

Deliverable D3.4

Report on Defect Detection and Classification Algorithm for PBF-LB

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them.



Action number: #101138859

Action Acronym: DILAPRO
Action Name: Digital Laser Production

Deliverable Due date: 30.06.2025
Deliverable hand-in date: 30.06.2025

Revision	Date	Author/Organisation	Description
1	24/05/2025	Medad Monu/DCU	Initial draft
2	29/05/2025	Medad Monu/DCU	initial model results update
3	02/06/2025	Medad Monu/DCU	New best model update
4	04/06/2025	Medad Monu/DCU	latest best model update
5	10/06/2025	DQCC	Feedback
6	27/06/2025	DCU	Last revision

Table of contents

Executive Summary **4**

List of Figures **5**

Abbreviations **5**

1. Introduction **6**

2. Defect Detection Approach **7**

2.1 Relative Density as Defect Indicator 7

2.2 Defect Probability from Relative Density - Sigmoid Transformation 9

2.3 Defect Probability Implementation in *Deliverable D3.4* 10

2.4 Dataset, Model Training, Evaluation, and Selection Pipeline 12

2.5 Model Selection Criteria 14

3. Initial Findings **14**

3.1 Model exploration 14

3.2 Model Performance Improvement 15

4. Integration into DILAFACT **17**

5. Conclusion **18**

6. References **18**

7. Appendix 1: Tunned Model Performance **21**

Executive Summary

Process variability in PBF-LB often leads to internal defects that can compromise mechanical integrity. Ensuring part quality—especially for safety-critical applications—requires robust monitoring and control mechanisms. This *Deliverable D3.4* builds on the algorithm development framework for Real-Time Feedback Control (RTFC) introduced in *Deliverable D3.3* by proposing and adopting a data-driven defect detection approach based on sigmoid-transformed relative density. Relative density serves as a proxy for internal quality, and the sigmoid transformation maps continuous density predictions into probabilistic defect likelihoods, enabling both soft classification and binary decision-making (e.g., red/green light control in RTFC systems).

As part of this methodology, we retrained the machine learning models from *Deliverable D3.3* using an expanded dataset that includes both novel process conditions and defect-prone builds. A 98% relative density threshold is applied within the sigmoid transformation, aligning with quality specifications defined in use-case definitions presented in *Deliverable D1.1*. This enables probabilistic defect prediction at the layer-level, whilst supporting scalable decision-making for part-level qualification based on aggregated defect trends in the fully built part/component.

Although the final trained and validated model are still underway, the framework presented in this *Deliverable D3.4* document provides a practical and interpretable foundation for integrating defect classification into DILAFACT, real-time control, and digital certification workflows.

List of Figures

Fig. 1. Schematic illustration of the Defect detection approach. Note: IR-data refers to infra-red emissions from melt pools, captured using in-situ monitoring pyrometers. 8
Fig. 2. Three diagrams illustrating how the sigmoid-based defect probability changes to threshold $\boldsymbol{x0}$, Steepness $\boldsymbol{k}$, and example of Relative Density points illustrating how different relative densities translate into specific defect probabilities under fixed $\boldsymbol{x0}$ and $\boldsymbol{k}$. ©Authors 9
Fig. 3. Raw relative density of all fabricated samples, with a threshold line indicating 98%..... 11
Fig. 4. Defect probability as a function of relative density of all fabricated samples, computed using Equation [2] and experimentally acquired relative density data..... 12
Fig. 5. Performance of 13 out of 18 regression algorithms evaluated on the combined IR and PBF-LB datasets. Polynomial Regression, Gaussian Process Regressor, Kernel Ridge, Elastic Net, and Theil-Sen Regressor are excluded from the chart due to their exceptionally poor predictive performance, with mean cross-validation R^2 scores ranging from –3.5 to –95.02. Bars without visible error bars represent models with zero standard deviation (i.e., ±0.000), indicating consistent performance across folds. 15
Fig. 6. Performance of four (4) tunned regression algorithms using GridSearchCV and a set of 15 features. 16

Abbreviations

AE	Acoustic Emission
AM	Additive Manufacturing
API	Application Programming Interface.
IR	Infra-red
ML	Machine Learning
OEM	Original Equipment Manufacturer
PBF-LB	Powder Bed Fusion using a Laser Beam
RFECV	Recursive Feature Elimination with Cross Validation
RTFC	Real-Time Feedback Control

1. Introduction

Powder Bed Fusion – Laser Beam (PBF-LB) is a sophisticated additive manufacturing (AM) technique used to fabricate complex metal components by selectively melting successive layers of metal powder using a high-precision laser. Renowned for its ability to produce intricate geometries and high-performance parts, PBF-LB is widely adopted in sectors such as aerospace and automotive, where strength-to-weight ratio and part accuracy are critical.

Despite its advantages, PBF-LB presents several key challenges, including:

1. Material and process complexity
2. Powder quality and handling
3. Process parameter optimization
4. Production planning and control

The latter two (process parameter optimization and production control) are central concerns of the DILAPRO project, which aims to improve production reliability and part quality through advanced process analytics and process control strategies. In this context, machine learning (ML) techniques offer promising solutions for managing these challenges by enabling predictive modeling, real-time monitoring, and adaptive process adjustments [1], [2].

To facilitate advanced process analytics and process control strategies, a robust methodology for defect detection and classification is essential for ensuring part quality and functional reliability. Defects such as porosity, lack of fusion, or keyholing can significantly compromise the mechanical strength, fatigue life, and safety of printed components—particularly in critical applications. Early and accurate identification of defects enables timely corrective actions, reduces material waste, and supports certification processes. Leveraging machine learning models for defect classification, especially those informed by *in-situ* process data can provide a scalable, automated approach to quality assurance. When trained on quality indicators like relative density, such models can transform continuous outputs into probabilistic defect classifications, thus bridging the gap between process monitoring and actionable decision-making [3].

Relative density is a key indicator of part quality in metal additive manufacturing (AM), as established in *Deliverables D1.1*[4], *D2.1*[5], *and D3.1,* and the digital part certification framework in *Deliverable D1.4*[6]. It directly correlates with mechanical integrity, structural performance, and functional reliability, making it an essential metric for assessing compliance with industry standards. Parts with high relative density exhibit minimal internal defects—such as porosity and voids—thereby ensuring greater strength, fatigue resistance, and durability.

Given its critical role, *Deliverable D3.4* employs relative density as a proxy for defect classification. To enable flexible and interpretable defect prediction, a sigmoid transformation is applied to predicted relative density values. This maps continuous density estimates into probabilistic defect likelihoods, allowing for scalable, soft classification of defects rather than relying on rigid thresholds.

Building on the methodology introduced in *Deliverable D3.3*[7], *Deliverable D3.4* adopts an integrated framework/methodology for a defect detection and classification algorithm. This framework is designed to support:

a. Real-time feedback control (RTFC) during layer-wise PBF-LB manufacturing, and
b. Post-build digital part qualification, enabling confidence-driven certification of the entire component.

2. Defect Detection Approach

This *Deliverable D3.4* document presents a defect detection strategy based on a sigmoid transformation of relative density values. The approach transforms continuous regression outputs—specifically, machine learning model predictions of layer-wise relative density—into probabilistic indicators of defect presence. The method is designed to support real-time quality assessment and adaptive control in PBF-LB systems, such as in digital twins or feedback-driven manufacturing platforms. This is central to the goals of *Task 3.5* within the DILAPRO project. A schematic illustration of the defect detection approach is shown in **Fig. 1**.

2.1 Relative Density as Defect Indicator

Relative density is widely recognized as a key quality metric in metal Powder Bed Fusion – Laser Beam (PBF-LB) additive manufacturing. It quantifies the proportion of solid material within a printed part, with lower values typically indicating the presence of internal voids or discontinuities. Such imperfections—stemming from phenomena like gas entrapment, lack of fusion between powder layers, or keyholing due to excessive energy input—can significantly degrade the mechanical properties of a component, including strength, fatigue life, and ductility. Further details on AM defects have been fully covered in *Deliverable D3.1*[9] and in published review articles [10]–[12].

Due to this strong correlation, relative density serves as an effective indirect indicator of build quality and defectiveness. However, relying solely on a rigid threshold (e.g., ≥99% = acceptable, <99% = defective) can be overly simplistic and may not capture the nuanced effects of the inherently stochastic PBF-LB process, process variability, measurement uncertainty, or application-specific tolerance requirements.

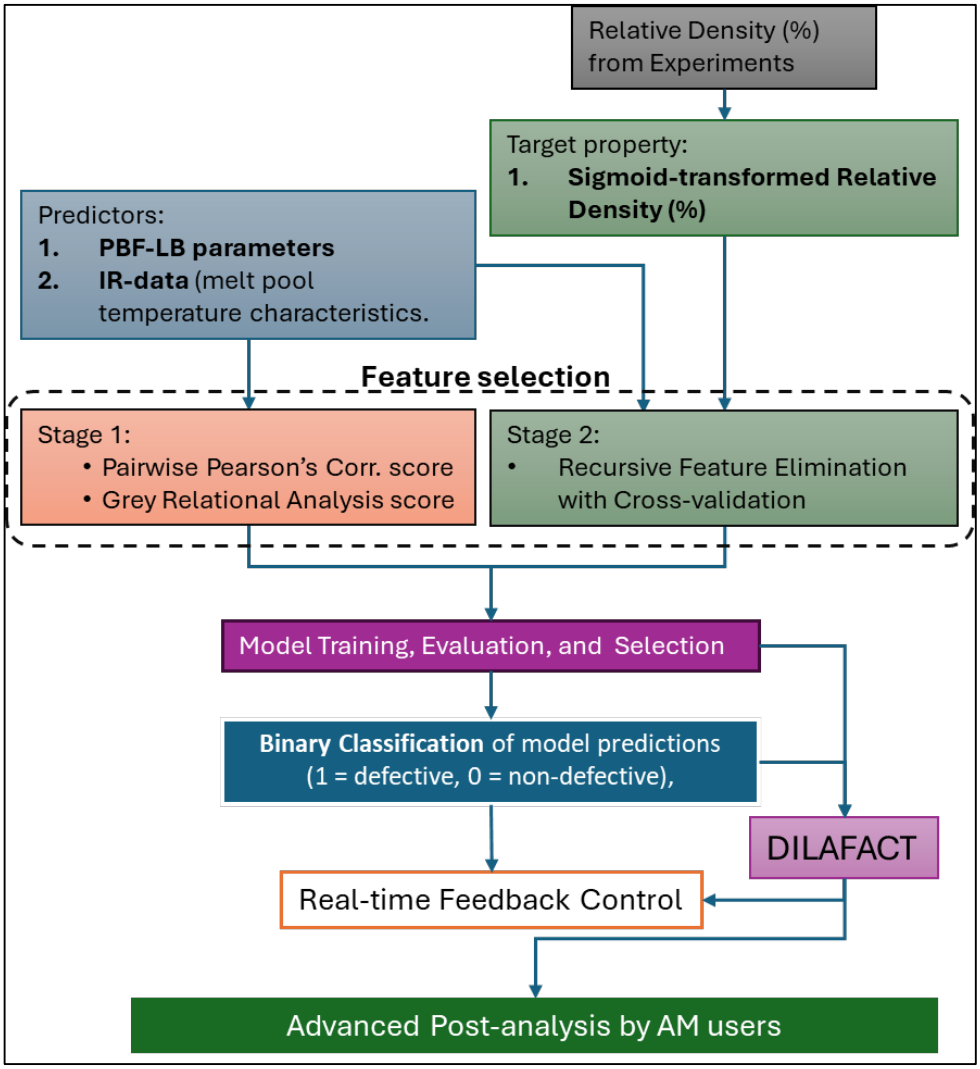


Fig. 1. Schematic illustration of the Defect detection approach. Note: IR-data refers to infra-red emissions from melt pools, captured using in-situ monitoring pyrometers.

In this *Deliverable D3.4*, we instead adopt a probabilistic mapping of relative density to defect likelihood using a sigmoid transformation. This approach supports soft classification, wherein each density value is associated with a defect probability rather than a binary label. This probabilistic framing offers several advantages:

1. It improves decision-making under uncertainty, especially for borderline cases near the quality threshold where absolute thresholds may result in over- or under-classification.
2. It provides interpretable, risk-aware outputs that can be tuned based on application criticality.
3. It enables smoother integration into feedback control systems or digital twins for real-time quality monitoring and adaptive process control.

Overall, this method allows for more flexible, informative, and robust quality classification within the digital part certification framework established across the DILAPRO project.

2.2 Defect Probability from Relative Density - Sigmoid Transformation

The probability that a given layer is defective, denoted as $P_{(defect)}$, is defined using the standard form of a sigmoid function [13]:

$$P_{(defect)} = \frac{1}{1+e^{-k(x-x_0)}} \quad [1]$$

where: $\boldsymbol{x}$ is the predicted relative density (%), and $\boldsymbol{x_0}$ is the defect threshold around which the transition from defect to no defect occurs. Specifically, $\boldsymbol{x_0}$ is the center or threshold point relative density value at which the probability of defect is 0.5. $\boldsymbol{k}$ is a tuneable steepness parameter controlling how sharp the transition is, and $\boldsymbol{e}$ is Euler's number (~ 2.718). $P_{(defect)} \in (0, 1)$ gives the likelihood that a layer is defective.

By tuning $\boldsymbol{k}$ and $\boldsymbol{x_0}$, the sigmoid function can mimic domain-specific expectations about when a part should be considered defective based on its density. For instance, a high $\boldsymbol{k}$ value makes the function resemble a step function (i.e., more like a hard threshold), suitable for stringent component quality assurance for critical aerospace parts. Conversely, a lower $\boldsymbol{k}$ value yields a more gradual transition (smoother curve). This is useful in cases with measurement uncertainty or applications with more tolerance in part quality. This is illustrated in **Fig. 2**.

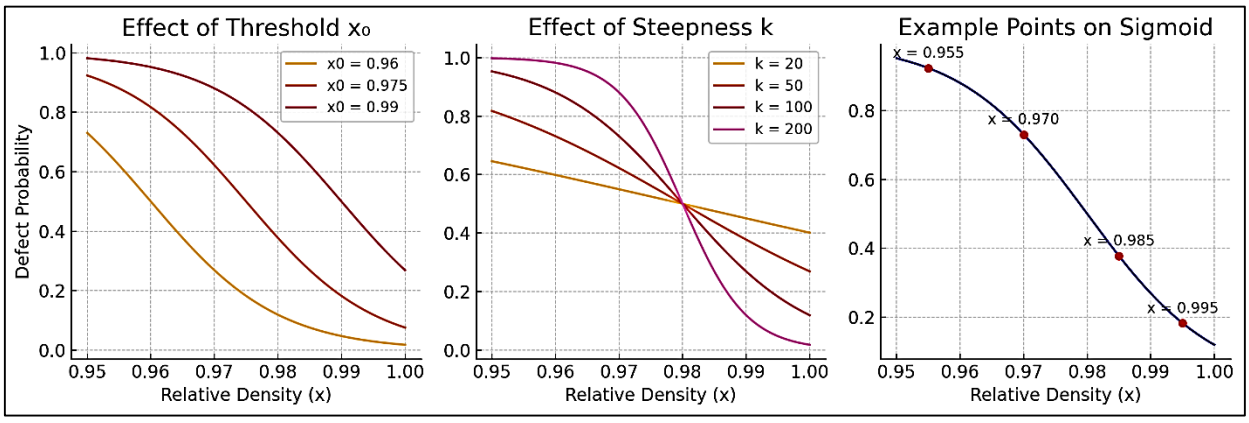


Fig. 2. Three diagrams illustrating how the sigmoid-based defect probability changes to threshold $\boldsymbol{x_0}$, Steepness $\boldsymbol{k}$, and example of Relative Density points illustrating how different relative densities translate into specific defect probabilities under fixed $\boldsymbol{x_0}$ and $\boldsymbol{k}$. ©Authors

In the context of defect detection using relative density as the input $\boldsymbol{x}$, the probabilistic output of the sigmoid function enables interpretability and flexibility as follows:

1. At low relative density values ($x \ll x_0$), $P_{(defect)} \rightarrow 1$: The built layer/part is likely to be defective since low density indicates a higher risk of porosity, lack of fusion, or other internal flaws.
2. At high relative density values ($x \gg x_0$), $P_{(defect)} \rightarrow 0$: The built layer/part is likely not defective, as high density suggests minimal internal defects.
3. At the threshold values ($x = x_0$), $P_{(defect)} = 0.5$: The probability of defect is 50%, marking the point of maximum uncertainty or the model's decision boundary. Essentially, the probability of the part being defective or non-defective is equal (50%).

In summary, predicted sigmoid transformed relative density values near 1 suggest a high risk of defect, and values near 0 indicate high confidence in part quality.

2.3 Defect Probability Implementation in *Deliverable D3.4*

In this deliverable, we adopt a threshold of $\boldsymbol{x_0}$ = 98% relative density per the use-case specifications defined in *Deliverable D1.1*[4]. The steepness parameter, $\boldsymbol{k}$ was set at -1, leading to the following revised form of the standard sigmoid transformation function:

$$P_{(defect)} = \frac{1}{1+e^{(x-98)}} \quad [2]$$

This formulation means that components with relative density significantly below 98% are increasingly likely to be defective, while those above this threshold are considered more reliable. To support this decision-making logic, **Fig. 3** presents the raw experimentally measured relative density values, and **Fig. 4** shows the corresponding sigmoid-transformed defect probabilities. The transformation makes it easier to visually distinguish between acceptable and defective builds in RTFC senarios, howbeit, with slightly increased complexity in model training.

Once a machine learning model has been trained to predict the sigmoid-transformed defect probability, the output value (ranging from 0 to 1) can be interpreted as a confidence score. For operational decision-making, a binary threshold is applied at $P_{(defect)} = 0.5$:

$$\hat{y}_{binary} = \begin{cases} 1 \ if \ \frac{1}{1+e^{(x-98)}} \geq \tau \\ 0 \ otherwise \end{cases} \quad [3]$$

Where $\hat{y}_{binary}$ is the binary classification outcome (1 = defective, 0 = non-defective), and τ is the classification threshold, 0.5. The simple interpretation is as follows:

- If $P_{(defect)} > 0.5$, the layer or component is classified as **defective (1)**.
- If $P_{(defect)} < 0.5$, the layer or component is considered **non-defective (0)**.

This binary classification supports real-time feedback control mechanisms in PBF-LB systems. For example:

- A $P_{(defect)} \geq 0.5$ can trigger a **red light warning/alert**, indicating the need for immediate corrective action or layer reprocessing.
- A $P_{(defect)} < 0.5$ can trigger a **green light**, allowing the build to proceed without interruption.

By mapping continuous density predictions to intuitive defect probabilities, the system enables both machine-readable control logic and human-interpretable quality assurance**,** contributing to smarter, closed-loop process control strategies.

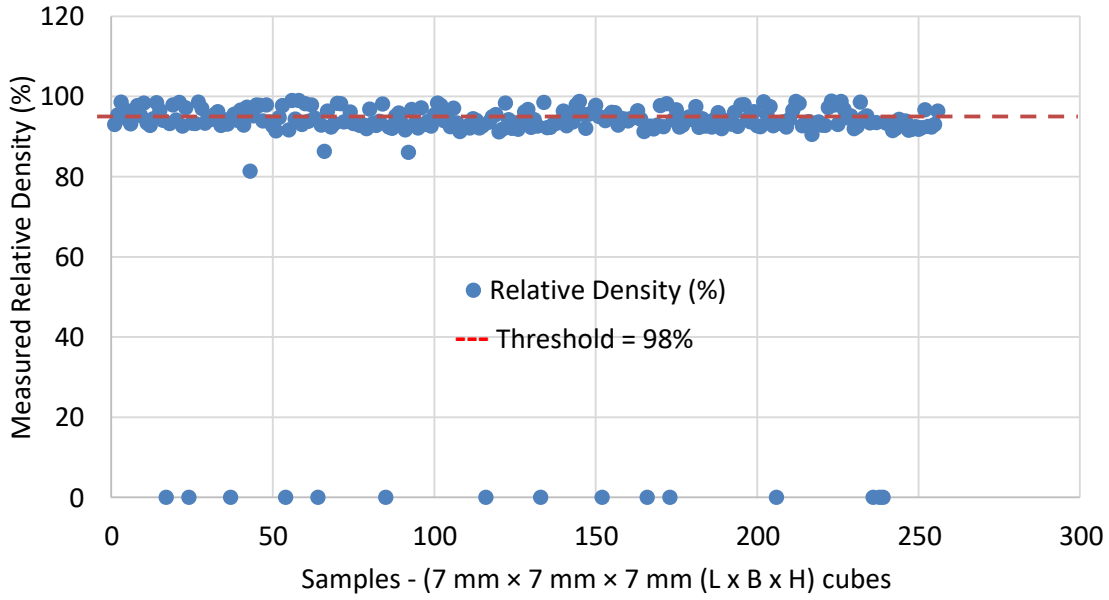


Fig. 3. Raw relative density of all fabricated samples, with a threshold line indicating 98%.

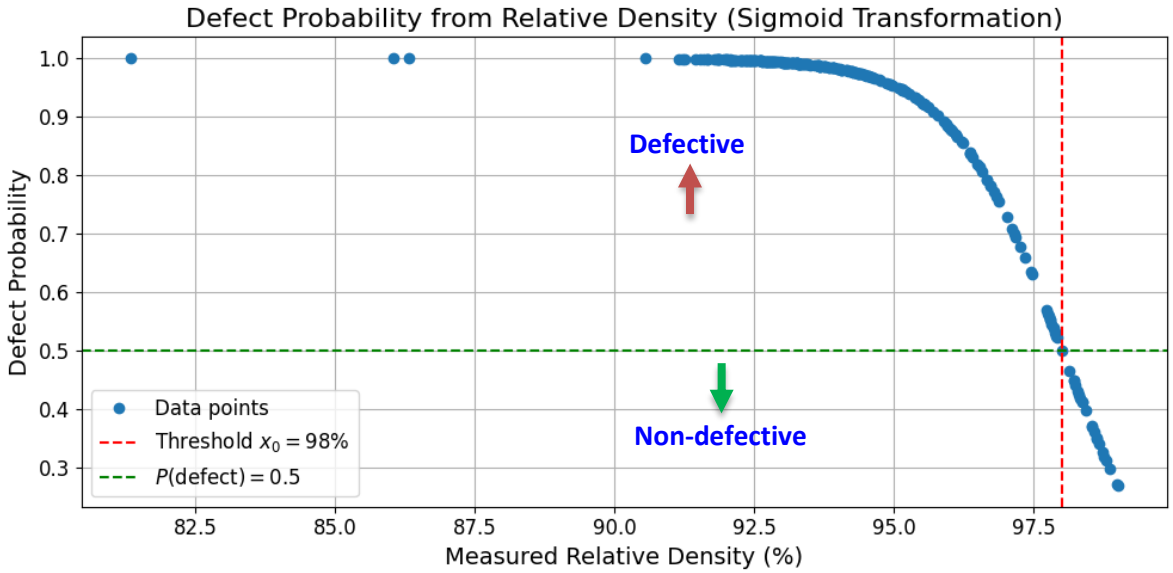


Fig. 4. Defect probability as a function of relative density of all fabricated samples, computed using Equation [2] and experimentally acquired relative density data**.**

2.4 Dataset, Model Training, Evaluation, and Selection Pipeline

To support and validate the proposed classification framework, we expanded the dataset used in *Deliverable D3.3*[7] and re-evaluated the performance of 18 machine learning models. The expanded dataset includes:

1. 24 new parts fabricated using previously unseen process parameter combinations, aimed at testing the models' ability to generalize to novel conditions,
2. 15 parts intentionally produced with known defect-prone parameter sets, used to evaluate model sensitivity to defect-related patterns.

These additions improve the diversity and representativeness of the dataset, especially for edge cases, and provide a more rigorous basis for evaluating the robustness and reliability of the predictive models.

As established in *Deliverable D3.3*[7], the optimal combination of data fusion strategy included:

1. PBF-LB process parameters (laser power, scan speed, laser spot size, hatch spacing, VED),
2. Melt pool temperature characteristics extracted from high-speed infrared pyrometry.

Only these selected features were used in *Deliverable D3.4*, in alignment with prior validation results from sensor fusion and feature selection studies detailed in **Section 3** (Pipeline for RTFC ML Model Selection and Optimisation) in the *Deliverable D3.3*[7] document.

Given that the sigmoid transformation of relative density produces a continuous probability value within the range (0, 1), the underlying machine learning task remains a regression problem. Accordingly, a modified model evaluation algorithm was adopted, as illustrated in **Object A**. This revised algorithm builds upon and adapts **Object B** from Section 3 (Pipeline for RTFC ML Model Selection and Optimisation) of the *Deliverable D3.3*[7] document:

Object A: Model Training Algorithm on Sigmoid-Transformed Relative Density

Algorithm:	Model Training, Evaluation, and SHAP Analysis
Preprocessing:	**Probability mapping of Relative density using** $\boldsymbol{P(defect) = \frac{1}{1+e^{(x-98)}}}$ Multi-stage feature selection using GRA, PCC, and RFECV (refer to **Object B** in *Deliverable D3.3*[7] *document*).
Require:	Top selected features: [*X_filtered[top_features*], Train-test split datasets: [*X_train, X_test, y_train, y_test*], Algorithms to evaluate: *models{}* (A dictionary with names of eighteen (18) regression-based ML algorithms instantiated through models
Ensure:	Evaluation metrics data frame: [metrics_df] Saved models and associated metadata: such as feature names and order

1. Create an empty dictionary, *results{ }*, to store evaluation metrics for each model.
2. For each *name, model* pair in *models{}*, **do**
3. Train (fit) the model using *X_train*
4. Save the trained model and feature order to a file.
5. For each model, ensure feature names are saved for future alignment.
6. Reload the saved feature names and order.
7. Align *X_test* column data to match the training phase (*X_train*) feature order.
8. Record the start time.
9. Predict *y_test* using the aligned *X_test* features and in the same order.
10. Calculate and store the model prediction time.
11. If required *name, model* pair, **do**
12. Optionally, compute Shapley Additive Explanations and associated plots.
13. **end for**
14. Perform 10-fold cross-validation on *X_train[top_features*]
15. Calculate the following metrics:
    - R^2 Score: Coefficient of determination.
    - MAE: Mean Absolute Error
    - MSE: Mean Squared Error
    - Cross-validation R^2 mean and standard deviation.
16. Store the metric values to *results{ }*
17. Generate training and performance plots.
18. **end for**
19. Convert *results{ }* into a transposed DataFrame, *metrics_df*
20. Sort the DataFrame (*metrics_df*) by R^2 Score in descending order.
21. Return the sorted *metrics_df* containing evaluation metrics for all models.

The same suite of regression models introduced in *Deliverable D3.3*[7] is reused in this deliverable. Namely:

1. AdaBoost Regressor (ABR)
2. Bayesian Ridge (BR)
3. Decision Tree (DT)
4. Elastic Net (EN)
5. Extra Trees Regressor (ETR)
6. Gaussian Process Regressor (GPR)
7. Gradient Boosting Regressor (GBR)
8. Gradient Boosted Trees (XGBoost Regressor)
9. Huber Regressor (HR)
10. K-Nearest Neighbours (KNN)
11. Kernel Ridge (KR)
12. Partial Least Squares (PLS) Regression
13. Polynomial Regression (PR)
14. Random Forest (RF)
15. Random Sample Consensus (RANSAC) Regressor
16. Stochastic Gradient Descent (SGD)

17. Support Vector Machine (SVM)
18. Theil-Sen Regressor (TSR)

2.5 Model Selection Criteria

After all the models have been trained to predict the sigmoid-transformed relative density, the best-performing model is selected based on the criteria established in *Deliverable D3.3*[7]. In summary, the ideal machine learning algorithm will be selected based on its ability to meet or exceed the following performance criteria:

1. Predictive Accuracy based on R^2 score during exploratory training phases.
2. Consistency based on mean R^2 across multiple cross-validation folds.
3. Lower Bound of True Performance during cross-validation 95% confidence interval.
4. Performance Superiority in at least two of the following performance metrics:
    a. Mean Squared Error (MSE)
    b. Root Mean Squared Error (RMSE)
    c. Mean Absolute Error (MAE)
    d. Mean Absolute Percentage Error (MAPE)

3. Initial Findings

3.1 Model exploration

Initial results indicate encouraging training performance, with the baseline Gradient Boosting Regressor and Extra Trees Regressor achieving average R^2 scores of 0.81 ± 0.00 and 0.78 ± 0.01, respectively (see blue bars in **Fig. 5**). Although these scores are lower than the 0.95 R^2 previously achieved by the Extra Trees Regressor on raw (non-transformed) relative density data in *Deliverable D3.3*[7], they remain strong.

Notably, these current models—trained using a feature consisting of PBF-LB process parameters and infrared (IR) melt pool data—outperform those trained on alternative feature combinations, such as acoustic emission (AE) only, PBF-LB parameters only, or AE combined with PBF-LB, as documented in *Deliverable D3.3*[7].

In comparison to related work in the literature, our models also show competitive performance. For instance, Barrionuevo et al. [14] reported a best R^2 score of 0. 63 using a Gradient Boosting Regression model trained to predict the relative density of 316L stainless steel. While their objective differed from our current focus on defect probability classification, this benchmark reinforces the strength of our approach. In works specifically targeting sigmoid-transformed relative density for defect probability classification, our initial model performance is slightly below that of Wang et al. [15], who reported R^2

scores of 86.2% with Random Forest and 85.8% with XGBoost after hyperparameter optimization. Their models utilized the thermodynamic properties of the metal powder alongside PBF-LB process parameters.

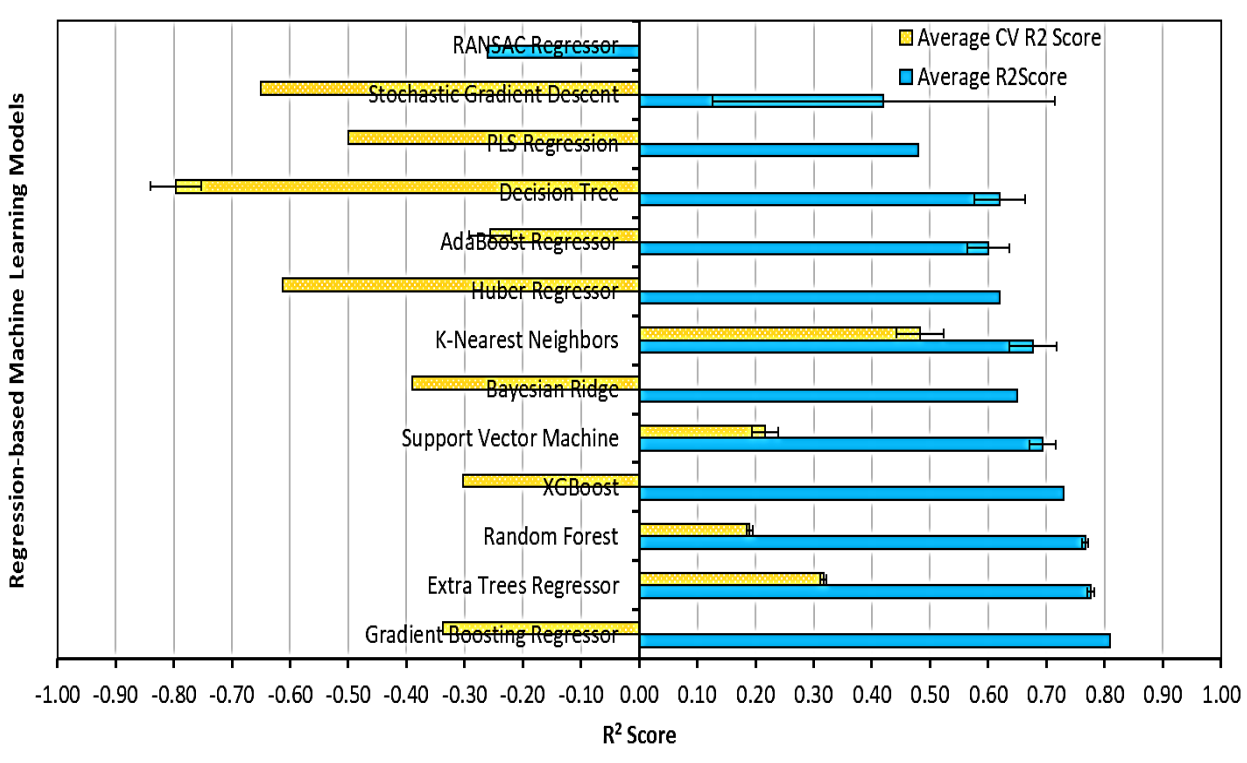


Fig. 5. Performance of 13 out of 18 regression algorithms evaluated on the combined IR and PBF-LB datasets. Polynomial Regression, Gaussian Process Regressor, Kernel Ridge, Elastic Net, and Theil-Sen Regressor are excluded from the chart due to their exceptionally poor predictive performance, with mean cross-validation R^2 scores ranging from –3.5 to –95.02. Bars without visible error bars represent models with zero standard deviation (i.e., ±0.000), indicating consistent performance across folds.

On the aspect of cross-validation metrics (yellow bars in **Fig. 5**), all initial model performances range from very poor to even negative in most cases. This is partly attributed to the sigmoid-transformed state of the target variable (i.e., relative density), which significantly compresses the range of target values—especially at the extremes. As a result, the transformed values become less sensitive to differences in the original relative density, making it harder for the model to learn meaningful patterns. This loss of variance and non-linearity in the transformed target reduces the model's ability to generalize, leading to poor CV performance.

3.2 Model Performance Improvement

Improving the overall baseline model performance necessitated a minor revision of the methodological approach described in *Deliverable D3.3*[7], specifically in the areas of feature engineering and model selection. In place of the initial strategy, Principal Component Analysis (PCA) and Recursive Feature Elimination with Cross-Validation (RFECV) were employed to enhance feature selection. Additionally,

hyperparameter tuning is being performed using GridSearchCV across ten (10) subsets of selected features, each containing between 8 and 21 variables. For each subset, four regression models selected from **Fig. 5**—ExtraTreesRegressor (ETR), XGBoost (XGB), K-Nearest Neighbors (KNN), and GradientBoostingRegressor (GBR)—are tuned in parallel using 50-fold cross-validation. Each model undergoes approximately 1,000 fits, amounting to a total of around 40,000 individual training and evaluation runs. While these training processes are automated on a workstation PC, the scale of computation still imposes a significant burden, demanding considerable processing time and computational resources to complete the optimization effectively.

Nevertheless, **Fig. 6** shows the current results of the tuned models using only 15 features, with remarkabely improved training and cross validation R^2 scores compared to **Fig. 5**. The features include: *Q1_Temp, Coefficient_of_Variation_Temp, Skewness_Temp, Kurtosis_Temp, Interquartile_Range_Temp, Percentile_10_Temp, Variance_Temp, Stdev_Temp,* ***Laser Power (W),*** *Mode_Temp, VEDf,* ***Scan Speed (mm/s),*** *Median_Temp, VEDh, Range_Temp.* unlike the feature list from which included only spot-size as the only acessible feature that can be changed only the fly during a pbf-lb buil process, the updated feature list include laser power and laser scan speed. he selected features include: Q1_Temp, Coefficient_of_Variation_Temp, Skewness_Temp, Kurtosis_Temp, Interquartile_Range_Temp, Percentile_10_Temp, Variance_Temp, Stdev_Temp, Laser Power (W), Mode_Temp, VEDf, Scan Speed (mm/s), Median_Temp, VEDh, and Range_Temp.

Unlike the earlier feature set described in *Deliverable D3.3*[7], which included only laser spot size as the single dynamically adjustable parameter during a PBF-LB build process, the updated feature set now incorporates two additional real-time tunable parameters: laser power and scan speed. This enhancement increases the potential for real-time control and optimization during the build.

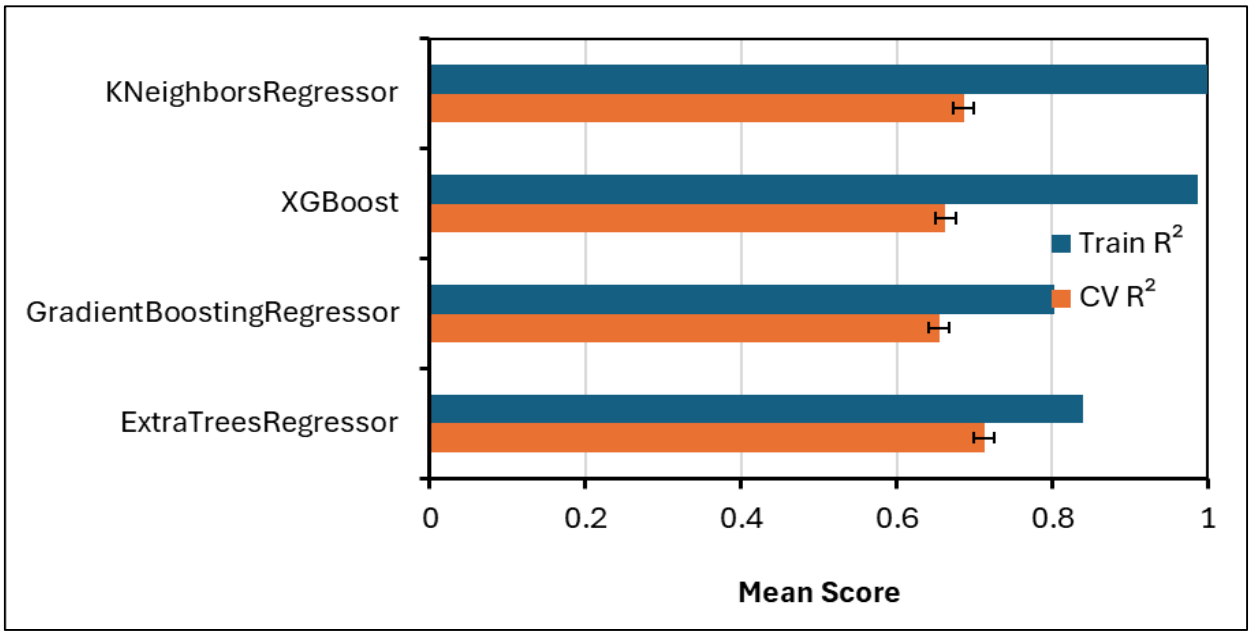


Fig. 6. Performance of four (4) tunned regression algorithms using GridSearchCV and a set of 15 features.

Regarding model performances, the mean training R^2 scores ranged from 0.8397 to 1. The best tunned model as of yet, is the **ExtraTressRegressor** model with a highest mean cross-validation R^2 score of 0.7126. this implies that it showed best generalization (mean CV R^2) and competitive test metrics (mean Test R^2: 0.7403, Test mean squared error: 0.0087). XGBoost had significant train-test gap (Train R^2: 0.9878 vs Test R^2: 0.7793) and KNeighborsRegressor were perfectly fit on training (Train R^2: 1.0000). However, these values are indicators of overfitting, therefore raising concerns of the credibility of their performances. Refer to **Appendix 7** for a tabular summary of the tunned model performance.

Currently, the ExtraTreesRegressor balances performance and generalization, even though its R^2 scores are below the desired value training and cross-validation targets suitable for RTFC, DILAFACT, and Digital certification for the Dilapro project. Therefore, the search for the best feature set and hypertunning to achieve those performance target is currently still ongoing.

An option would be to utilise deep learning and ANN models, however, the challenge is that the model parameters are not interpretable via shap analysis (explainable machine learning), and therefore make its challenging to know what exact parameters will need to be changed/tweaked by the RFTC system, considering that there is a finite set of parameters (laser power, speed, and spot size) that can only be changed on-the-fly during a typical PBF-LB build process.

4. Integration into DILAFACT

Given the constraints associated with implementing cross-OEM (Original Equipment Manufacturer) and cross-platform control systems within additive manufacturing (AM) environments—as well as the requirement for direct access to local machine resources—a locally hosted Python-based control logic is proposed. This script interfaces with the proprietary AconityMINI APIs and installed hardware, and while also capable of calling the pre-trained ML models hosted on the DILAFACT API.

This architecture promotes model transferability and scalability by allowing external users or systems to query centralized models via secure API calls to DILAFACT. Users only need to supply the required input features in the correct format to receive predictions, without needing access to the original training environment or local model files. This modular setup supports platform-independent integration across various PBF-LB machines, enabling consistent logic for real-time control.

Tangible outputs from this *Deliverable D3.4* work to be hosted on DILAFACT include the following files:

1. Pretrained model file,
2. Training Feature Names and Order file, and
3. Pretrained Scaler file

5. Conclusion

This *Deliverable D3.4* report presents the proposed approach and preliminary findings for a defect detection and classification algorithm for additively manufactured 316L stainless steel produced via PBF-LB. The methodology is based on a sigmoid transformation of relative density values, allowing continuous predictions to be interpreted as probabilistic defect likelihoods.

The work builds on the foundation established in *Deliverable D3.3*, particularly in terms of model training, evaluation, and selection using PBF-LB process parameters and melt pool temperature features extracted from in-situ infrared data.

Preliminary results demonstrate promising model performance but also highlight the need for further refinement to the limit of available computing resources.

6. References

[1] Z. Jin, Z. Zhang, K. Demir, and G. X. Gu, "Machine Learning for Advanced Additive Manufacturing," *Matter*, vol. 3, no. 5, pp. 1541–1556, 2020, doi: 10.1016/j.matt.2020.08.023.

[2] P. Akbari, M. Zamani, and A. Mostafaei, "Machine learning prediction of mechanical properties in metal additive manufacturing," *Addit. Manuf.*, vol. 91, no. March, p. 104320, 2024, doi: 10.1016/j.addma.2024.104320.

[3] R. McCann *et al.*, "In-situ sensing, process monitoring and machine control in Laser Powder Bed Fusion: A review," *Addit. Manuf.*, vol. 45, no. April, p. 102058, Sep. 2021, doi: 10.1016/j.addma.2021.102058.

[4] DILAPRO, "Deliverable D1.1: Use case definition and requirements including business case definitions," 2024.

[5] DILAPRO, *Deliverable D2.1: Multi-scale multiphysics simulation framework with hierarchical modeling workflows for selected material*. 2024.

[6] DILAPRO, "Deliverable D1.4: Certification analyses," 2024.

[7] DILAPRO, "Deliverable D3.3: Report on Real-time Feedback Control Algorithm." 2025.

[8] DILAPRO, "Deliverable D5.2: DEC Plan : Plan for Dissemination , Exploitation , and Communication," 2024.

[9] DILAPRO, "Deliverable D3.1: Analysis of sensors and identification of technical solutions," 2024.

[10] M. Grasso and B. M. Colosimo, "Process defects and in situ monitoring methods in metal powder bed fusion: a review," *Meas. Sci. Technol.*, vol. 28, no. 4, p. 044005, Apr. 2017, doi:

10.1088/1361-6501/aa5c4f.

[11] E. Malekipour and H. El-Mounayri, “Common defects and contributing parameters in powder bed fusion AM process and their classification for online monitoring and control: a review,” *Int. J. Adv. Manuf. Technol. 2017 951*, vol. 95, no. 1, pp. 527–550, Oct. 2017, doi: 10.1007/S00170-017-1172-6.

[12] H. Taheri, M. R. B. M. Shoaib, L. W. Koester, T. A. Bigelow, P. C. Collins, and L. J. Bond, “Powder-based additive manufacturing - a review of types of defects, generation mechanisms, detection, property evaluation and metrology,” *Int. J. Addit. Subtractive Mater. Manuf.*, vol. 1, no. 2, p. 172, 2017, doi: 10.1504/ijasmm.2017.088204.

[13] A. Alivanoglou and A. Likas, “Probabilistic models based on the Π-sigmoid distribution,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 5064 LNAI. pp. 36–43, 2008. doi: 10.1007/978-3-540-69939-2_4.

[14] G. O. Barrionuevo, J. A. Ramos-Grez, M. Walczak, and C. A. Betancourt, “Comparative evaluation of supervised machine learning algorithms in the prediction of the relative density of 316L stainless steel fabricated by selective laser melting,” *Int. J. Adv. Manuf. Technol.*, vol. 113, no. 1–2, pp. 419–433, Mar. 2021, doi: 10.1007/s00170-021-06596-4.

[15] J. Wang, S. G. Jeong, E. S. Kim, H. S. Kim, and B.-J. Lee, “Material-agnostic machine learning approach enables high relative density in powder bed fusion products,” *Nat. Commun.*, vol. 14, no. 1, p. 6557, Oct. 2023, doi: 10.1038/s41467-023-42319-x.

7. Appendix 1: Tunned Model Performance

Model	CV R^2	CV MAE	CV MSE	Train R^2	Train MAE	Train MSE	Test R^2	Test MAE	Test MSE	Time (s)
ExtraTreesRegressor	0.7126	0.0606	0.0117	0.8397	0.0418	0.0064	0.7403	0.0562	0.0087	30.93
GradientBoosting	0.6543	0.0725	0.0139	0.8032	0.055	0.0078	0.6975	0.066	0.0101	21.75
XGBoost	0.6628	0.0592	0.0134	0.9878	0.0129	0.0005	0.7793	0.0473	0.0074	8.58
KNeighborsRegressor	0.6862	0.0626	0.0124	1	0	0	0.71	0.0595	0.0097	0.32