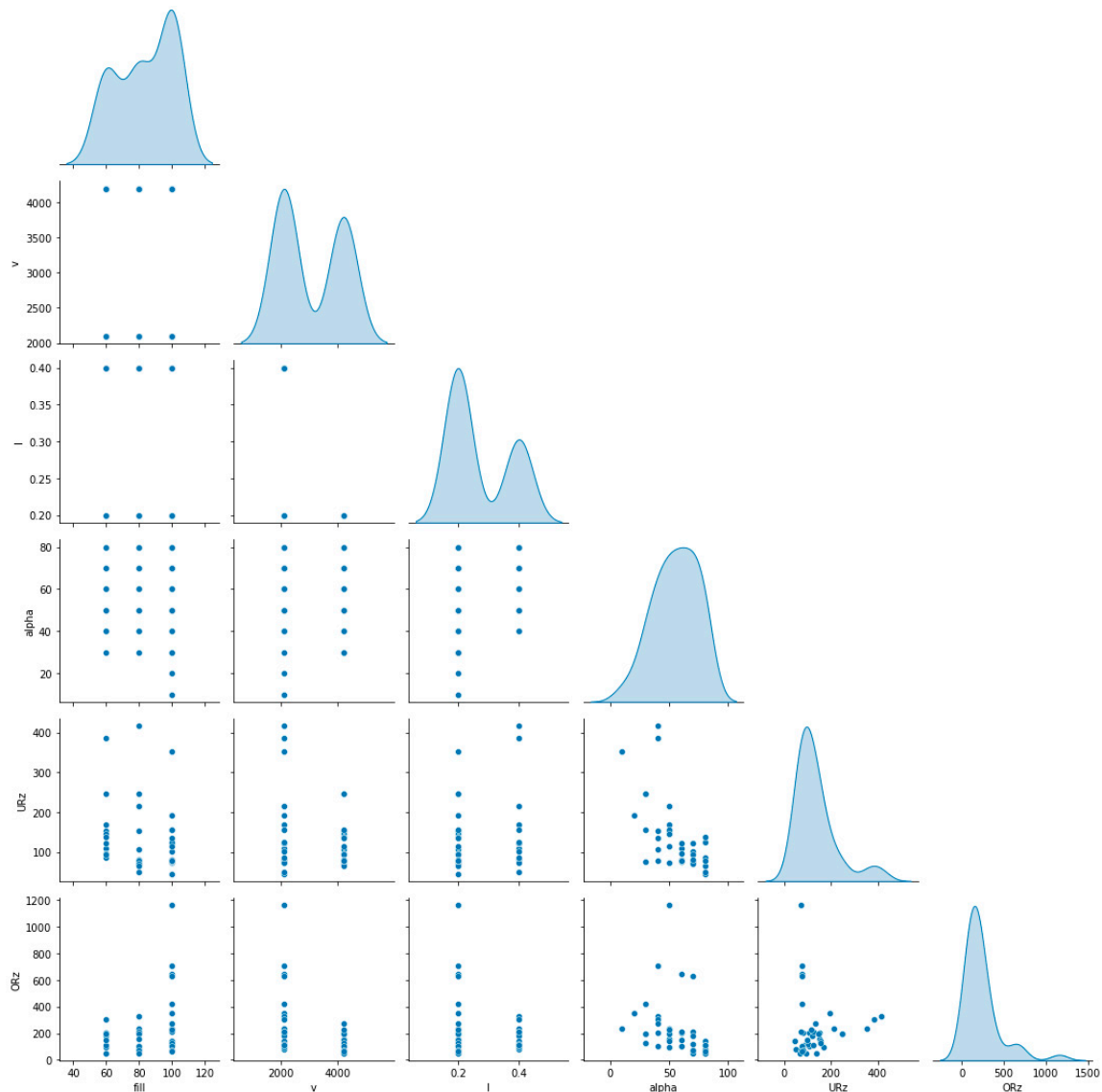


**Figure 6.** Permutation importance of the features for the ML pipeline.

Pairplots are created to visualize the relationships between each pair of features and are displayed in Figure 7. The absence of a clear linear relationship between the input variables and the target variables suggests the necessity of nonlinear models. The features exhibit only slight visual linear correlations, indicating a degree of independence that is advantageous for many machine learning models. The varying distribution of points highlights the need for normalization, which can be achieved through scaling. Ultimately, the target values indicate specific asperity ranges.

The selection of algorithms for model training was based on the criteria of robustness, prediction accuracy and interpretability. The models considered include linear regression (LR) based on its simplicity and good interpretability for linear relationships [33], random forest regressor (RFR) due to its robustness to outliers and ability to model nonlinear relationships [34], support vector machines (SVM) because the method is suitable for small datasets and offers good generalization capabilities [35], k-nearest-regressor (kNN) based on its simple implementation and adaptation to nonlinear structures [36,37], multilayer perceptron (MLP) for potentially more complex correlations and pattern recognition [38–40], and finally a bagging regressor (Bag) with kNN [41], decision tree regressors, or MLP as base estimators to reduce the variance and increase the stability of the model. The kNN algorithm is based on the fundamental principle of identifying the  $k$  closest data points based on the input data and a distance metric [42]. The common distance metric is the calculation of the Euclidean distance, which is described in Equation (1) [43].

$$d_E(x, y) = \sqrt{\sum_{i=1}^k (x_i - y_i)^2} \quad (1)$$



**Figure 7.** Pairplot displaying feature relationships and distributions.

The algorithm is based on several steps. First, the number of  $k$  neighbors is determined. Second, the distances between the object to be classified and all data points are then calculated. All distances are sorted in ascending order to identify the  $k$  nearest neighbors. Finally, the classes of the  $k$  nearest neighbors are then considered, and based on these classes, the class of the object is determined [43].

RFR is an ensemble algorithm that generates a large number of decision trees during training and combines their outputs to increase the accuracy of the predictions. The main principles are based on ensemble learning, bootstrap aggregation, feature randomness, and prediction [44]. The supervised learning model SVM uses kernel functions to transform nonlinear data into a higher dimension [45], also known as a hyperplane, in order to perform linear classification on the data [43]. MLPs are a type of neural network consisting of several layers of nodes. These include the input layer, one or more hidden layers, and an output layer. Each neuron in one layer is connected to every neuron in the subsequent layer [46]. The MLP output is calculated using a weight function  $s_j$ , which is described in Equation (2) as follows [43]:

$$s_j = \sum_{i=1}^n (W_{ij} \cdot X_i) - \delta_j, \quad j = 1, 2, \dots, t \quad (2)$$

The factor  $X_i$  is the  $i$ th input,  $W_{ij}$  represents the weight amount from the  $i$ th node to the  $j$ th node,  $n$  is the number of nodes, and  $\delta_j$  is the threshold of the hidden node. For the calculation of the output, the following equations are used:

$$o_p = \sum_{i=1}^n (w_{jp} \cdot S_j) - \vartheta_p, \quad p = 1, 2, \dots, g \quad (3)$$

$$\text{with } S_j = \frac{1}{1 + e^{-s_j}} \quad (4)$$

$w_{jp}$  is the connection weight from the  $i$ th to the  $k$ th node, and  $\vartheta_p$  is the threshold of the  $p$ th output node.

To train the models, a pipeline was defined that includes the following steps: polynomial features, SelectKBest, scaling, and the respective estimator. To evaluate the model performance, the RMSE and the  $R^2$  value were defined as common error measures [47], supplemented by a qualitative evaluation using plots of the correct versus the predicted values. The RMSE is given in the respective unit of the target value— $\mu\text{m}$  for URz and ORz and percentage points in % for density. RMSE and  $R^2$  are defined in [9].

To optimize the models, hyperparameter tuning is performed using grid search (Grid-SearchCV) [27] with cross-validation (CV). The corresponding parameter spaces are given for each algorithm in Table 4. All pipelines consider a variation of the scaler, SelectKBest, and polynomial features in the grid search.

**Table 4.** Selected hyperparameter for all investigated ML models.

Algorithm	Parameter	Values	
Polynomial features	activated	True, False	
KBest	k	5, 10, 15, 'all'	
Scaler	scaler	MinMaxScaler(), StandardScaler()	
Estimator	RFR	n_estimators	10, 50, 100
		max_depth	None, 2, 5, 10
		min_samples_leaf	1, 2, 3
		min_samples_split	2, 5
		max_features	None, sqrt
		bootstrap	True, False
		n_neighbors	3, 5, 7
	kNN	weights	uniform, distance
		algorithm	auto, ball_tree, kd_tree
		p	1, 2
	MLP	hidden_layer_sizes	(10,), (20,), (10, 10,), (25, 25,)
		alpha	0.0001, 0.001, 0.01
		activation	relu, tanh
		solver	lbfgs, adam
learning_rate		constant, adaptive	
Bag	estimator	kNN, DTR, MLP	
	n_estimators	5, 10, 20	
	max_samples	0.5, 0.7, 1.0	
	bootstrap	True, False	
	bootstrap_features	True, False	

In addition, to evaluate the generalization capability of the final models, a separate k-fold cross-validation (with  $k = 5$ ) was performed on the full dataset. The resulting mean performance metrics (e.g.,  $R^2$ , RMSE) and standard deviations provide insight into the model's stability and expected real-world performance, even under limited data availability, and are displayed in Table 5.

**Table 5.** Mean performance metric for the output data.

	URz	ORz	Density
RMSE	$69.41 \pm 44.1$	$157.29 \pm 163.18$	$2.63 \pm 0.84$
$R^2$	$0.19 \pm 0.32$	$-1.52 \pm 2.33$	$0.85 \pm 0.14$

For the training of the ML model, a statistical evaluation was performed, starting with grid-search-based parameter tuning. The search over the respective parameter spaces is performed 10 times each. The best configurations are then selected. The models are then trained 50 times with the selected parameter combinations, and the mean error is determined. This procedure ensures optimal parameter selection and checks the robustness of the model.

#### 4. ML Results

The various ML models LR, RFR, SVM, kNN, MLP, and Bag were trained and evaluated with hyperparameter tuning. Subsequently, the  $R^2$  factor and the RSME for downskin and upskin angles of the experimental data were evaluated (compare Figure 3). Table 6 shows the respective values of each model. The highest  $R^2$  value in the training dataset can be seen in the MLP model for the URz.

**Table 6.**  $R^2$  value and RMSE of URz and ORz in training and test case for all ML models.

Model	Target Value	Indicators	Training	Test
LR	URz	$R^2$	0.55	0.51
		RMSE [ $\mu\text{m}$ ]	57.44	60.21
	ORz	$R^2$	0.58	0.41
		RMSE [ $\mu\text{m}$ ]	109.34	237.45
RFR	URz	$R^2$	0.86	0.70
		RMSE [ $\mu\text{m}$ ]	32.04	47.51
	ORz	$R^2$	0.89	0.31
		RMSE [ $\mu\text{m}$ ]	56.79	255.7
SVM	URz	$R^2$	0.44	0.39
		RMSE [ $\mu\text{m}$ ]	63.79	67.60
	ORz	$R^2$	0.26	0.1
		RMSE [ $\mu\text{m}$ ]	144.90	293.12
kNN	URz	$R^2$	0.66	0.65
		RMSE [ $\mu\text{m}$ ]	49.80	51.27
	ORz	$R^2$	0.73	0.69
		RMSE [ $\mu\text{m}$ ]	87.98	170.96
MLP	URz	$R^2$	0.95	0.26
		RMSE [ $\mu\text{m}$ ]	18.76	74.66
	ORz	$R^2$	0.99	0.13
		RMSE [ $\mu\text{m}$ ]	13.71	288.44
Bag	URz	$R^2$	0.90	0.53
		RMSE [ $\mu\text{m}$ ]	27.15	59.10
	ORz	$R^2$	0.75	0.60
		RMSE [ $\mu\text{m}$ ]	84.63	194.01

RFR and Bag also show high values of 0.86 and 0.9, respectively. Similarly, these models also have low RMSE values for the training data (MLP 18.75  $\mu\text{m}$ , RFR 32.04  $\mu\text{m}$  and

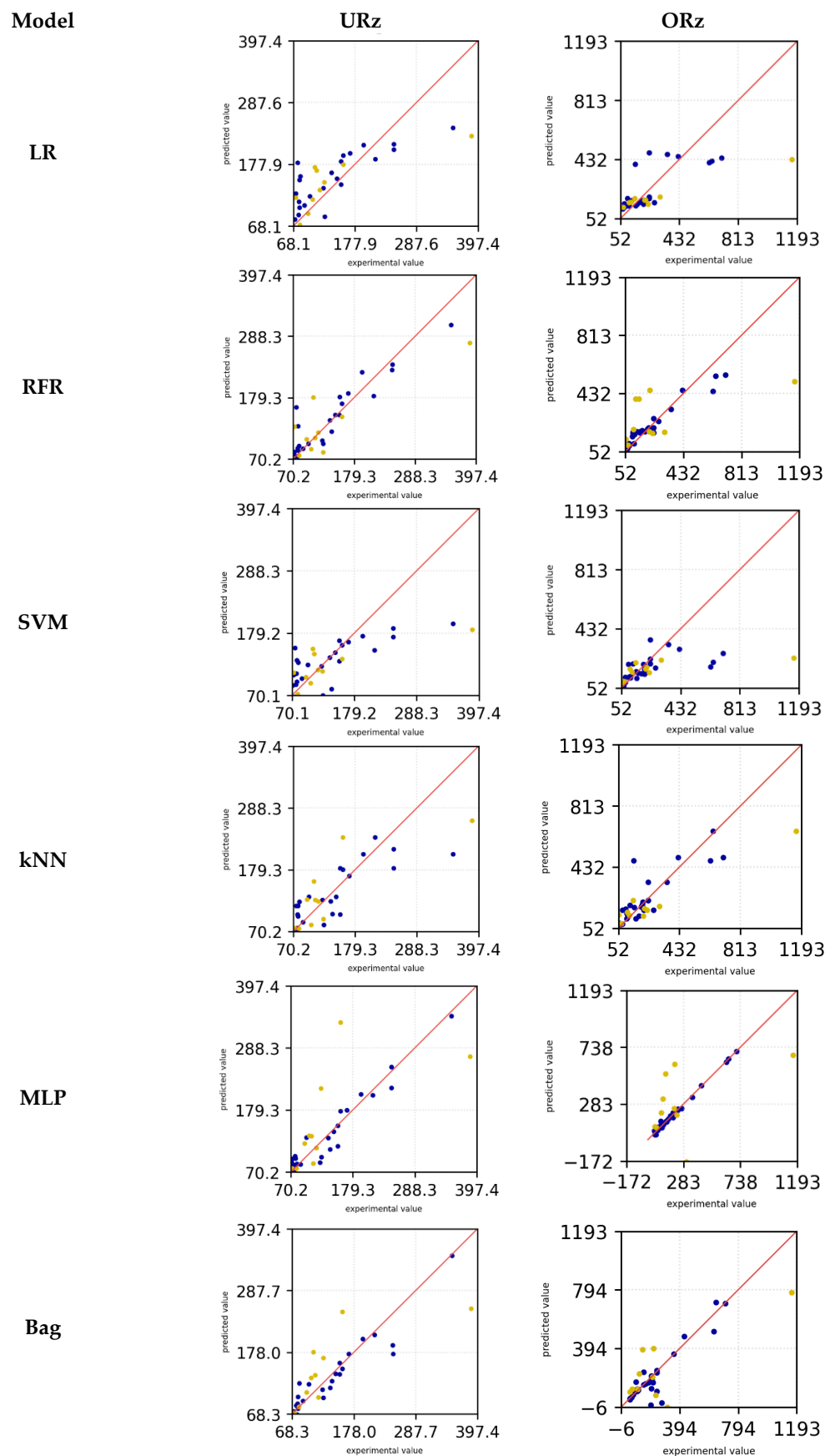
Bag 27.15  $\mu\text{m}$ ). A similar observation can be made in the training dataset for ORz, in which MLP (0.99) also has the highest  $R^2$  value, followed by RFR (0.88) and Bag (0.75). SVM has the lowest  $R^2$  value in the training dataset for URz with 0.44, followed by LR with a value of 0.55. The RMSE value is highest for SVM at 63.79  $\mu\text{m}$  and for LR at 57.44  $\mu\text{m}$ . In the case of ORz, the  $R^2$  factor is also lowest for SVM with 0.26, followed by LR with 0.58. Here, the RMSE values are 144.90  $\mu\text{m}$  (SVM) and 109.34  $\mu\text{m}$  (LR). In the test dataset, the RFR model achieved a maximum coefficient of determination ( $R^2$ ) of 0.7 for the target variable URz, while the kNN model attained an  $R^2$  value of 0.65. The corresponding RMSE values were 51.27  $\mu\text{m}$  and 47.52  $\mu\text{m}$ , respectively. For the target variable ORz, the kNN model also recorded the highest  $R^2$  value of 0.69, with an RMSE value of 170.96  $\mu\text{m}$ .

Additionally, diagrams of the respective models were generated, illustrating the predictions compared to the experimental data. The red line indicates a perfect prediction. The farther the points are from the line, the less accurate the predictions are. Figure 8 presents the prediction diagrams of the models under examination, facilitating an initial selection of the models in question. The blue dots depict the training data, while the yellow dots represent the test data. For the LR model, it is observed that both the test and training data predictions deviate from the experimental data. This phenomenon is observable for both the downskin angle URz and the upskin angle ORz. Only for small overhang angles do the predictions appear to align more closely with the experimental data, resulting in smaller deviations for both URz and ORz. The SVM model exhibits a behavior similar to that of the LR model's predictions. In this instance as well, the predictions significantly deviate from the actual values. Both the LR and SVM models tend to underestimate the actual surface roughness for URz and ORz. In contrast, the RFR model demonstrates higher accuracy in predicting the training data with larger URz values, a trend that is also observed in the test data. Nonetheless, in both scenarios, the roughness prediction is underestimated.

The kNN model demonstrates that more accurate predictions can be achieved for low URz and ORz values. However, for higher roughness, the predictions deviate from the experimental data. In this context, both the training and test datasets underestimate the actual values for larger roughness values (>200  $\mu\text{m}$ ). One reason for this is the low experimentally recorded roughness values. The majority of the data (see Figure 3) show a roughness of less than 250  $\mu\text{m}$ . Because the kNN algorithm makes predictions based on the  $k$  nearest neighbors, smaller roughness values are identified as neighbors, resulting in an underestimation of the roughness. The Bag model exhibited a high level of predictive accuracy for URz in the training dataset, including for large roughness values. Nevertheless, for small roughness values, the model showed a tendency to overestimate predictions in the test dataset, whereas for large roughness values, it occasionally underestimated the predictions. In the case of ORz, the training data is even more accurate in its prediction and tends to underestimate small roughness values.

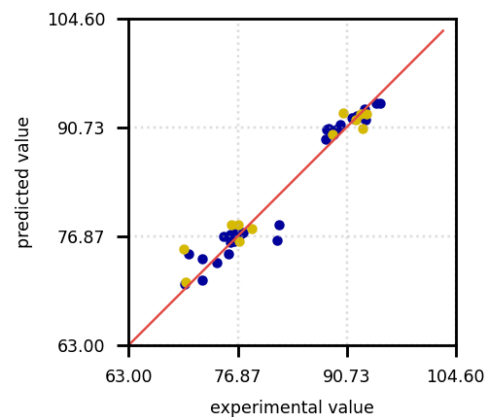
The test data overestimate the experimental data and predict higher values for the roughness. However, for large roughnesses, the predictions in the test dataset tend to be underestimated. In the training dataset, the MLP model shows the smallest deviations in the predicted values for both ORz and URz. Only in the test data, small roughnesses are overestimated and larger roughnesses are underestimated.

Because the training data for MLP and Bag are heavily aligned with the experimental data, while the test data fluctuate more around the exact prediction, the machine learning models appear to be overfitted. For URz and ORz, the MLP model appears to perform the best despite signs of overfitting and is therefore selected for further parameter optimization. The other algorithms are not further investigated, as MLP seems to be the most promising algorithm for this dataset.



**Figure 8.** Surface roughness prediction versus experimental measurements in [ $\mu\text{m}$ ] for the training (blue) and test data (yellow) of all investigated ML models.

The same approach was applied to the target variable density, where, with parameter optimization using polynomial feature importances and the SelectKBest method, the Bag model appears to be the best starting point. The prediction diagram is shown in Figure 9. It can be observed that the predictions for both the training data and the test data are located close to the actual values. Occasionally, the predictions for the training data are underestimated, which may indicate that overfitting is not occurring. Both training and test data oscillate within a similar range around the optimal axis, which means that a good generalization of the model can be assumed. In the following, only the Bag model will be investigated further for the density prediction.



**Figure 9.** Density prediction versus experimental measurements in percentage points for the training (blue) and test data (yellow).

The hyperparameter optimization over 10 runs using polynomial feature importances and SelectKBest led to the following results in terms of the  $R^2$  factor and RMSE for URz, Orz, and density (see Table 7). A direct comparison with Table 6 for the MLP model shows a significant increase in the  $R^2$  value from 0.25 to approximately 0.71. Additionally, the RMSE was reduced by more than 25  $\mu\text{m}$  to around 49.27  $\mu\text{m}$ . Although the  $R^2$  factor in the training dataset decreased to 0.83 and the RMSE increased to 34.82  $\mu\text{m}$ , the model initially seemed to be overfitted, so the optimization counteracted this trend. The standard deviation of the  $R^2$  value for URz is 0.11 for the training data and 0.3 for the test data. Thus, the variability among the training data is less than that of the test data. This suggests variability in the model's performance and indicates the need for further iterations to achieve a more robust  $R^2$  value. The RMSE for both the training data and test data also exhibits significant variability, making it difficult to draw definitive conclusions about the error of the mean values after 10 iterations. This variability suggests that the MLP model is sensitive to the hyperparameters, necessitating further optimization iterations to achieve greater robustness.

**Table 7.** Hyperparameter optimization after 10 iterations for the MLP and Bag models.

	Value	URz (MLP)	ORz (MLP)	Density (Bag)
Train	R2	$0.83 \pm 0.11$	$0.72 \pm 0.12$	$0.96 \pm 0.02$
	RMSE	$34.82 \pm 13.82 [\mu\text{m}]$	$90.8 \pm 34.7 [\mu\text{m}]$	$1.64 \pm 0.34 [\%]$
Test	R2	$0.71 \pm 0.3$	$0.46 \pm 0.15$	$0.91 \pm 0.04$
	RMSE	$49.27 \pm 18.95 [\mu\text{m}]$	$222.92 \pm 41.41 [\mu\text{m}]$	$2.53 \pm 0.53 [\%]$

A similar observation can be made for the ORz MLP model. Here, the  $R^2$  value decreased from 0.99 to 0.72, and the RMSE increased from 13.71  $\mu\text{m}$  to approximately 90.8  $\mu\text{m}$  in the training dataset from the initial iteration step (compare Table 6). The  $R^2$

value for the test data, on the other hand, increased from 0.12 to around 0.46, and the RMSE error was reduced to 222.92  $\mu\text{m}$ . The reduction of the  $R^2$  value in the training data is an indication that overfitting is counteracted, but the high standard deviations of  $R^2$  and RMSE indicate a low robustness of the model.

The Bag model was used for the density forecast. In the training dataset as well as in the test dataset, the  $R^2$  value is already high at 0.96 and 0.91, respectively, so that the independent variables can be well described by the number of dependent variables. The standard deviation is also much smaller than for the other two MLP models, which means that the model can be expected to be more robust. The same applies to the RMSE, which is small at 1.64% (training) and 2.53% (test) and also has a standard deviation of less than 0.55%.

After completing 10 iterations, the optimal hyperparameters were determined by maximizing the coefficient of determination and minimizing the root mean square error. Table 8 shows the values with the optimized hyperparameters. The selected hyperparameters for the MLP model are illustrated in Table 9 and for the Bag model in Table 10.

**Table 8.** Parameter optimization (best models).

	Value	URz (MLP)	ORz (MLP)	Density (Bag)
Train	$R^2$	0.96	0.91	0.98
	RMSE	16.04 [ $\mu\text{m}$ ]	51.29 [ $\mu\text{m}$ ]	1.12 [%]
Test	$R^2$	0.88	0.91	0.95
	RMSE	32.11 [ $\mu\text{m}$ ]	91.89 [ $\mu\text{m}$ ]	1.97 [%]

**Table 9.** Parameter optimization (selected hyperparameter)—URz, ORz.

Algorithm	Parameter	URz	ORz
Polynomial features	activated	False	True
KBest	K	-	10
Scaler	scaler	MinMaxScaler	MinMaxScaler
Estimator	model	MLP	
Hyperparameters	hidden_layer_sizes	(10,)	(10,)
	alpha	0.01	0.01
	activation	relu	relu
	solver	lbfgs	lbfgs
	learning_rate	adaptive	adaptive

**Table 10.** Parameter optimization (selected hyperparameter)—density.

Algorithm	Parameter	Density
Polynomial features	activated	True
KBest	K	15
Scaler	scaler	StandardScaler
Bag	estimator	DecisionTreeRegressor
	n_estimators	10
	max_samples	0.7
	bootstrap	false
	bootstrap_features	True

An  $R^2$  of 0.96 shows that the model can explain 96% of the variance of the surface roughness (URz) in the training data. The RMSE is also significantly lower here at 16.04  $\mu\text{m}$ . In the test dataset, the  $R^2$  factor was also increased to 0.88 and also has a low RMSE value of 32.11  $\mu\text{m}$ . A significant increase in the error on the test data indicates that the model has difficulties predicting the test data with the same precision as the training data. There is a possibility of a slight overfitting effect occurring in this case. The  $R^2$  factor of 0.91 does not change for ORz. A constant  $R^2$  value indicates that the model has learned the patterns in the data well and also transfers them to the test data. An RMSE increase from 51.29  $\mu\text{m}$  (training) to 91.89  $\mu\text{m}$  (test) strongly indicates overfitting. Regularization, optimized data processing, and model adjustments can improve the generalization and reduce the test error. For the Bag model for density, the  $R^2$  values of 0.98 (training) and 0.95 (test) are also high. The RMSE is lower in the training (1.12%) than in the test (1.97%). Further optimization iterations should reduce the different errors.

Table 11 indicates the averaged evaluation parameters of 50 iterations with the optimized hyperparameters. The observed discrepancy between training and test data indicates that the model performs better on the training set, which could suggest some degree of overfitting. However, the low standard deviation across both datasets demonstrates a high level of robustness and consistency in the model's predictions. For ORz, the  $R^2$  factor is 0.94 in the training dataset and 0.73 in the test dataset, with a standard deviation of 0.07. Although the performance on the test data is lower, the small standard deviation highlights the stability of the model across samples and its delivery of similar results. Similarly, for density, the  $R^2$  factor is 0.98 on the training set with a standard deviation of 0.003, and 0.92 on the test set with a standard deviation of 0.02. This consistency in performance, coupled with the relatively small deviations, reinforces the robustness of the model, even if the performance on the test data does not reach the same level as that on the training data. Overall, while there is evidence of better training data performance, the low standard deviations suggest that the model's predictions are reliable and not overly sensitive to variations within the datasets. For the RMSE, the error for URz during training is 14.32 with a standard deviation of 3.27, reflecting improvements compared to the results obtained after hyperparameter optimization with 10 iterations. Furthermore, the error in the test set was reduced to 34.93, accompanied by a standard deviation of 6.28. These results indicate that the optimization process enhanced both the accuracy and stability of the predictions for URz.

**Table 11.** Model training with best parameters (50 runs)—MLP, MLP, and Bagging.

	Value	URz (MLP)	ORz (MLP)	Density (Bag)
Train	R2	0.97 $\pm$ 0.01	0.94 $\pm$ 0.06	0.98 $\pm$ 0.003
	RMSE	14.32 $\pm$ 3.27 [ $\mu\text{m}$ ]	47.67 $\pm$ 16.45 [ $\mu\text{m}$ ]	1.06 $\pm$ 0.11 [%]
Test	R2	0.85 $\pm$ 0.06	0.73 $\pm$ 0.07	0.92 $\pm$ 0.02
	RMSE	34.93 $\pm$ 6.28 [ $\mu\text{m}$ ]	155.63 $\pm$ 20.47 [ $\mu\text{m}$ ]	2.51 $\pm$ 0.29 [%]

For ORz, the RMSE during training is 47.67  $\mu\text{m}$  with a standard deviation of 16.45  $\mu\text{m}$ . However, the test RMSE is significantly higher at 155.63  $\mu\text{m}$ , with a standard deviation of 20.47  $\mu\text{m}$ . This suggests that, while the model achieved reasonable performance in training, its ability to generalize to test data is more limited for ORz. Additionally, both the roughness and deviations are notably higher for ORz compared to URz.

Importantly, despite the higher error and variability in the predictions for ORz, there was a reduction in the standard deviation for both the training and test datasets when compared to the results obtained after 10 iterations of hyperparameter optimization (as seen in Table 7). This improvement in stability indicates that the optimization process

contributed to reducing variability in the predictions, even if the overall performance still shows room for improvement. For density, the RMSE in training is 1.06% with a standard deviation of 0.11%, while in the test dataset, it increases to 2.51% with a deviation of 0.29%. This indicates a noticeable performance drop when transitioning from training to test data. Furthermore, the test RMSE and its deviation are higher compared to the results after 10 iterations of hyperparameter optimization.

Using the selected hyperparameters from Tables 9 and 10 for the MLP and Bag models, the corresponding values for RMSE and  $R^2$  as best results are presented in Table 12. These results offer insights into how the optimized hyperparameters affect model performance in terms of accuracy and explanatory power. For URz, the model demonstrated a strong fit with an  $R^2$  of 0.96 in the training set, indicating that 96% of the variance in the data was explained. However, the test set  $R^2$  decreased to 0.87, and the RMSE increased from 17.38  $\mu\text{m}$  in training to 32.93  $\mu\text{m}$  in testing, suggesting some overfitting. The ORz results showed a similar trend, with a high  $R^2$  of 0.94 in the training set, but a significant drop to 0.79 in the test set. Additionally, the RMSE for ORz was substantially higher, moving from 41.58  $\mu\text{m}$  in training to 138.99  $\mu\text{m}$  in testing, indicating a severe generalization issue. In contrast, the model performed exceptionally well for density, achieving an  $R^2$  of 0.99 in the training set and 0.95 in the test set, with an RMSE of 0.95% and 1.88%, respectively. The minimal difference between the training and test RMSE values for density suggests that the model generalizes well to unseen data. Overall, the results highlight that while the model is highly effective for density, additional improvements in feature selection and regularization are needed to enhance the generalization performance for URz and especially ORz, where overfitting is more pronounced.

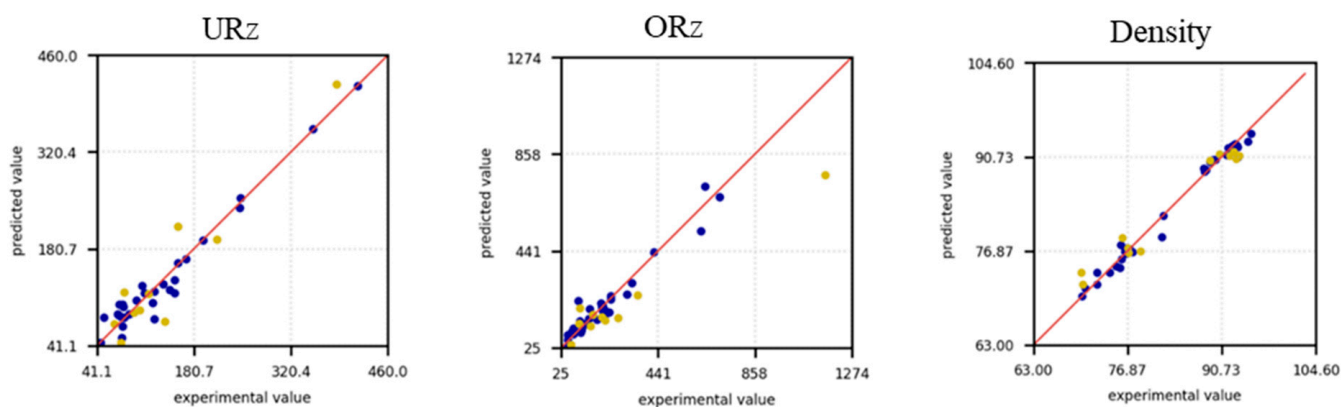
**Table 12.** Model training with best models (best models).

	Value	URz (MLP)	ORz (MLP)	Density (Bag)
Train	R2	0.96	0.94	0.99
	RMSE	17.38 [ $\mu\text{m}$ ]	41.58 [ $\mu\text{m}$ ]	0.95 [%]
Test	R2	0.87	0.79	0.95
	RMSE	32.93 [ $\mu\text{m}$ ]	138.99 [ $\mu\text{m}$ ]	1.88 [%]

Figure 10 shows the predictions of the experimental values as a further evaluation criterion. In this case too, the blue dots are the training dataset, and the yellow dots are the test data. For URz (left) it can be seen that the trending data for small roughness values deviate from the exact values and make precise predictions with increasing roughness. The scattering for small roughness values can be observed in the test data. With increasing roughness, a deviation from the experimental data can still be observed, but this deviation remains constant with increasing roughness values.

For ORz (Figure 10, center), the model demonstrates superior predictive accuracy for small roughness values (<441  $\mu\text{m}$ ), as evidenced by better alignment with the training data and reduced scatter in the predictions. This indicates that the model effectively captures the underlying patterns within this range, likely due to a higher density of data points or less variability in the observed values. However, as roughness increases beyond this threshold, the accuracy of the predictions deteriorates, accompanied by a noticeable increase in scatter. This decline suggests that the model struggles to generalize to larger roughness values, potentially due to a lack of sufficient data or increased complexity in the relationships governing higher roughness levels. Addressing this issue may require additional data for large roughnesses or the inclusion of features that better capture the variability in this range. The situation is the same for the test data of ORz, although the deviation of the predictions is generally significantly larger. In the density prediction (Figure 10, right), the

training data are well adapted to the experimental data, and an accurate prediction with few outliers is made for both low and high density values. For the test data, the predictions for low density values tend to be overestimated with a few outliers, while for high density predictions, the values are primarily underestimated.



**Figure 10.** Surface roughness prediction versus experimental measurements for the training (blue) and test data (yellow) of all investigated ML models for URz (left) in  $\mu\text{m}$ , ORz (center) in  $\mu\text{m}$ , and density (right) percentage points.

## 5. Discussion

The initial attempt to utilize and optimize a single ML model for predicting all three parameters—URz, ORz, and density—was unsuccessful. The primary limitation was the small size of the measurement dataset, coupled with the substantial variability in the parameter values. These factors hindered the model's ability to capture consistent patterns across all targets, ultimately preventing it from providing reliable and usable predictions. Furthermore, the small dataset restricts the model's ability to generalize to unseen data, and with the limited samples, the risk of overfitting increases, reducing predictive reliability for new process conditions. Additionally, the small datasets may not fully capture the variability in process parameters, leading to biased or unstable model performance. This observation underscores the need for either an increased dataset size, feature-specific models, or tailored preprocessing strategies to better account for the distinct characteristics and variances of each parameter in order to improve model robustness and generalizability. As a result, it became necessary to train and test the models individually for each parameter. The evaluated models LR, RFR, SVM, kNN, MLP, and Bag demonstrated varying levels of performance and robustness during hyperparameter optimization. As anticipated, the LR model failed to achieve a high  $R^2$  value and a low RMSE due to the complexity of the relationships between input and output variables. The predictions from the LR model exhibited significant deviations from the actual experimental data, indicating that this approach is unsuitable for accurately modeling the investigated parameters. According to [9], this is an indication that conventional regression models cannot solve the problem and that an ML approach is required. This can be attributed to the inherent limitations of LR in capturing complex, nonlinear relationships within the data. The MEX/M process involves multiple interacting parameters, such as material properties, process conditions, and geometric features, which exhibit nonlinear dependencies. Because LR assumes a linear relationship between the input features and the target variables, it was unable to adequately model the intricate patterns present in the dataset. Consequently, the LR model showed poor generalization performance and was deemed unsuitable for making accurate predictions in this context.

After 10 optimization iterations, the RFR model showed worse results than the MLP and Bag models. A negative value of  $R^2$  of  $-1.97$  led to the exclusion of the model from

further investigations. A negative  $R^2$  value means that the model performs worse than a trivial model that only uses the mean value of the target variable as a prediction. Specifically, the model not only does not explain any variance in the data, but also provides predictions that increase the error variance, which is also observed in [48]. For RFR, the negative  $R^2$  values may be attributed to over-complexity in the model, such as an excessive number of trees or insufficient regularization during hyperparameter tuning. This caused the model to overfit to subtle patterns or noise in the training dataset, which were not representative of the overall data distribution. In both cases, negative  $R^2$  values highlight that the models performed worse than a simple baseline prediction (e.g., predicting the mean of the target variable). This underscores the importance of carefully balancing model complexity and regularization during hyperparameter optimization to ensure reliable predictions and robust generalization. Moreover, with a maximum of 48 data points, the dataset is relatively small and noisy in this case, which prevents the RFR model from learning meaningful patterns and can also result in a negative  $R^2$  value. In addition, because some of the input features correlate weakly with the target values, the model struggles to make accurate predictions [44].

The SVM model appears to have difficulties in accurately capturing the variability of the URz and ORz values. The data could be nonlinearly separable or have a highly nonlinear pattern that the SVM model does not capture effectively. With a small number of data points, SVM may have difficulty making robust predictions, as it is highly dependent on the number and quality of support vectors, also observed in [49]. SVM is sensitive to unscaled input data. If the data have not been correctly normalized or standardized, this can affect performance. A similar observation was made in the roughness study of SVM in [50] where the accuracy of 0.56 was also the worst for roughness predictions. This poor performance can be attributed to several factors inherent to the nature of the dataset and the characteristics of the SVM model. Firstly, SVM models, especially when used with nonlinear kernels, can struggle with small datasets that have high variability or noise. The limited number of experimental data points in this study likely led to overfitting during training, where the model attempted to fit to every detail in the data, including noise. This overfitting reduced the model's ability to generalize to unseen data, resulting in poor test performance. Secondly, the high dimensionality or complex feature interactions in the dataset may have exacerbated the model's inability to find an optimal decision boundary. SVM requires careful tuning of hyperparameters such as the regularization parameter  $C$  and the kernel-specific parameters (e.g., gamma for the RBF kernel). Without sufficient data to guide this tuning process effectively, the model's performance likely deteriorated further. Lastly, SVMs are sensitive to feature scaling and may fail to perform well if input features are not properly normalized. Given these challenges, SVM was unable to capture the intricate relationships in the dataset, resulting in poor predictive accuracy and high error rates, making it the least effective model for this application.

For the kNN model, a negative  $R^2$  factor also appeared in the test for URz after 10 training iterations. In addition, a value of 1 and an RMSE of 0 occurred in the training for the  $R^2$  value of ORz. A negative  $R^2$  value indicates that the kNN model's predictions are worse than a simple baseline model. This suggests that the model fails to capture the patterns in the test data, which may be due to overfitting to the training data. An  $R^2$  value of 1 and an RMSE of 0 indicate that the kNN model memorized the training data perfectly for ORz. While this may seem ideal, it often signals overfitting, where the model performs exceptionally well on the training data but struggles to generalize to unseen test data. This issue arises because the kNN algorithm stores the training dataset and bases predictions on the closest neighbors. If  $k$  is too small (e.g.,  $k = 1$ ), the model essentially memorizes the training points, making it sensitive to noise and unable to generalize well to

unseen data. This behavior indicates that the models were overfitting to the training data, leading to a significant loss of generalization ability. For kNN, this issue likely arose from an inappropriate selection of the number of neighbors (k). Therefore, the kNN model was not considered further for parameter optimization.

The experimentally measured surface roughnesses are significantly higher than the roughness values found in the literature due to the overhang angle [51–53]. As shown in [51], the process parameters have a significant influence on the surface roughness of the components. A comparison with the recorded measurement data shows that the geometry of the inclined walls has a much greater influence (factor 10) than the parameter variations. The measured values, which act as a dataset for the ML models, also vary considerably. For URz, the variance in the measurement data is notably lower compared to ORz, which facilitates more effective training and hyperparameter optimization. The reduced variability allows the model to capture the underlying patterns more reliably, leading to improved predictive accuracy. After 50 iterations of hyperparameter optimization, the predictions for URz exhibit higher precision and better alignment with experimental values compared to ORz. This suggests that the lower variance in the measurement data for URz not only enhances model training but also contributes to improved generalization performance. In contrast, the higher variance observed in the ORz data likely introduces additional challenges in modeling and optimization, resulting in less accurate predictions.

The average accuracy of the predictions is defined as:

$$\text{Average accuracy} = 1 - \frac{RMSE}{\sum_1^n R_{z_n}} \quad (5)$$

The accuracy in the test dataset for URz is 75.71%, which makes a prediction possible. For ORz, the average accuracy is only 39.26% and an exact prediction is not generally feasible. The MLP model performed more accurately for the prediction of URz than for ORz based on the evaluation criteria examined and the predictions of the test and training data. The representativeness and size of the training dataset are critical for the model's ability to generalize to new data [54]. Because the ORz data are highly variable and there is only a small amount of data, an accurate prediction is difficult and requires the input of more data. In addition, even after 50 training sessions, the standard deviation of 20.47  $\mu\text{m}$  in the test case is a power of 10 higher than for URz, and a prediction is inadmissible. Larger roughness values may be associated with increased complexity or variability in the underlying physical processes such as the staircase effect [55,56], making them harder to predict accurately.

The density of the sample geometries was experimentally determined to range between 70% and 95% using the Archimedes method. While this does not meet the typical standards for metal injection molding (MIM), it falls within the empirically established density range for the material extrusion for metals (MEX/M) process, as reported in the literature [23]. This suggests that the density values observed are consistent with the inherent characteristics of the MEX/M process, which is known to result in slightly lower densities compared to MIM due to differences in processing parameters and material behavior. The average accuracy of the density predictions in the test dataset using the Bag model, calculated according to Equation (5), is 97.44%. This indicates that the model provides highly accurate predictions within the same order of magnitude as the experimental data. The deviation of only 1.55% compared to the accuracy of [4] for the density prediction underscores the reliability of the model in capturing the underlying patterns of the dataset. Such a small deviation highlights the model's robustness and suitability for predicting density in similar experimental setups.

The insights gained from the performance of the suitable ML models make it possible to identify process parameter combinations for the MEX/M processes for the desired surface roughness in a component with overhang features. On the one hand, this can estimate and minimize the post-processing effort. On the other hand, the surface roughness can be determined by prediction for surfaces where post-processing is not possible. As a result, the cost of manufacturing a component using the MEX/M process can also be optimized in the wider sense, and cost estimation in AM processes can be better determined by knowing the more precise post-processing costs [57]. Density prediction in the MEX/M process helps to significantly reduce the experimental effort regarding process parameters in the shaping, debinding, and sintering steps, and in this way it helps to find suitable process parameter combinations that have the highest achievable density. The findings regarding density can contribute to design guidelines in MEX/M green part manufacturing [25] and thus produce functional prototypes. Because there is a strong correlation of the infill percentage with the density in the Bag model, it is now possible to evaluate porous components in the context of a solid material using the density prediction, which has only been investigated experimentally in isolated cases to date [23]. The methodology is basically transferable to other AM processes, so that the density prediction in the PBF-LB/M process can be applied in principle, and thus the material characteristics can be added to the predictions of the maximum stress in [9].

## 6. Conclusions

For the MEX/M process, the upskin and downskin surface roughness values, along with the density, were empirically determined as experimental values for various overhang angles. These measurements provide critical insights into the relationship between geometric features and resulting material properties, which are essential for optimizing process parameters and achieving desired part characteristics in MEX/M applications. The experimental data served as the foundation for predicting the upskin and downskin surface roughnesses, as well as the density of MEX/M samples. To achieve this, machine learning models, including LR, RFR, SVM, kNN, MLP, and Bag, were trained on the dataset. The performance of these models was evaluated using the  $R^2$  coefficient and RMSE as metrics. Additionally, prediction plots were analyzed to assess the accuracy and reliability of the models in capturing the relationships between experimental variables and target properties. The following findings were obtained from the investigations.

The LR model proved to be unsuitable for predicting the upskin and downskin roughnesses, as well as the density, due to its low  $R^2$  values and high RMSE in both training and testing phases.

The SVM model exhibited the lowest  $R^2$  value and the highest RMSE among all evaluated models, making it even less suitable than the LR model for predicting the upskin and downskin roughnesses of MEX/M samples.

The kNN and RFR models demonstrated greater robustness compared to LR, achieving relatively higher  $R^2$  values and lower RMSE during initial evaluations. However, after several iterations of hyperparameter optimization, both models exhibited negative  $R^2$  values for certain target variables, particularly in the test phase.

The MLP model was identified as a strong candidate for predicting the surface roughness (upskin and downskin) in MEX/M samples. Its suitability stems from its ability to handle complex, nonlinear relationships within the data, which are characteristic of the MEX/M process. The MLP model's architecture, consisting of interconnected layers of neurons, enables it to capture intricate patterns and interactions among the input features, making it well-suited for modeling the variability in surface roughness. Moreover, through proper hyperparameter optimization, including the adjustment of the number of hidden

layers, neurons, activation functions, and learning rates, the MLP achieved high  $R^2$  values and low RMSE, demonstrating its robustness and accuracy in both the training and test datasets for URz. Furthermore, an average accuracy of only 39% is too imprecise, so no predictions can be made for ORz with the current dataset. In the case of iterative learners such as MLP, early stopping could be employed based on validation loss to further avoid overfitting to the training data.

The Bag model was identified as a suitable approach for predicting the density of MEX/M samples, achieving an average prediction accuracy of 97.44%. This high level of accuracy can be attributed to the ensemble nature of the Bag method, which combines multiple base models to reduce variance and improve robustness.

An attempt to predict both surface roughness and density within a single model for different stages of the process chain was not feasible. This limitation can be attributed to the fundamentally different natures of the target variables and the complex interactions between process parameters at various stages.

Overall, the failure of regression models underscores the necessity of employing more experimental data and eventually advanced machine learning approaches, such as ensemble methods or neural networks, which are better equipped to handle the nonlinear relationships and complex data structures inherent to additive manufacturing processes.

The results demonstrate that the choice of model and its hyperparameter optimization significantly influence the predictive accuracy for MEX/M sample properties, providing a robust framework for future process optimization and design improvements. With the aid of ML models, it will be possible in future to reduce intensive experimental test series and efficiently find suitable material properties.

**Author Contributions:** Conceptualization, K.A.; Methodology, K.A. and M.K.; Software, K.A. and M.K.; Validation, K.A., T.R. and C.E.; Formal analysis, K.A. and T.R.; Investigation, K.A.; Resources, K.A. and C.E.; Data curation, K.A.; Writing—original draft, K.A. and M.K.; Writing—review & editing, K.A.; Visualization, K.A.; Supervision, K.A. and C.E.; Project administration, K.A.; Funding acquisition, K.A. All authors have read and agreed to the published version of the manuscript.

**Funding:** Publishing fees supported by Funding Programme Open Access Publishing of Hamburg University of Technology (TUHH).

**Data Availability Statement:** The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. ISO/ASTM 52900:2023; Additive Manufacturing—General Principles—Part Positioning, Coordinates and Orientation. ISO: Geneva, Switzerland, 2023.
2. Kuehne, M.; Bartsch, K.; Bossen, B.; Emmelmann, C. Predicting melt track geometry and part density in laser powder bed fusion of metals using machine learning. *Prog. Addit. Manuf.* **2023**, *8*, 47–54. [[CrossRef](#)]
3. Bartsch, K. *Digitalization of Design for Support Structures in Laser Powder Bed Fusion of Metals*; Light Engineering für die Praxis; Springer Nature: Cham, Switzerland, 2023; ISBN 978-3-031-22955-8.
4. Bossen, B.; Kuehne, M.; Kristanovski, O.; Emmelmann, C. Data-driven density prediction of AlSi10Mg parts produced by laser powder bed fusion using machine learning and finite element simulation. *J. Laser Appl.* **2023**, *35*, 042023. [[CrossRef](#)]
5. Gor, M.; Dobriyal, A.; Wankhede, V.; Sahlot, P.; Grzelak, K.; Kluczyński, J.; Łuszczek, J. Density Prediction in Powder Bed Fusion Additive Manufacturing: Machine Learning-Based Techniques. *Appl. Sci.* **2022**, *12*, 7271. [[CrossRef](#)]
6. Kumar, S.; Gopi, T.; Harikeerthana, N.; Gupta, M.K.; Gaur, V.; Krolczyk, G.M.; Wu, C. Machine learning techniques in additive manufacturing: A state of the art review on design, processes and production control. *J. Intell. Manuf.* **2023**, *34*, 21–55. [[CrossRef](#)]
7. Wischeropp, T.M.; Tarhini, H.; Emmelmann, C. Influence of laser beam profile on the selective laser melting process of AlSi10Mg. *J. Laser Appl.* **2020**, *32*, 022059. [[CrossRef](#)]

8. Wu, D.; Wei, Y.; Terpenney, J. Predictive modelling of surface roughness in fused deposition modelling using data fusion. *Int. J. Prod. Res.* **2019**, *57*, 3992–4006. [CrossRef]
9. Asami, K.; Roth, S.; Krukenberg, M.; Röver, T.; Herzog, D.; Emmelmann, C. Predictive modeling of lattice structure design for 316L stainless steel using machine learning in the L-PBF process. *J. Laser Appl.* **2023**, *35*, 042046. [CrossRef]
10. Sivakumar, N.K.; Palaniyappan, S.; Bodaghi, M.; Azeem, P.M.; Nandhakumar, G.S.; Basavarajappa, S.; Pandiaraj, S.; Hashem, M.I. Predictive modeling of compressive strength for additively manufactured PEEK spinal fusion cages using machine learning techniques. *Mater. Today Commun.* **2024**, *38*, 108307. [CrossRef]
11. Wang, H.; Al Shraida, H.; Jin, Y. Predictive modeling for online in-plane shape deviation inspection and compensation of additive manufacturing. *Rapid Prototyp. J.* **2024**, *30*, 350–363. [CrossRef]
12. Rodríguez-Martín, M.; Fueyo, J.G.; Gonzalez-Aguilera, D.; Madruga, F.J.; García-Martín, R.; Muñoz, Á.L.; Pisonero, J. Predictive Models for the Characterization of Internal Defects in Additive Materials from Active Thermography Sequences Supported by Machine Learning Methods. *Sensors* **2020**, *20*, 3982. [CrossRef]
13. Sarkon, G.K.; Safaei, B.; Kenevisi, M.S.; Arman, S.; Zeeshan, Q. State-of-the-Art Review of Machine Learning Applications in Additive Manufacturing; from Design to Manufacturing and Property Control. *Arch. Comput. Methods Eng.* **2022**, *29*, 5663–5721. [CrossRef]
14. Zhang, Y.; Safdar, M.; Xie, J.; Li, J.; Sage, M.; Zhao, Y.F. A systematic review on data of additive manufacturing for machine learning applications: The data quality, type, preprocessing, and management. *J. Intell. Manuf.* **2023**, *34*, 3305–3340. [CrossRef]
15. Mahmoud, D.; Magolon, M.; Boer, J.; Elbestawi, M.A.; Mohammadi, M.G. Applications of Machine Learning in Process Monitoring and Controls of L-PBF Additive Manufacturing: A Review. *Appl. Sci.* **2021**, *11*, 11910. [CrossRef]
16. Razvi, S.S.; Feng, S.; Narayanan, A.; Lee, Y.-T.T.; Witherell, P. A Review of Machine Learning Applications in Additive Manufacturing. In Proceedings of the 39th Computers and Information in Engineering Conference, Anaheim, CA, USA, 18–21 August 2019; American Society of Mechanical Engineers: Houston, TX, USA, 2019; Volume 1, p. V001T02A040.
17. Babu, S.S.; Mourad, A.-H.I.; Harib, K.H.; Vijayavenkataraman, S. Recent developments in the application of machine-learning towards accelerated predictive multiscale design and additive manufacturing. *Virtual Phys. Prototyp.* **2023**, *18*, e2141653. [CrossRef]
18. Gaikwad, M.U.; Gaikwad, P.U.; Ambhore, N.; Sharma, A.; Bhosale, S.S. Powder Bed Additive Manufacturing Using Machine Learning Algorithms for Multidisciplinary Applications: A Review and Outlook. *Recent. Pat. Mech. Eng.* **2025**, *18*, 12–25. [CrossRef]
19. Jyeniskhan, N.; Keutayeva, A.; Kazbek, G.; Ali, M.H.; Shehab, E. Integrating Machine Learning Model and Digital Twin System for Additive Manufacturing. *IEEE Access* **2023**, *11*, 71113–71126. [CrossRef]
20. Yao, X.; Moon, S.K.; Bi, G. A hybrid machine learning approach for additive manufacturing design feature recommendation. *Rapid Prototyp. J.* **2017**, *23*, 983–997. [CrossRef]
21. Botella, R.; Fernández-Catalá, J.; Cao, W. Experimental Ni<sub>3</sub>TeO<sub>6</sub> synthesis condition exploration accelerated by active learning. *Mater. Lett.* **2023**, *352*, 135070. [CrossRef]
22. Davletshin, A.; Korznikova, E.A.; Kistanov, A.A. Machine Learning Prediction of the Corrosion Rate of Zinc-Based Alloys Containing Copper, Lithium, Magnesium, and Silver. *J. Phys. Chem. Lett.* **2025**, *16*, 114–122. [CrossRef]
23. Suwanpreecha, C.; Manonukul, A. A Review on Material Extrusion Additive Manufacturing of Metal and How It Compares with Metal Injection Moulding. *Metals* **2022**, *12*, 429. [CrossRef]
24. Herzog, D.; Asami, K.; Scholl, C.; Ohle, C.; Emmelmann, C.; Sharma, A.; Markovic, N.; Harris, A. Design guidelines for laser powder bed fusion in Inconel 718. *J. Laser Appl.* **2022**, *34*, 012015. [CrossRef]
25. Asami, M.K.; Herzog, D.; Bossen, B.; Geyer, L.; Klemp, C.; Emmelmann, C. Design Guidelines for Green Parts Manufactured with Stainless Steel in the Filament Based Material Extrusion Process for Metals (MEX/M). In Proceedings of the World PM2022 Congress & Exhibition, Lyon, France, 9–13 October 2022. [CrossRef]
26. Asami, K.; Crego Lozares, J.M.; Ullah, A.; Bossen, B.; Clague, L.; Emmelmann, C. Material extrusion of metals: Enabling multi-material alloys in additive manufacturing. *Mater. Today Commun.* **2024**, *38*, 107889. [CrossRef]
27. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.
28. Asami, M.K.; Kuehne, M. Machine Learning Models and Data for the Application of Machine Learning in Predicting Quality Parameters in Metal Material Extrusion (MEX/M). Available online: <https://tore.tuhh.de/entities/product/c0c59885-cf8e-4191-acd1-4d0097b5a57c> (accessed on 26 April 2025).
29. Othman, W.; Hamoud, B.; Kashevnik, A.; Shilov, N.; Ali, A. A Machine Learning-Based Correlation Analysis between Driver Behaviour and Vital Signs: Approach and Case Study. *Sensors* **2023**, *23*, 7387. [CrossRef]

30. Huang, J.; Wei, Y. Data-driven analysis of stroke-related factors and diagnostic prediction. In Proceedings of the 2024 IEEE 6th Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC), Chongqing, China, 24–26 May 2024; IEEE: Piscataway, NJ, USA, 2024; pp. 60–65.
31. Rezazadeh, A. Toe-Heal-Air-Injection Thermal Recovery Production Prediction and Modelling Using Quadratic Poisson Polynomial Regression. *arXiv* **2020**, arXiv:2012.02262.
32. Géron, A. *Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*, 2nd ed.; O'Reilly: Beijing, China; Boston, MA, USA; Farnham, UK; Sebastopol, CA, USA; Tokyo, Japan, 2019; ISBN 978-1-4920-3264-9.
33. Xiao, W.; Wang, Y. The Application of Partially Functional Linear Regression Model in Health Science. *Sci. Discov.* **2020**, *8*, 134. [[CrossRef](#)]
34. Roy, M.-H.; Larocque, D. Robustness of random forests for regression. *J. Nonparametric Stat.* **2012**, *24*, 993–1006. [[CrossRef](#)]
35. Adankon, M.M.; Cheriet, M. Support Vector Machine. In *Encyclopedia of Biometrics*; Li, S.Z., Jain, A.K., Eds.; Springer: Boston, MA, USA, 2015; pp. 1504–1511. ISBN 978-1-4899-7487-7.
36. Rachdi, M.; Laksaci, A.; Kaid, Z.; Benchiha, A.; Al-Awadhi, F.A.  $k$ -Nearest neighbors local linear regression for functional and missing data at random. *Stat. Neerl.* **2021**, *75*, 42–65. [[CrossRef](#)]
37. Kramer, O. Dimensionality Reduction by Unsupervised  $k$ -Nearest Neighbor Regression. In Proceedings of the 2011 10th International Conference on Machine Learning and Applications and Workshops, Honolulu, HI, USA, 18–21 December 2011; IEEE: Piscataway, NJ, USA, 2011; pp. 275–278.
38. Multilayer Perceptron. In *Pattern Recognition and Image Preprocessing*; Bow, S.-T., Ed.; Signal Processing and Communications; CRC Press: Boca Raton, FL, USA, 2002; pp. 201–224. ISBN 978-0-8247-0659-3.
39. Taud, H.; Mas, J.F. Multilayer Perceptron (MLP). In *Geomatic Approaches for Modeling Land Change Scenarios*; Camacho Olmedo, M.T., Paegelow, M., Mas, J.-F., Escobar, F., Eds.; Lecture Notes in Geoinformation and Cartography; Springer International Publishing: Cham, Switzerland, 2018; pp. 451–455. ISBN 978-3-319-60800-6.
40. Hassan, H.A.; Abd Ghani, M.N.A.; Zabidi, A.; Md Adnan, W.N.W.; Rahman, J.A.; Rusni, I.M. Pattern Classification in Recognising Idgham Maal Ghunnah Pronunciation Using Multilayer Perceptrons. In Proceedings of the 2024 IEEE 6th Symposium on Computers & Informatics (ISCI), Kuala Lumpur, Malaysia, 10 August 2024; IEEE: Piscataway, NJ, USA, 2024; pp. 333–338.
41. Meng, L.; McWilliams, B.; Jarosinski, W.; Park, H.-Y.; Jung, Y.-G.; Lee, J.; Zhang, J. Machine Learning in Additive Manufacturing: A Review. *JOM* **2020**, *72*, 2363–2377. [[CrossRef](#)]
42. Altman, N.S. An Introduction to Kernel and Nearest-Neighbor Nonparametric Regression. *Am. Stat.* **1992**, *46*, 175–185. [[CrossRef](#)]
43. Le, T.M.; Vo, T.M.; Pham, T.N.; Dao, S.V.T. A Novel Wrapper-Based Feature Selection for Early Diabetes Prediction Enhanced With a Metaheuristic. *IEEE Access* **2021**, *9*, 7869–7884. [[CrossRef](#)]
44. Breiman, L. Random Forests. *Mach. Learn.* **2001**, *45*, 5–32. [[CrossRef](#)]
45. Vapnik, V.N. An overview of statistical learning theory. *IEEE Trans. Neural Netw.* **1999**, *10*, 988–999. [[CrossRef](#)]
46. Bisong, E. The Multilayer Perceptron (MLP). In *Building Machine Learning and Deep Learning Models on Google Cloud Platform*; Apress: Berkeley, CA, USA, 2019; pp. 401–405. ISBN 978-1-4842-4469-2.
47. Chicco, D.; Warrens, M.J.; Jurman, G. The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation. *PeerJ Comput. Sci.* **2021**, *7*, e623. [[CrossRef](#)] [[PubMed](#)]
48. Klusowski, J.M. Sharp Analysis of a Simple Model for Random Forests. *arXiv* **2020**, arXiv:1805.02587. [[CrossRef](#)]
49. Battineni, G.; Chintalapudi, N.; Amenta, F. Machine learning in medicine: Performance calculation of dementia prediction by support vector machines (SVM). *Inform. Med. Unlocked* **2019**, *16*, 100200. [[CrossRef](#)]
50. Toorandaz, S.; Taherkhani, K.; Liravi, F.; Toyserkani, E. A novel machine learning-based approach for in-situ surface roughness prediction in laser powder-bed fusion. *Addit. Manuf.* **2024**, *91*, 104354. [[CrossRef](#)]
51. Caminero, M.Á.; Romero Gutiérrez, A.; Chacón, J.M.; García-Plaza, E.; Núñez, P.J. Effects of fused filament fabrication parameters on the manufacturing of 316L stainless-steel components: Geometric and mechanical properties. *Rapid Prototyp. J.* **2022**, *28*, 2004–2026. [[CrossRef](#)]
52. Gloeckle, C.; Konkol, T.; Jacobs, O.; Limberg, W.; Ebel, T.; Handge, U.A. Processing of Highly Filled Polymer–Metal Feedstocks for Fused Filament Fabrication and the Production of Metallic Implants. *Materials* **2020**, *13*, 4413. [[CrossRef](#)]
53. Damon, J.; Dietrich, S.; Gorantla, S.; Popp, U.; Okolo, B.; Schulze, V. Process porosity and mechanical performance of fused filament fabricated 316L stainless steel. *Rapid Prototyp. J.* **2019**, *25*, 1319–1327. [[CrossRef](#)]
54. Siamidoudaran, M.; İşciöğlü, E. Injury Severity Prediction of Traffic Collision by Applying a Series of Neural Networks: The City of London Case Study. *PROMET—TrafficTransportation* **2019**, *31*, 643–654. [[CrossRef](#)]
55. Eyercioğlu, Ö.; Aladağ, M. Non-Planar Toolpath for Large Scale Additive Manufacturing. *Int. J. 3D Print. Technol. Digit. Ind.* **2021**, *5*, 477–487. [[CrossRef](#)]

56. Rajput, A.S.; Babu, P.; Das, M.; Kapil, S. Influence of toolpath strategies during laser polishing on additively manufactured biomaterials. *Surf. Eng.* **2024**, *40*, 967–982. [[CrossRef](#)]
57. Asami, K.; Herzog, D.; Deutschmann, T.; Röver, T.; Kelbassa, I.; Emmelmann, C. Methodology for Cost Estimation Using Characteristic Factors in Additive Manufacturing. *J. Jpn. Soc. Powder Powder Metall.* **2025**, *72*, S75–S82. [[CrossRef](#)]

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