Enhanced Prediction of Manufacturing Quality Ratings Using Optimized Stacking Ensemble Modeling

Lanbao Hou, Zhiqiong Zou*
School of Mathematics and Physics, Jingchu University of Technology, Jingmen 448000, Hubei, China
*Corresponding author
E-mail: 11095881@163.com

Keywords: manufacturing quality, machine learning (ML), stacking, SHAP analysis, material transformation metric (MTM)

Received: May 20, 2025

The predictive modeling of quality in manufacturing is vital in improving efficiency, cutting costs, and attaining zero-defect production. This paper addresses this requirement through an empirical comparison of machine learning models to predict manufacturing quality ratings. On a simulated dataset of 3,957 samples with five important features of the process, this study compared the performance of six basic models (AdaBoost, Bagging, Decision Tree, Gradient Boosting, K-Nearest Neighbors, and XGBoost) and two more advanced ensemble models (Averaging and Stacking). The findings demonstrated that the Stacking Ensemble model outperformed all other candidates, with higher performance in terms of an R^2 of 0.999 and the lowest values of error (MSE: 0.004, RMSE: 0.065, MAE: 0.016) in the test set. Moreover, SHAP analysis of the Stacking ensemble as the best model revealed that Material Transformation Metric (MTM) and Temperature (T) were the top features having a significant impact on the quality of products. The analysis finds that the Stacking Ensemble method provides a valid and very effective framework for predicting quality, which is helpful in the optimization of the manufacturing process.

Povzetek: Članek primerja več modelov strojnega učenja za napoved ocene kakovosti v proizvodnji in pokaže, da je ansambelski model stacking *najboljši (R^2≈0,999, najnižje napake), pri čemer sta po analizi SHAP najpomembnejša vplivna dejavnika MTM in temperatura.*

1 Introduction

Manufacturing has evolved significantly with the advent of the new sophisticated information systems into smart manufacturing. Quality of products in this extremely competitive environment will be a decisive factor in profitability, competitiveness and sustainability. On the other hand, low quality results in higher expenses, loss of reputation and customers. High-quality and zero-defect manufacturing production necessitates strong systems to anticipate and manage quality variables in advance.

In this regard, machine learning (ML) has become a potent solution to examine large volumes of manufacturing data to forecast quality results. ML models can compute the complex trends and relationships between the parameters of processes and final product quality, using historical data, thus making it possible to optimize and model manufacturing processes [1]. Proper quality prediction gives huge benefits throughout the supply chain, allowing correction beforehand and resource conservation [2].

The application of ML in this area has been extensively studied. In order to give a systematic description of the existing research environment, Table 1 describes the key related works, with their diversity in terms of dataset, methodology, and contribution.

Table 1: Summary of related works in manufacturing quality prediction using ML.

Reference	Dataset & Context	Methodology	Key Contributions	Main Findings
Li et al. [1]	Manufacturing quality data	Combined system: Particle Swarm Optimizer + Neural Networks	Proposed an intelligent prediction model and evaluation system based on big data.	The model's performance aligned with actual manufacturing outcomes, proving suitable for real-world construction activities.
Sankhye and Hu [2]	Product quality data	Not Specified (Focus on feature construction)	Emphasized the critical role of feature engineering and expert knowledge in training	Highlighted that prior quality understanding can save costs related to recalls, packaging, and transferring.

			models for quality prediction.	
Link et al. [3]	Manufacturing quality data	Integration of expert knowledge with ML models	Combined heuristic expert knowledge with data-driven ML to predict product quality.	Direct expert involvement saved time and aided model interpretation, while ML compensated for human iteration limits.
Weichert et al. [4]	Review article (Various manufacturing processes)	Review of ML and optimization methods	Provided a comprehensive review of ML for optimizing production processes.	Stressed the need to consider the correlation between data, algorithms, optimizers, and the specific quality problem.
Psarommatis and Azamfirei [5]	Conceptual/Review	Review and framework for Zero Defect Manufacturing (ZDM)	Provided a complete guide for advanced and sustainable quality management towards ZDM.	Positioned ZDM as a holistic approach requiring a comprehensive examination of the entire manufacturing system.
Msakni et al. [6]	Automotive product quality	Neural Networks, Random Forest, LSTM	Applied ML to forecast specific automotive product quality and detect tolerance violations.	Enabled process improvements by early detection of quality issues, preventing costly defects.
Yang et al. [7]	Manufacturing firm data	Dual Machine Learning Models	Investigated the relationship between ESG (Environmental, Social, Governance) ratings and digital technological innovation.	Showed that external factors like governance and society can influence manufacturing technology adoption.
Akbari et al. [8]	Spatter behavior in Laser Powder Bed Fusion (LPBF)	Six ML methods (Neural Networks outperformed others)	Predicted ejection velocity and spatter direction to minimize defects in additive manufacturing.	Neural Networks outperformed other models, significantly progressing defect identification and product quality improvement.
Azamfirei et al. [9]	Manufacturing quality inspection	Automation and in-line inspection systems	Applied automation for in-line quality inspection as a zero-defect manufacturing approach.	Demonstrated the practical application of automated systems to achieve high-quality production goals.
This Study	Simulated manufacturing process data (3,957 samples, 5 features: T, P, TxP, MFM, MTM)	**Comprehensive comparison of 8 models: AdaBoost, Bagging, DT, GB, KNN, XGBoost, Averaging, Stacking**	**Extensively compared multiple base and ensemble models; Identified Stacking as optimal; Used SHAP for granular feature importance analysis.**	**Stacking Ensemble achieved superior performance (R^2: 0.999, lowest errors); Quantified feature influence (MTM most critical, P least).**

Even as the literature indicates that ML has enormous potential in quality prediction, there still exist gaps. Most of the literature is based on a few models or localized specific manufacturing settings, which limits the generalizability of their results [2], [6]. Moreover, though ensemble methods are known to be powerful, more extensive comparisons of a broad range of base and ensemble models, namely with respect to the manufacturing quality rating, are less frequent. Additionally, model interpretability is also desired to understand which process parameters have the strongest effect on quality outcome, which is a feature that is essential to practical implementation. This study will address these gaps by conducting a general empirical comparison of different emerging ML models, including base learners and state-of-the-art ensemble models, in predicting manufacturing quality ratings. Furthermore, this study goes beyond model execution to offer practical information using explainable AI (XAI) methods, i.e., SHAP analysis, to prioritize the effect of essential manufacturing characteristics.

The ultimate objective of the paper is to construct, analyze and assess a predictive model of quality rating in manufacturing. The given goals are achieved by the following specific research questions:

1) Which of the following ML models (AdaBoost, Bagging, Decision Tree, Gradient Boosting, K-Nearest Neighbors, and XGBoost) proves to be the most accurate and efficient in forecasting manufacturing quality rating?
2) Can more sophisticated ensemble methods (Averaging and Stacking) increase prediction performance than the best single model?
3) How do the relative values of the major process parameters, Temperature (T), Pressure (P), their combination (TxP), Material Fusion Metric (MFM), and Material Transformation Metric (MTM) affect the predicted quality rating?

The study hypothesizes that the Stacking Ensemble model will outperform all the base models since it can be used to combine their strengths and thus give the most effective and reliable results in predicting manufacturing quality.

The remainder of the paper will consist of the following: Section 2 will contain the description of the methodology, including the dataset, computational environment and the metrics employed. Section 3 discusses the ML models, and hyperparameter optimization is then explained in Section 4. The results and findings are given in Section 5, and a discussion of the findings will be presented in Section 6. The study ends by giving a conclusion of the study in terms of summarizing the major findings of the study and proposing future research directions in Section 7.

2 Methodology

The article focuses on the performance of various machine learning architectures to estimate manufacturing quality scores. The overall methodology is the data description, preprocessing, model selection, validation, and evaluation illustrated in Fig. 1.

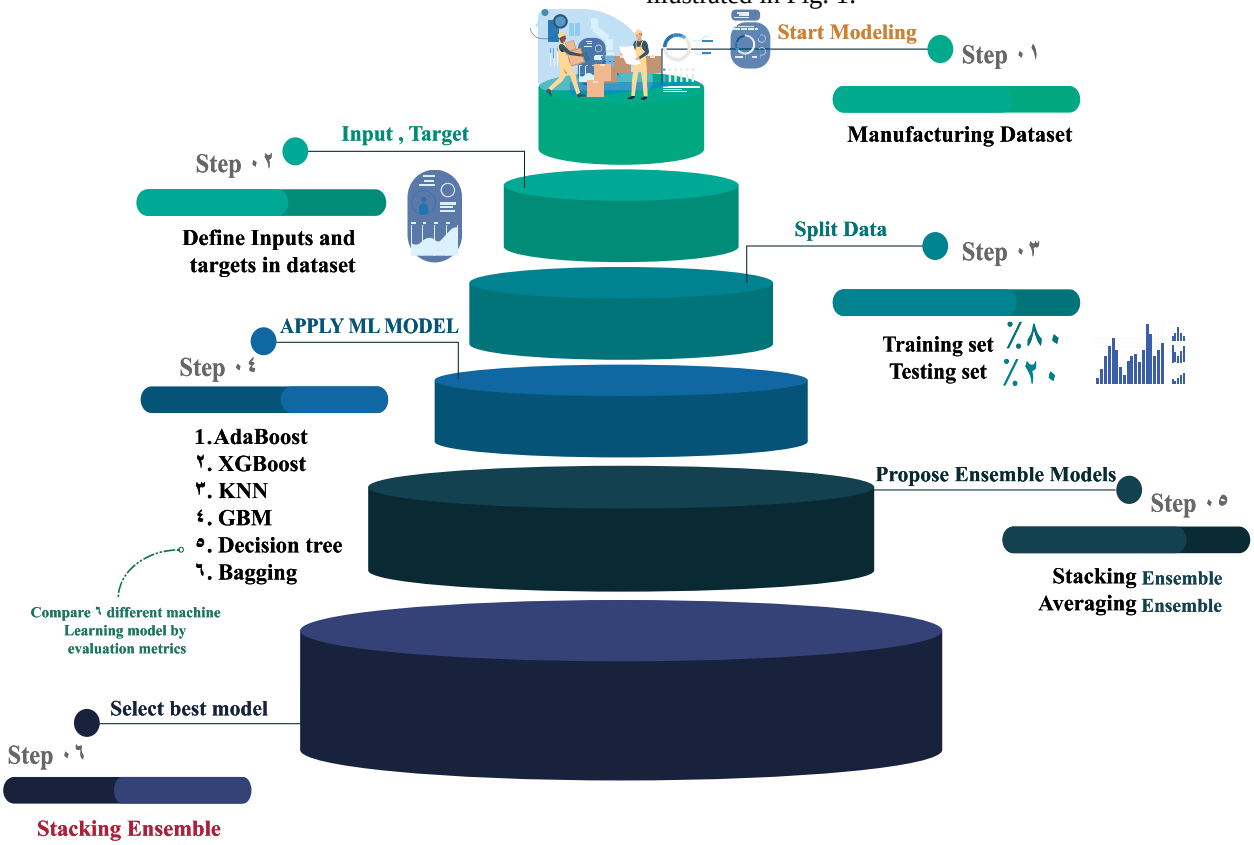


Figure 1: The overall modeling procedure for quality prediction in manufacturing.

The simulated dataset has 3,957 samples, which is a manufacturing process. Five numerical characteristics of each sample include Temperature (T), Pressure (P), the interaction of these characteristics (TxP), Material Fusion Metric (MFM), and Material Transformation Metric (MTM), and a target variable, the Quality Rating. There were no missing values, and imputation was not necessary. In order to assess this, the data were randomly split into a training set (80%, 3,165 samples) and a hold-out test set (20%, 792 samples). All features were standardized with the aid of StandardScaler of scikit-learn, and the model was fitted only to the training data to avoid information leakage.

The selection of the models and the optimization of the hyperparameters were performed with a powerful 5-fold cross-validation scheme (K-Fold (n-splits = 5, shuffle = True, random-state = 42)) on training data. This procedure offers an effective model performance estimate by training and assessing every model using five data subsets. The average RMSE and R^2 of these folds are reported as the final model performance. Wilcoxon signed-rank tests of the fold-level RMSE were conducted to statistically confirm differences in performance of the best-performing models. The model with the highest score in cross-validation was then retrained on the whole

training set and tested on the held-out test set to verify its generalization ability.

Comparisons were made between 8 ML models: six base models (AdaBoost, Bagging, Decision Tree, Gradient Boosting, K-Nearest Neighbors and XGBoost) and two ensemble models (Averaging and Stacking).. All the experiments were carried out in Python 3.10.11, using major libraries, such as pandas, numpy, scikit-learn, XGBoost, and SHAP to interpret models.

The independent variables were as follows:

- Temperature (°C) & Pressure (kPa): The fundamental process parameters.
- Temperature x Pressure (TxP): The interaction term, which represents the joint action of T and P.
- Material Fusion Metric (MFM): This is a derived measure ($T^2 + P^3$), which measures the fusion of material. Material.
- Transformation Metric (MTM): This is a derived measure ($T^3 – P^2$), which characterizes the process of material transformation.

The dependent variable, Quality Rating, serves as the comprehensive score for the final product's quality, and is a continuous, unbounded numerical score that represents a comprehensive assessment of the final product's quality, which the models are trained to predict.

Further, to evaluate the performance of each model, several evaluation criterion values were computed, such as MSE, RMSE, MAE, R2, NMSE, MDAPE, STD and VAF. MDAPE gives the median percentage error, robust to outliers, unlike standard MAE. VAF presents the percentage of variance explained, which is more easily interpreted than R^2 alone. Standard metrics (MSE, RMSE, MAE, R^2) measure overall fit, while MDAPE and VAF provide complementary, interpretable insights. Ultimately, the effectiveness of the suggested ML models was identified by contrasting and comparing the outcomes of these assessed criteria. Their equations are shown in Fig. 2.

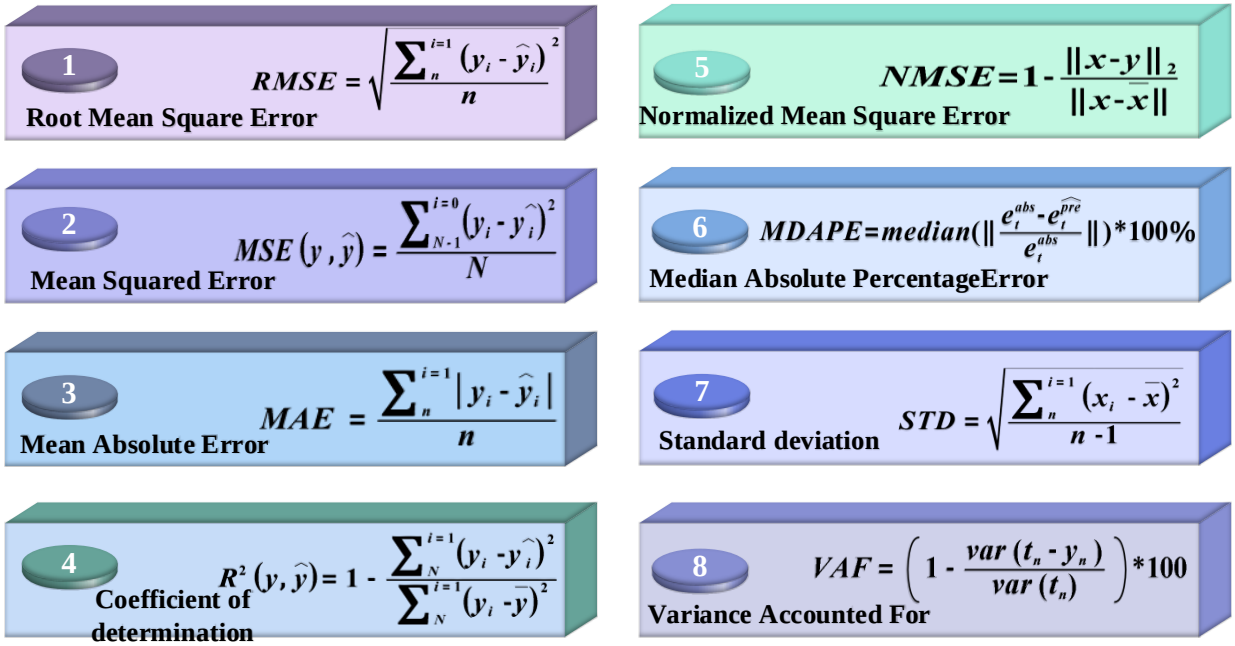


Figure 2: Equations for the evaluation of statistical metrics criteria

By accurately predicting manufacturing quality ratings, stakeholders and manufacturers can make informed decisions to optimize quality and prevent wasting costs and time. This project, which aims to use ML techniques to improve quality ratings in manufacturing, provides invaluable insights into the factors affecting manufacturing quality and offers optimization strategies for the production of manufacturing items.

3 ML models

3.1 Adaptive Boosting (AdaBoost)

AdaBoost is a sequential boosting, a type of algorithm that generates a powerful regressor by assembling a series of weak learners (e.g., decision stumps) [10]. It operates by training models repeatedly, and each time it increases the weight of false identifications [11]. This compels the following models to concentrate on instances that are more difficult to predict. The last prediction assumes that the so-called weak learners will additively make predictions, giving it high flexibility to reduce regression errors [12].

3.2 Bagging

The Bagging (Bootstrap Aggregating) is a parallel ensemble method, which tries to reduce variance. It creates numerous bootstrap copies of the initial information and educates a base model on each subset. Considering regression, the most probable prognosis will be a combination of all the individual prognoses. It works

by combining the output of individual models, which improves the stability of the predicting model by producing a more robust one that is generalized [13].

3.3 Decision Tree

A Decision Tree is used to model the predictions in a tree-like form of decisions. It divides the data into branches according to feature conditions which begin at a root node and culminate with leaf nodes that represent the ultimate predicted values. Although it is simple and interpretable, the complexity of a DT can increase with the amount of data. A tree that has the maximum possible number of branches will be able to reach high discrimination but may overfit in the absence of constraints [14].

3.4 Gradient Boosting (GB)

Gradient Boosting is a stage-wise ensemble algorithm, which builds models sequentially. It starts with a base prediction and successively includes new models that predict the negative gradient (residual errors) of the loss function of the existing ensemble. Every weak learner is tuned to correct all the mistakes of its predecessors, and its contribution is weighted by a learning rate [15]. The loss is minimized gradually until a very accurate predictor is obtained.

3.5 K-Nearest Neighbors (KNN)

KNN is an instance-based learning algorithm. In the case of regression, a target value is predicted at a new point in the data set by making the mean of the target values of the k nearest neighbors of that point in the feature space [16]. Euclidean distance is normally used in the measurement of distance. Although it is intuitively easy and efficient to represent the local patterns, its computational cost rises with the size of the dataset, because it needs to store the entire training set.

3.6 Extreme Gradient Boosting (XGBoost)

XGBoost is a regularized gradient boosting. It builds models sequentially, whereby the result of a sequence of successive trees is refined by its predecessors [17]. Some of the most significant advantages include its computational efficiency, its capability of handling missing values and the availability of both L1 and L2 regularization to control overfitting. It is an efficient and highly-utilized algorithm because it reduces a pre-defined loss function through a gradient descent optimization [18].

3.7 Averaging ensemble model

The Averaging Ensemble is a simple method that takes the mean or weighted average of predictions of several base models. This approach capitalizes on the wisdom of the crowd, in which the flaws of the individual models are likely to balance each other. It works best when there is diversity in the base models and they are accurate, which results in a reduced overall prediction variance and better performance as compared to a single model [19].

3.8 Stacking ensemble

Stacking Ensemble refers to a type of learning that combines several models to produce an improved final predictive performance [12]. The stacking model employs a meta-learner to generalize the predictions from the base models. The choice of this meta-learner is flexible and is not predefined by the method itself. The necessity of applying stacking is when multiple ML methods reveal various advantages for a certain task. In this case, the stacking ensemble method employs a discrete ML technique for specifying the efficient application of various algorithms [20]. In the Stacking model, presented in Fig. 3**,** training data are divided into different subsets, then blended and processed to produce distinctive predictions separately. Each prediction results are gathered and combined as the ultimate result of the processed training data in the study.

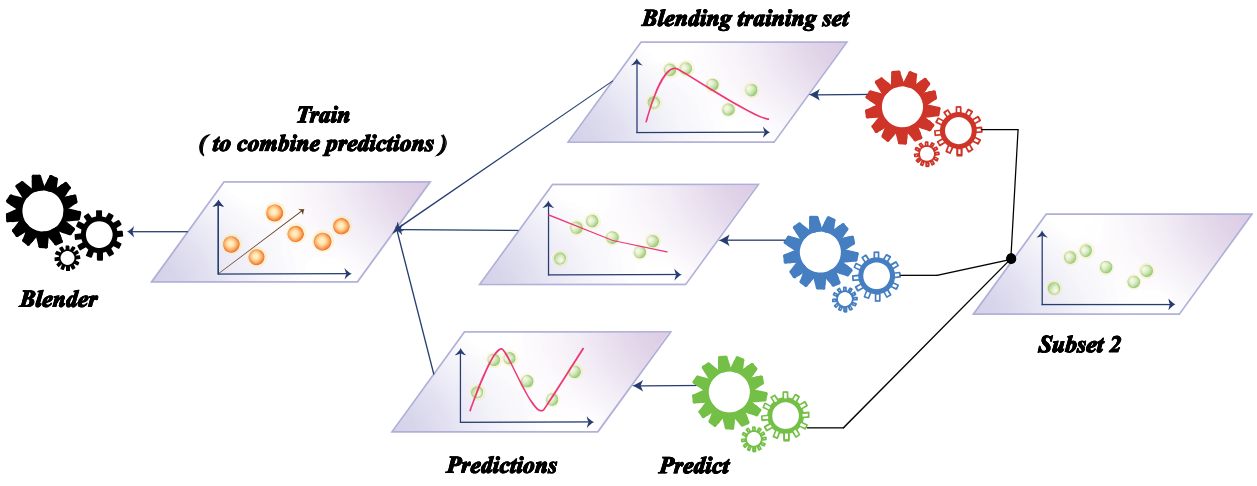


Figure 3: The steps of the Stacking Ensemble model applied for manufacturing quality rating predictions

4 Hyperparameter optimization

After the first model descriptions, hyperparameter tuning was done strictly to optimize the predictive ability of each algorithm. Optimization was done using the Grid Search technique with 5-fold cross-validation on the training data, where minimizing the Root Mean Square Error (RMSE) was the goal.

Table 2 shows the final, optimized hyperparameters of each model. This was essential to allow a fair comparison because it enabled every model to operate at its optimal level, and not in default settings.

Table 2: Hyperparameter optimization of each ML model.

Model	Hyperparameters
AdaBoost Regressor	n-estimators=150, learning-rate=0.1, random-state=42
Bagging Regressor	n-estimators=150, max-samples=0.8, max-features=0.8, random-state=42
Decision Tree Regressor	max-depth=10, min-samples-split=5, min-samples-leaf=2, random-state=42
Gradient Boosting Regressor	n-estimators=300, learning-rate=0.05, max-depth=3, subsample=0.9, random-state=42
XGBoost Regressor	n-estimators=300, learning-rate=0.05, max-depth=4, subsample=0.8, colsample-bytree=0.8, reg-lambda=1.0, objective='reg:squarederror', random-state=42
KNN Regressor	n-neighbors=7, weights='distance', metric='minkowski', p=2
Averaging Ensemble	Mean of predictions from: GB, XGBoost Regressor, KNN (same settings as above)
Stacking Regressor	Base learners: AdaBoost, Bagging, Decision Tree, GBM, XGB, KNN (all with above hyperparameters) Meta-learner: GB (n-estimators=200, learning-rate=0.05, max-depth=3, random-state=42)

These tuned models were then tested through 5-fold cross-validation, where the findings are summarized in Table 3. The cross-validation measures give a strong measure of the model's generalizability before the model testing on the final hold-out set.

Table 3: Results of 5-fold cross-validation

Model	CV-RMSE-Mean	CV-RMSE-Std	CV-R2-Mean	CV-R2-Std
AdaBoost	0.740076	0.063869	0.996689	0.000671
Bagging	0.086855	0.013201	0.999954	1.31E-05
Decision Tree	0.149076	0.014953	0.999865	3.09E-05
GBM	0.084631	0.015778	0.999956	1.41E-05
XGBoost	0.549707	0.05415	0.998187	0.000312
KNN	1.81702	0.336062	0.980178	0.005059
Stacking	0.129288	0.013833	0.9999	1.52E-05
Averaging	0.660486	0.126223	0.997362	0.000739

5 Results

This paper has compared the eight machine learning models that forecast the manufacturing quality ratings with respect to five process features. A k-fold cross-validation and a held-out test set were used to strictly compare the models and determine their performance based on a variety of statistical values (R^2, MSE, RMSE, MAE, VAF). In order to depict how closely related these variables were, a Pearson correlation heatmap was sketched as an effective color-coded visual matrix (see Fig. 4). Based on the correlation heatmap, several significant technical findings can be drawn. The obtained features, Material Fusion Metric (MFM) and Material Transformation Metric (MTM), have a near-perfect positive correlation with Temperature (0.97), which may indicate that they are functionally dependent upon it and may not be necessary to model. In addition, MFM and MTM are strongly correlated with each other (0.98), which means high levels of multicollinearity that may render some model interpretations unstable. Importantly, the target variable, Quality Rating, has moderate to strong negative correlations with Temperature (-0.46) and MFM (-0.51), and even stronger with MTM (-0.58), which are, in its turn, confirmed by the SHAP analysis.

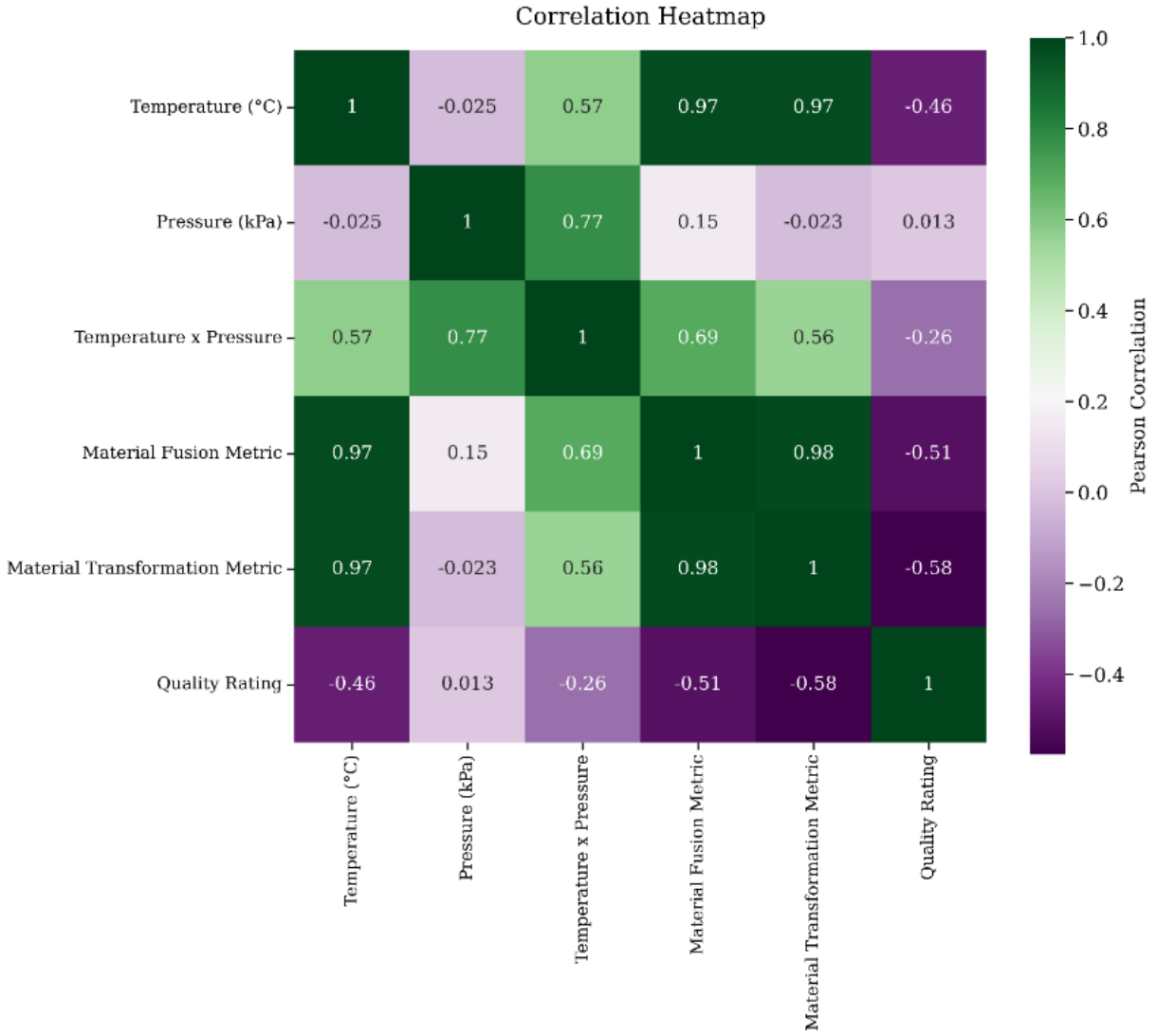


Figure 4: Pearson Correlation Heatmap for examining the relationship between the independent variables

Preliminary analysis of the six base models (AdaBoost, Bagging, DT, GB, KNN, XGBoost) showed that the majority of them demonstrated high and comparable results on the test sample, with R^2 and VAF metrics staying nearly at 0.999. Nevertheless, a closer examination of error measures showed that KNN is the most precise single model, with the lowest test errors (MSE: 0.006, RMSE: 0.078, MAE: 0.018). Conversely, AdaBoost and XGBoost showed significantly higher rates of error, which makes them the weakest among the base models (Table 4).

Table 4: Error metrics obtained by using the primary proposed models.

Model	Dataset	MSE	RMSE	MAE	R^2	NMSE	MDAPE	STD_dev	VAF
AdaBoost	Train	0.499	0.706	0.42	0.997	0.003	0.26	13.066	0.997
	Test	0.624	0.79	0.458	0.996	0.004	0.26	12.774	0.996
Bagging	Train	0.003	0.053	0.01	0.999	1.62E-05	3.34E-07	13.075	0.999
	Test	0.011	0.105	0.024	0.999	6.94E-05	5.14E-07	12.622	0.999
DT	Train	0	6.63E-07	0	0.999	2.57E-15	1.39E-07	13.085	0.999
	Test	0.012	0.107	0.022	0.999	7.26E-05	7.04E-07	12.612	0.999
GB	Train	0.002	0.047	0.015	0.999	1.31E-05	0.001	13.083	0.999
	Test	0.012	0.108	0.032	0.999	7.27E-05	0.001	12.602	0.999
XGB	Train	0.001	0.036	0.01	0.999	7.55E-06	0	13.085	0.999
	Test	0.461	0.679	0.147	0.997	0.003	0	12.695	0.997
KNN	Train	0.005	0.069	0.014	0.999	2.75E-05	1.17E-08	13.079	0.999
	Test	0.006	0.078	0.018	0.999	3.86E-05	2.09E-08	12.62	0.999

Considering Fig. 5 below, a comparison of data values obtained through the integrative ML parameters was conducted. This plot depicts the initial suggested models, which include AdaBoost, Bagging, DT, GB, KNN and XGBoost and distances of each to the target value. Each model trained on 80% and tested on 20% of the data. The findings of this plot indicate that the values of all proposed models overlap with the actual value, with KNN and Bagging showing the closest alignment.

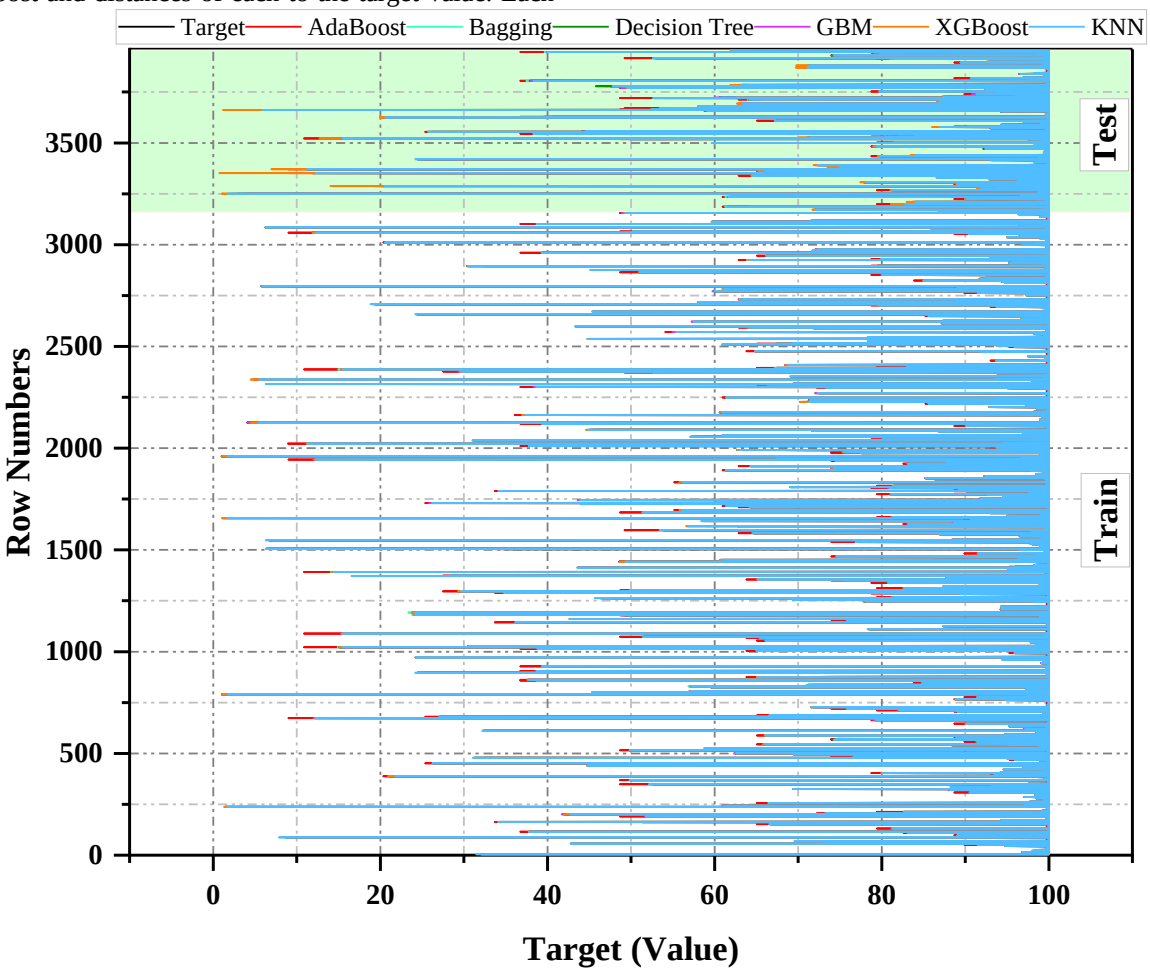


Figure 5: Target value plot comparing the primary proposed ML models for manufacturing quality predictions.

Additionally, as shown in Fig. 6, the error values of each proposed ML model are illustrated for both train and test data. A model with the least error values (i.e., closest to zero) is considered the best predictor among the employed ML models. This scheme visually evaluates the accuracy of model predictions. Based on this, among the base models, KNN emerged as the best model, exhibiting the lowest error values, while GB ranked second with the next lowest error. Conversely, AdaBoost was the least accurate, with the highest error values. This finding aligns with the subsequent histogram plot (Fig. 7), where the error density is analyzed. As evident in this figure, KNN and GB exhibit the lowest error density, making them the superior models, while AdaBoost shows a notably high error density, indicating its weaker predictive performance.

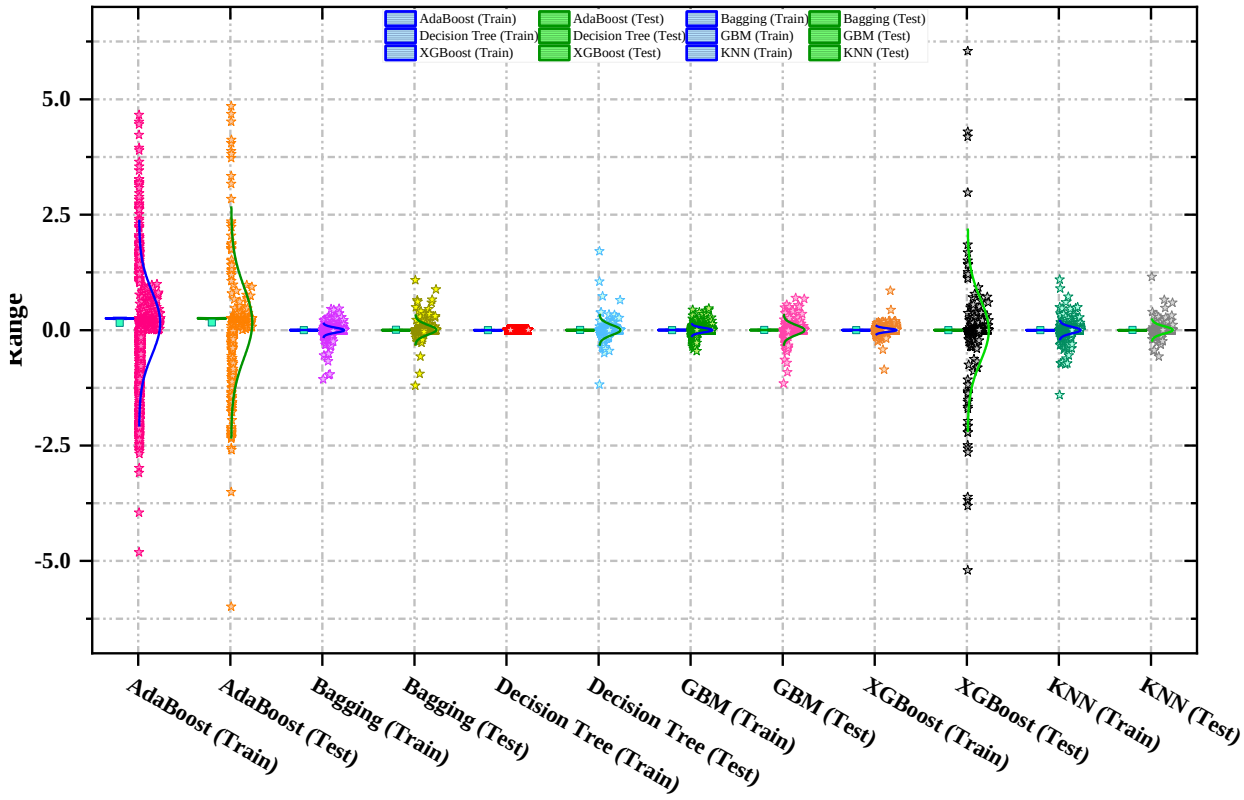


Figure 6: Boxplot of the error values of the proposed ML models for both train and test data in manufacturing quality prediction.

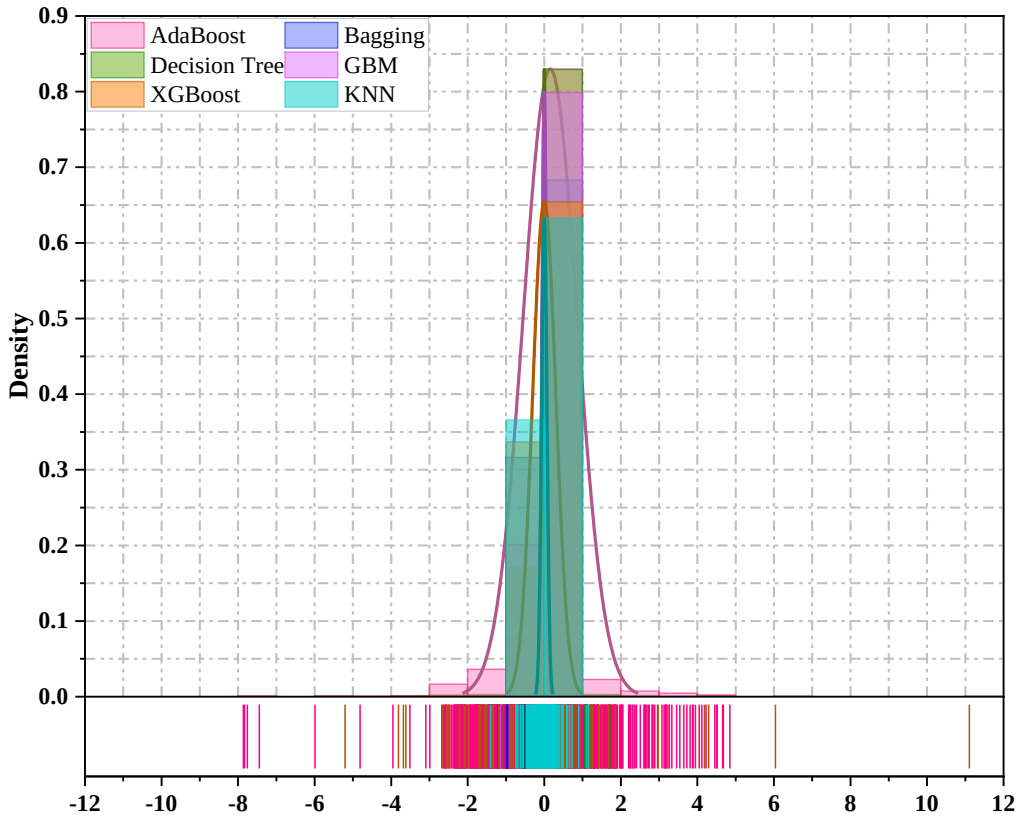


Figure 7: Histogram plot of the errors of the proposed ML models in manufacturing quality prediction

In addition to the output of Table 4, another graphic comparison has been illustrated based on eight significant error metrics below (see Figs. 8 and 9), fulfilling a precise and visual analysis of the primary models' performance in the present work. Clear from these plots, the histograms of each model's outcome are almost consistent with the

abovementioned results. A quick look into RMSE, MAE, MSE, and NMSE plots showed that KNN, GB, DT, and Bagging had almost similar performance, excluding AdaBoost and XGBoost, in the testing data. In Fig. 9, with respect to R^2, STD, VAF, and MDAPE histograms, all the models performed roughly the same once more.

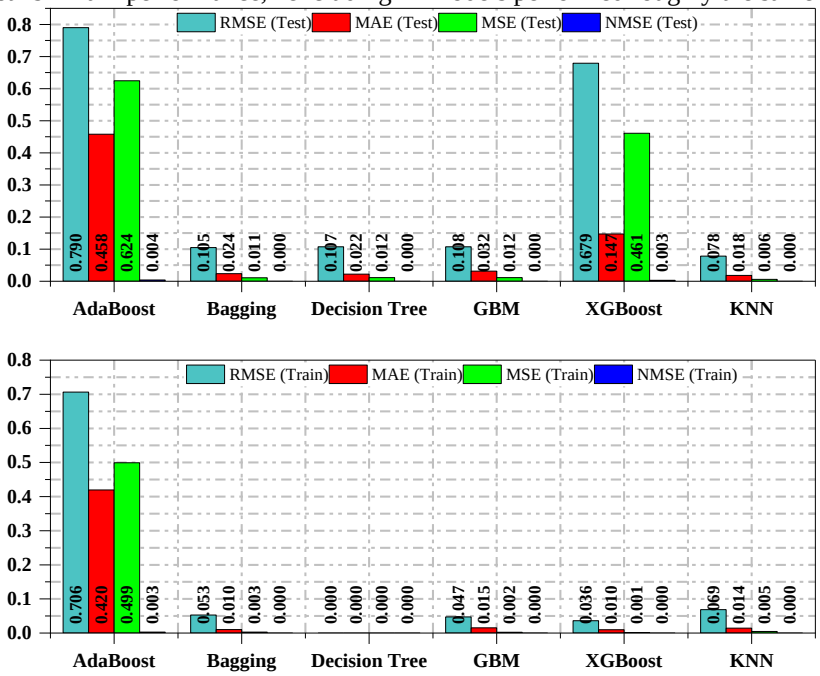


Figure 8: RMSE, MAE, MSE, NMSE histograms based on the primary proposed models' performance for both train and test data.

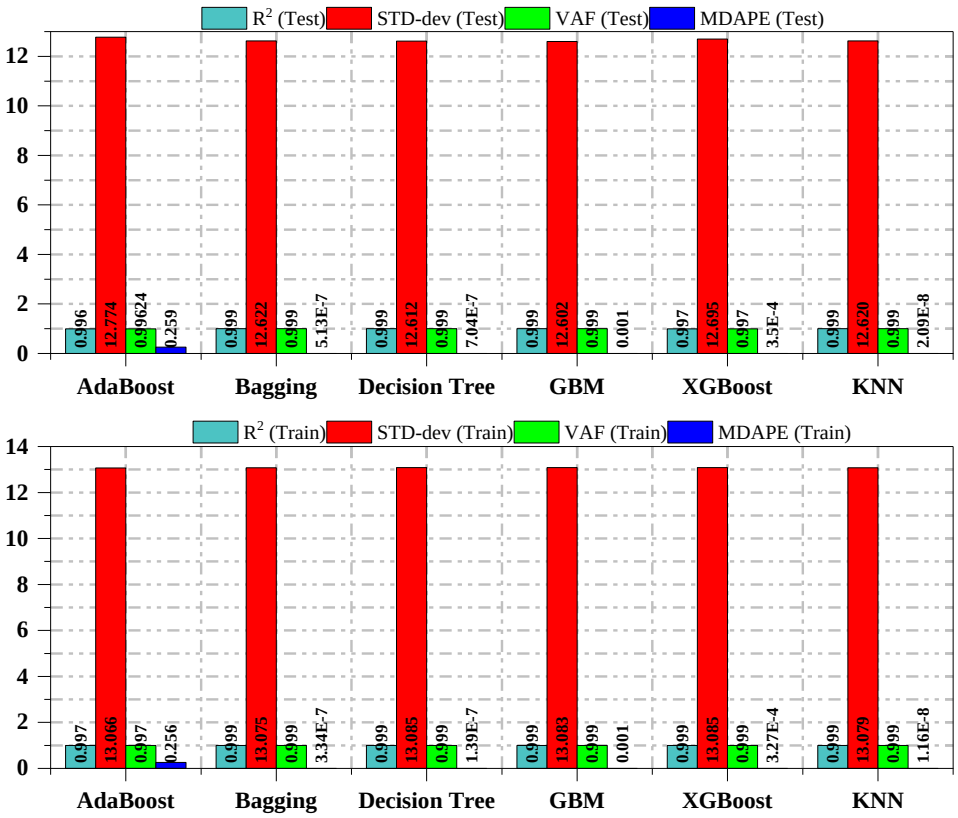


Figure 9: R^2, STD, VAF, MDAPE histograms based on the primary proposed models' performance for both train and test data

Since the performance of the top base models was closely matched, two sophisticated ensemble methods, Averaging and Stacking, were applied to determine whether the predictive accuracy could be improved. The Stacking ensemble that employed the GB meta-learner was superior. It had the lowest error measures on the test

set (MSE: 0.004, RMSE: 0.065, MAE: 0.016) with a perfect R^2 and VAF score of 0.999 (Table 5). This proves the hypothesis that a stacking ensemble can utilize the strengths of many base learners to produce the strongest and most accurate predictions.

Table 5: Error metrics of the two additive ensemble models.

Model	Dataset	MSE	RMSE	MAE	R^2	NMSE	MDAPE	STD-dev	VAF
Averaging	Train	0.017	0.131	0.078	0.999	0	0.047	13.083	0.999
	Test	0.033	0.181	0.092	0.999	0	0.047	12.656	0.999
Stacking	Train	0.002	0.048	0.009	0.999	1.37E-05	6.90E-07	13.089	0.999
	Test	0.004	0.065	0.016	0.999	2.69E-05	8.92E-07	12.628	0.999

Fig. 10 demonstrates the error distribution histograms of both Averaging and Stacking ensemble models, and gives a clear visual justification that the predictive performance of the Stacking ensemble model is superior to that of the Averaging ensemble model. The Stacking ensemble has a significantly smaller error profile centered around zero, meaning that it consistently predicts with a minimal error. Conversely, the Averaging model shows more spread of errors, which expresses greater variation in prediction. This specific difference highlights the effectiveness of the Stacking meta-learner in integrating the capabilities of its base models not only to reduce average errors but also to obtain a higher degree of reliability in predicting the quality of manufacturing.

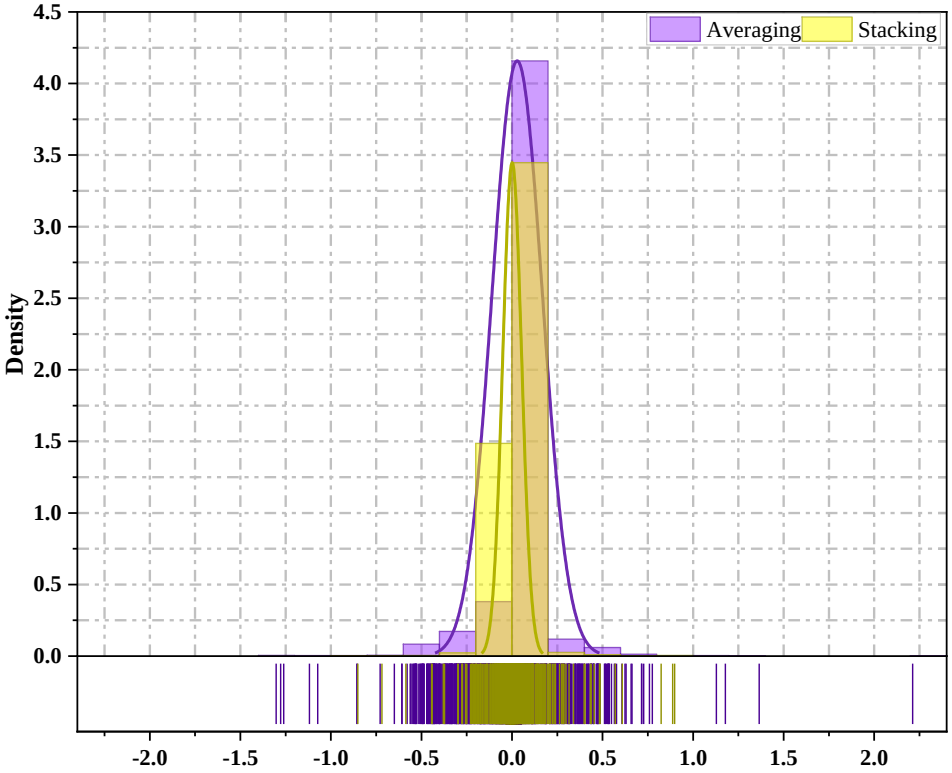


Figure 10: Histogram plot of the errors of the two additive ensemble models in manufacturing quality prediction.

For more scrutiny, another comparison for the accuracy and efficiency of the applied models in this study was carried out through examining R^2 scatter plots of these models. R^2 results as a significant error measure are presented in Fig. 11 and demonstrate the extent to which the predictions of the employed ML models correspond to the actual data. As seen in this figure, a model is deemed superior and more accurate if its prediction values nearly align with the norm line (when $R^2 = 1$). As shown, all the models served this purpose; i.e., their forecasts closely matched the norm line, excluding AdaBoost and XGBoost, which were weaker models. Hence, higher R^2 values were obtained in KNN, Stacking, Bagging, DT, GB, and Averaging, all of whose R^2 values equaled approximately 0.999 for both the train data and test data. However, the lowest RMSE was merely for the Stacking model, meriting being the best fit model; i.e., 0.048 and 0.065, respectively, for train and test data. As a result, staking outperformed the other models. This was also in accordance with the results in the above-mentioned training and testing tables' outcomes.

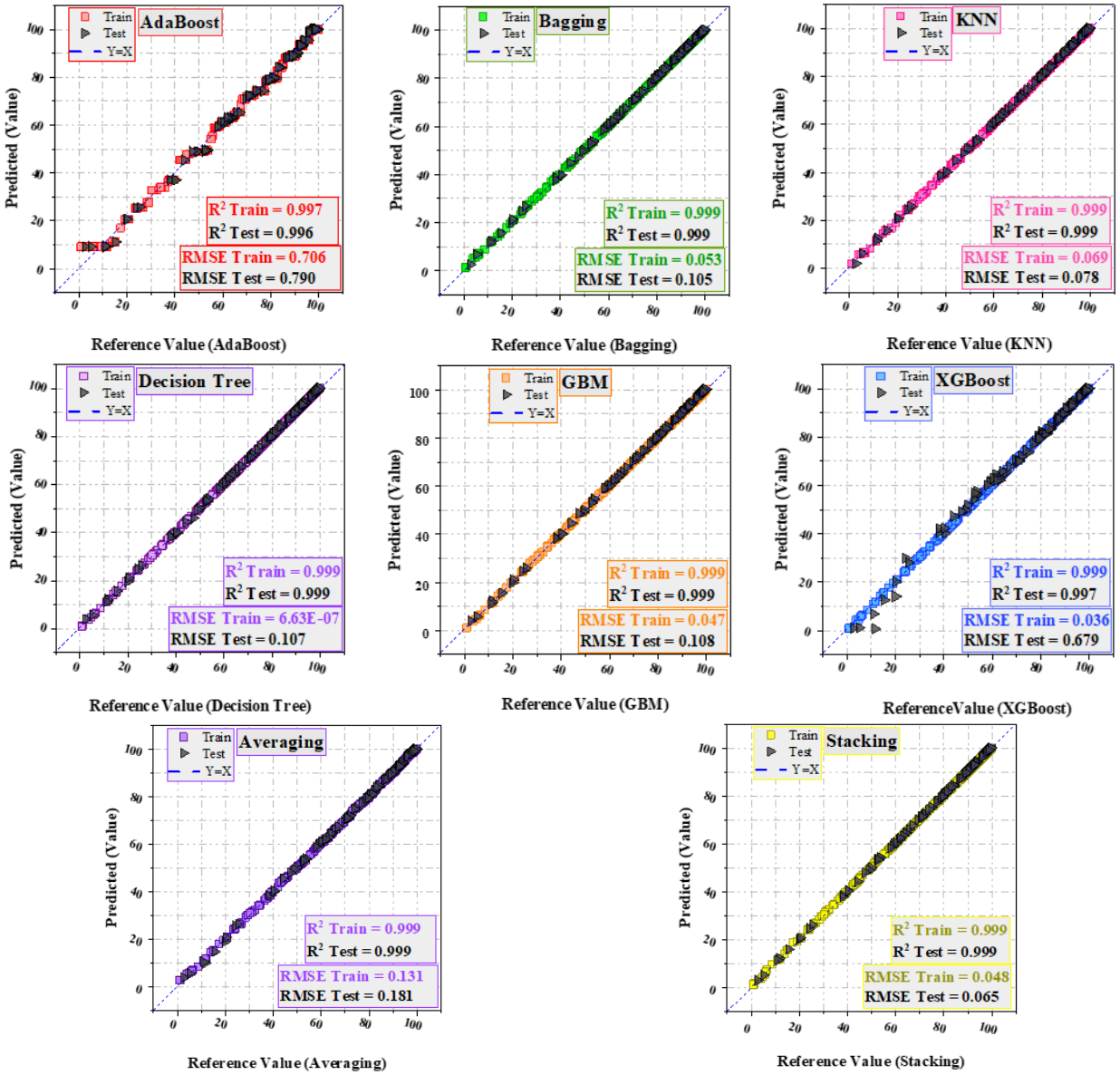
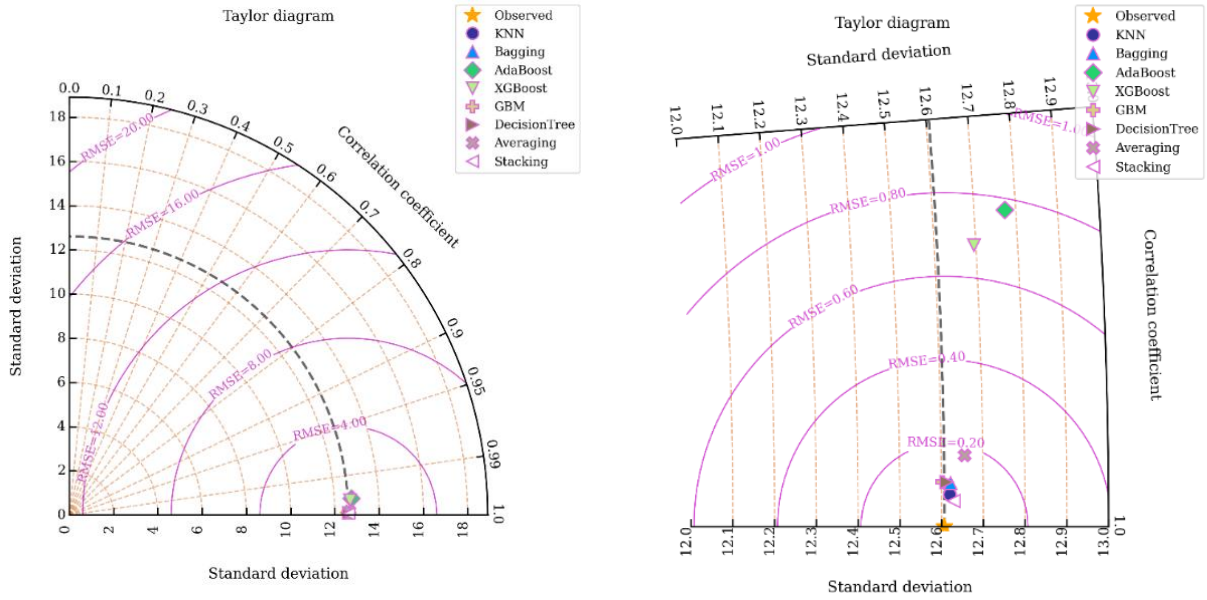


Figure 11: Comparing the coefficient of determination (R^2) for the individual proposed ML models

Another graphical diagram, named the Taylor diagram, has been used here for comparing the recommended models' performance. This chart compares models against one another based on the model accuracy using correlation coefficient, STD and RMSE [21]. The performance of the models is presented as a circle on this diagram where the better performance of the models is determined by the nearest points to the reference point of the reference [22]. The Taylor diagram is depicted in Fig. 12 for predicting manufacturing quality. According to the diagram, the relative performance of the models can be succinctly assessed using the distance between the models and the reference point of the models, which is an ideal model. The best-performing models, including Stacking, Bagging, GBM, and Decision Tree, are those that are closest to the reference point, with a high correlation coefficient (>0.99), a standard deviation that is nearly equal to that of the observations, and thus the lowest values of RMSE. Conversely, AdaBoost and XGBoost are further away, which reflects the fact that they have a high RMSE and weak correlation, which is consistent with their designation as the poorer performers in the study. The Stacking ensemble consistently appears within this top-performing cluster, providing a visual confirmation of its optimal balance between high accuracy and low error, as substantiated by the quantitative results.



Test Test (Zoom)

Figure 12: Taylor diagrams for the models' comparison based on RMSE, STD, R metrics

A series of pairwise Wilcoxon signed-rank tests was performed to statistically justify the performance differences in the cross-validation folds between the best-performing Stacking model and all other models. Table 6 shows that no pairwise comparison showed a statistically significant difference at the 0.05 level of significance (all p-values were larger than 0.05). This insignificance can probably be explained by the fact that the models already demonstrate extremely high and comparable performance (R^2 = 0.999), and it is hard to statistically prove that the difference is really present in this test. Thus, although the Stacking model is arguably the most accurate predictor based on its lower error scores, the Wilcoxon test indicates that its performance, in the strict statistical sense, does not differ significantly from the other base models, such as KNN and GBM, in this particular experimental setting.

Table 6: Results of the pairwise Wilcoxon tests.

Reference model	Compared model	Wilcoxon statistic	p-value	Significant-at-0.05
Stacking	AdaBoost	0	0.0625	FALSE
Stacking	Bagging	0	0.0625	FALSE
Stacking	Decision Tree	1	0.125	FALSE
Stacking	GBM	0	0.0625	FALSE
Stacking	XGBoost	0	0.0625	FALSE
Stacking	KNN	0	0.0625	FALSE
Stacking	Averaging	0	0.0625	FALSE

The Stacking model, as the best performing model, was examined with SHAP to explain its predictions and measure the effect of features to make sure that it was useful in practice. The analysis showed a definite order of feature value (Fig. 13). The most prevailing one was the Material Transformation Metric (MTM), then Temperature (T). The impacts of the Material Fusion Metric (MFM), the interaction term (TxP), and Pressure (P) were much lower.

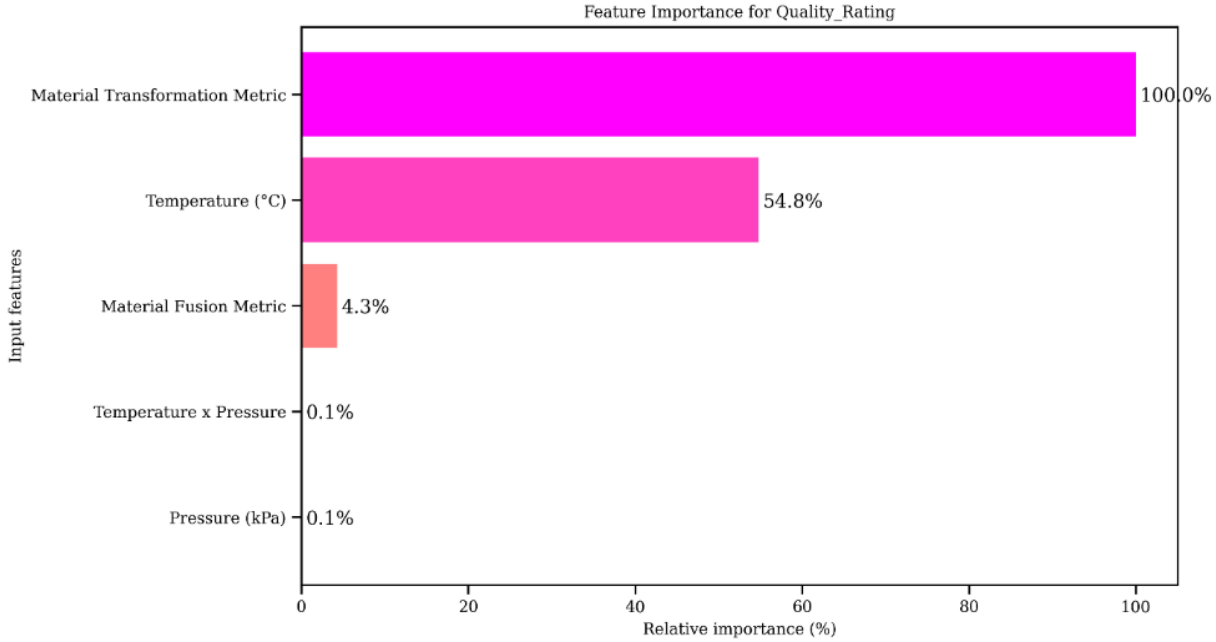


Figure 13: Features' importance for manufacturing quality rating predicted by the Stacking model

These relationships were further clarified by the SHAP summary plot (Fig. 14) [23], which revealed that greater values of MTM and T were consistently associated with higher predicted quality ratings. The individual SHAP dependence curves (Fig. 15) confirmed the steady and monotonic relationship of MTM and T with the output as opposed to the more dispersed and non-influential trends of MFM, TxP, and P.

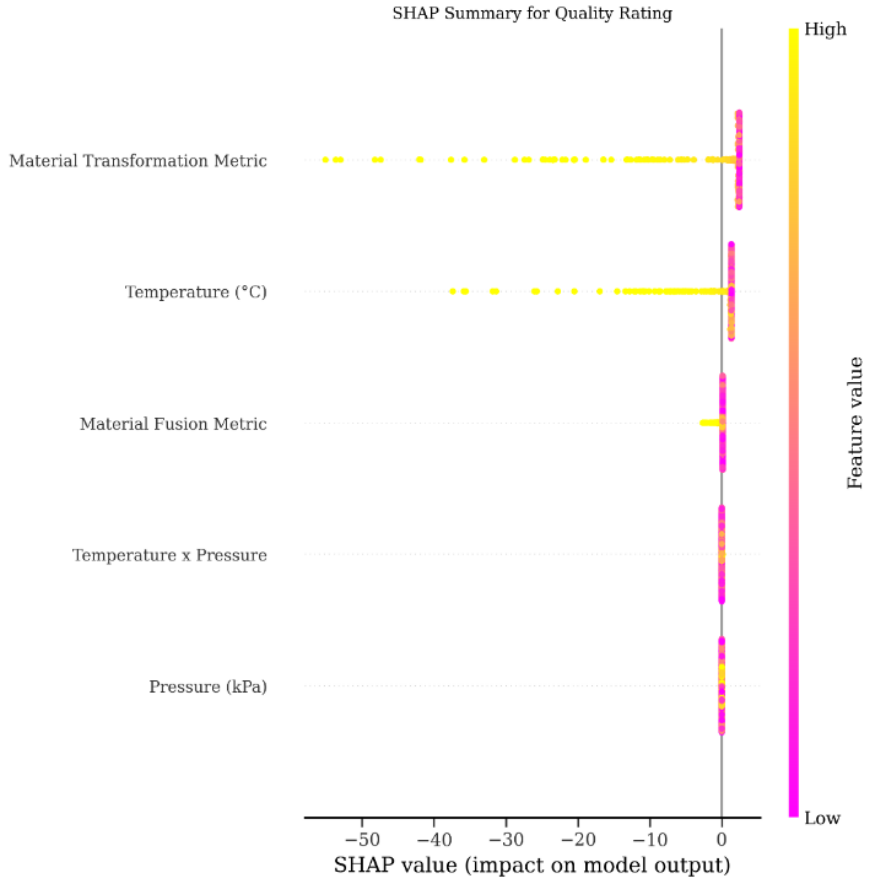


Figure 14: SHAP values providing an overview of the impacts of each manufacturing feature on the quality rating predictions by the selected Stacking model

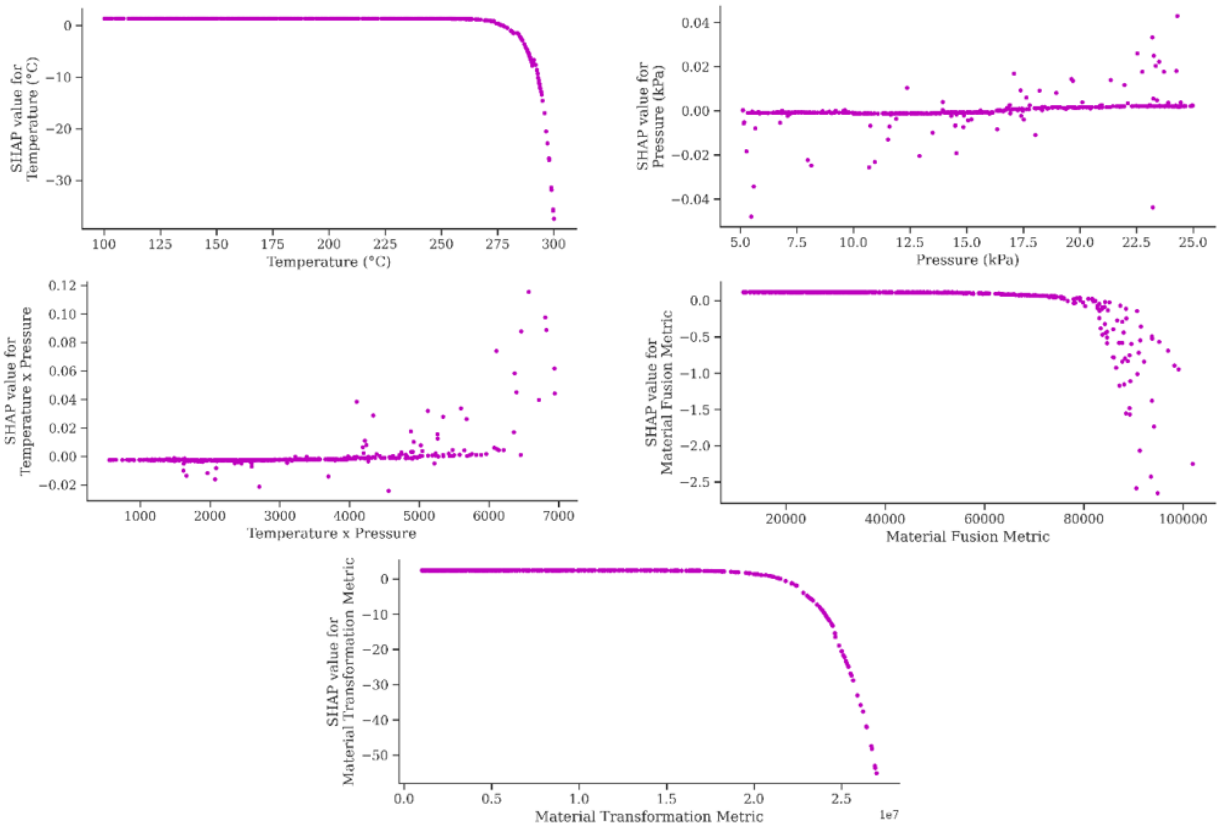


Figure 15: SHAP Subplots by Stacking model.

6 Discussion

This paper shows that the Stacking Ensemble model is the most effective for predicting manufacturing quality ratings, and the model provides almost perfect performance (R^2: 0.999) and low errors (RMSE: 0.065). This observation confirms the paper's hypothesis that a meta-learner can successfully combine the power of different base models to produce a more powerful predictor.

Compared to the existing literature (Table 1), this work's results contribute to the field in two important aspects. First, although earlier research, such as Li et al. [1] and Msakni et al. [6], achieved success in applying single or combined models to their particular situations, it finds that a general Stacking ensemble offers a final margin of performance improvement that could be pivotal to high-stakes, zero-defect manufacturing objectives. Second, it discusses the importance of model interpretability as emphasized by Link et al. [3] and Weichert et al. [4]. Using SHAP analysis on the best model, this study goes beyond a black box to numerically demonstrate that the most influential features are Material Transformation Metric (MTM) and Temperature (T), which can guide manufacturers to control their processes.

The Wilcoxon test revealed no statistically significant difference between the base models and the Stacking ensemble, which is probably because their performance is already exceptionally high. However, the Stacking model demonstrates a consistent tendency of having the lowest error metrics, which is why it can be selected as the most reliable one when it comes to the practical applications where marginal gains are of the essence. Therefore, the paper not only proposes an excellent predictive model but also offers a framework for exploiting the collective model intelligence with explainable outputs for decision-making in industry.

7 Conclusion

This research was able to create a high-precision predictive model of manufacturing quality ratings. After a thorough analysis of eight machine learning models, the Stacking Ensemble was determined to be the best model with almost perfect prediction (R^2 = 0.999) and minimal error rates (MSE = 0.004, RMSE = 0.065). The combination of several base learners via a meta-learner turned out to be the most effective strategy. Moreover, SHAP analysis offered essential interpretability, as MTM and Temperature were found to be the most significant features of quality, which could be used to introduce practical recommendations in terms of optimizing the process.

Nevertheless, this work has some limitations despite its high performance. Simulated data though controlled might not accurately reflect the noise and complexity of real-world production environments. On the other hand, the model was trained on a particular number of five features, which may not capture other contributory variables in real-life manufacturing contexts.

Further studies should concentrate on testing such results with actual industrial data to complete the practical generalizability of the model. It is also suggested to widen the feature set with more dynamic process parameters and

external factors. Future research may consider more sophisticated deep learning designs and online learning to enhance models in adaptive manufacturing systems. Finally, the adoption of more explainable AI (XAI) techniques can bring additional and more transparent insights into the cause-and-effect patterns in the production process.

Acknowledgements

This work was supported by.

Project 1: YB202215 Key Scientific Research Project of Jingchu University of Technology

Project 2: Provincial Teaching and Research Project for Universities in Hubei Province (2022450)

Project3: School-level Research Platform of Jingchu University of Technology:Data Analysis Science Laboratory

Project4: HX20220171 Horizontal Scientific Research Project of Jingchu University of Technology

Project5: Research project of Jingchu University of Technology (YB202301, JX2022-007)

Project6: Education Science Planning Project of Jingmen City (JMG2022004)

Project7: China Association for Educational Technology Project (XJJ202205022）

Project8: School Level Project, QN202419

References

[1] X. Li, Z. Huang, and W. Ning, “Intelligent manufacturing quality prediction model and evaluation system based on big data machine learning,” *Computers and Electrical Engineering*, vol. 111, p. 108904, 2023. https://doi.org/10.1016/j.compeleceng.2023.108904

[2] S. Sankhye and G. Hu, “Machine learning methods for quality prediction in production,” *Logistics*, vol. 4, no. 4, p. 35, 2020. https://doi.org/10.3390/logistics4040035

[3] P. Link *et al.*, “Capturing and incorporating expert knowledge into machine learning models for quality prediction in manufacturing,” *J Intell Manuf*, vol. 33, no. 7, pp. 2129–2142, 2022. https://doi.org/10.1007/s10845-022-01975-4

[4] D. Weichert, P. Link, A. Stoll, S. Rüping, S. Ihlenfeldt, and S. Wrobel, “A review of machine learning for the optimization of production processes,” *The International Journal of Advanced Manufacturing Technology*, vol. 104, no. 5, pp. 1889–1902, 2019. https://doi.org/10.1007/s00170-019-03988-5

[5] F. Psarommatis and V. Azamfirei, “Zero Defect Manufacturing: A complete guide for advanced and sustainable quality management,” *J Manuf Syst*, vol. 77, pp. 764–779, 2024. https://doi.org/10.1016/j.jmsy.2024.10.022

[6] M. K. Msakni, A. Risan, and P. Schütz, “Using machine learning prediction models for quality control: a case study from the automotive industry,” *Computational management science*, vol. 20, no. 1, p. 14, 2023. https://doi.org/10.1007/s10287-023-00448-0

[7] B. Yang, J. Huang, and Y. Chen, “The relationship between ESG ratings and digital technological innovation in manufacturing: Insights via dual machine learning models,” *Financ Res Lett*, vol. 71, p. 106362, 2025. https://doi.org/10.1016/j.frl.2024.106362

[8] P. Akbari, M. Zamani, and A. Mostafaei, “Machine learning predictions of spatter behavior in LPBF additive manufacturing,” *Materialia (Oxf)*, vol. 38, p. 102268, 2024. https://doi.org/10.1016/j.mtla.2024.102268

[9] V. Azamfirei, F. Psarommatis, and Y. Lagrosen, “Application of automation for in-line quality inspection, a zero-defect manufacturing approach,” *J Manuf Syst*, vol. 67, pp. 1–22, 2023. https://doi.org/10.1016/j.jmsy.2022.12.010

[10] S. Leiprecht, F. Behrens, T. Faber, and M. Finkenrath, “A comprehensive thermal load forecasting analysis based on machine learning algorithms,” *Energy Reports*, vol. 7, pp. 319–326, 2021. https://doi.org/10.1016/j.egyr.2021.08.140

[11] C. Ying, M. Qi-Guang, L. Jia-Chen, and G. Lin, “Advance and prospects of AdaBoost algorithm,” *Acta Automatica Sinica*, vol. 39, no. 6, pp. 745–758, 2013. https://doi.org/10.1016/S1874-1029(13)60052-X

[12] H. Peng, J. Xiong, C. Pi, X. Zhou, and Z. Wu, “A dynamic multi-objective optimization evolutionary algorithm with adaptive boosting,” *Swarm Evol Comput*, vol. 89, p. 101621, 2024. https://doi.org/10.1016/S1874-1029(13)60052-X

[13] A. Almomani *et al.*, “Age and Gender Classification Using Backpropagation and Bagging Algorithms.,” *Computers, Materials & Continua*, vol. 74, no. 2, 2023. DOI: 10.32604/cmc.2023.030567

[14] U. R. Konduru, A. P. Nagarajan, and C. V. S. Sai, “An improved performance of reversible data hiding in encrypted images using decision tree algorithm,” *Eng Appl Artif Intell*, vol. 137, p. 109100, 2024. https://doi.org/10.1016/j.engappai.2024.109100

[15] R. Huang, C. McMahan, B. Herrin, A. McLain, B. Cai, and S. Self, “Gradient boosting: A computationally efficient alternative to Markov chain Monte Carlo sampling for fitting large Bayesian spatio-temporal binomial regression models,” *Infect Dis Model*, vol. 10, no. 1, pp. 189–200, 2025. https://doi.org/10.1016/j.idm.2024.09.008

[16] L. Hardesty, “Explained: neural networks,” *MIT News*, vol. 14, 2017.

[17] T. Chen and C. Guestrin, “Xgboost: A scalable tree boosting system,” in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 2016, pp. 785–794. http://dx.doi.org/10.1145/2939672.2939785

[18] B. Kıyak, H. F. Öztop, F. Ertam, and İ. G. Aksoy, "An intelligent approach to investigate the effects of container orientation for PCM melting based on an XGBoost regression model," *Eng Anal Bound Elem*, vol. 161, pp. 202–213, 2024. https://doi.org/10.1016/j.enganabound.2024.01.018

[19] T. G. Dietterich, "Ensemble methods in machine learning," in *International workshop on multiple classifier systems*, Springer, 2000, pp. 1–15. https://doi.org/10.1007/3-540-45014-9_1

[20] U. Arif, C. Zhang, S. Hussain, and A. R. Abbasi, "An efficient interpretable stacking ensemble model for lung cancer prognosis," *Comput Biol Chem*, vol. 113, p. 108248, 2024. https://doi.org/10.1016/j.compbiolchem.2024.108248

[21] K. E. Taylor, "Summarizing multiple aspects of model performance in a single diagram," *Journal of geophysical research: atmospheres*, vol. 106, no. D7, pp. 7183–7192, 2001. https://doi.org/10.1029/2000JD900719

[22] M. Ehteram, A. N. Ahmed, P. Kumar, M. Sherif, and A. El-Shafie, "Predicting freshwater production and energy consumption in a seawater greenhouse based on ensemble frameworks using optimized multi-layer perceptron," *Energy Reports*, vol. 7, pp. 6308–6326, 2021. https://doi.org/10.1016/j.egyr.2021.09.079

[23] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," *Adv Neural Inf Process Syst*, vol. 30, 2017.