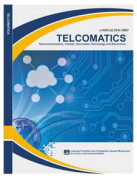
Available online at journal.uib.ac.id/index.php/telcomatics

TELCOMATICS

ISSN (online) 2541-5867



Product Quality Classification Based on Machine Learning in the Quality Control System of the Laser Metal Deposition Process

Elok Fiola, Rahma Neliyana, Try Yani Rizki Nur Rohmah, M. Syamsuddin Wisnubroto*, Fajri Farid

Department of Data Science, Faculty of Science, Institut Teknologi Sumatera, Jalan Terusan Ryacudu, Bandar Lampung, Indonesia

ARTICLE INFO

Keywords:
Laser Metal Deposition, Random Forest, Decision Tree, Support Vector Machine, Quality Control

Received: October 27, 2025
Revised: November 13, 2025
Accepted: November 20, 2025

***Corresponding author:**
E-mail: syamsuddin.wisnubroto@sd.itera.ac.id

DOI: 10.37253/telcomatics.v10i2.11432

ABSTRACT

Maintaining consistent product quality in Laser Metal Deposition (LMD) is challenging, as small parameter variations can cause critical defects. Traditional quality control methods are often manual, time-consuming, and error-prone, limiting their industrial scalability. The development of Industry 4.0 technologies enables the integration of smart sensors and machine learning to create intelligent and data-driven quality monitoring systems. This research focuses on classifying product quality in the Laser Metal Deposition (LMD) process by applying three machine learning algorithms, namely Decision Tree (DT), Random Forest (RF), and Support Vector Machine (SVM). The dataset consists of four numerical sensor variables, including Optical Sensor, Laser Power, Pressure, and Temperature, with Defect Label as the binary target variable. The Synthetic Minority Oversampling Technique (SMOTE) is used to balance the class distribution. Correlation analysis reveals weak linear relationships among all variables, suggesting the presence of complex non-linear interactions. The Random Forest model produces the best performance with accuracy of 0.88, recall of 0.79, and AUC of 0.80, outperforming Decision Tree and SVM. These findings indicate that ensemble-based methods effectively capture complex patterns within sensor data and provide reliable predictions for intelligent quality control in advanced metal manufacturing systems.

I. INTRODUCTION

The manufacturing industry has experienced rapid growth in line with the implementation of the *Industry 4.0* concept, which emphasizes the integration of automation, smart sensors, and artificial intelligence within production systems [1]. This transformation has changed the way companies maintain product quality consistency to ensure compliance with established standards. Quality control (QC) has become a vital component in manufacturing processes, as it determines both the reliability of production outcomes and the company's reputation. Products that fail to meet quality standards can lead to serious consequences, ranging from increased production costs to decreased customer satisfaction. Conventional manual inspection systems, which are still widely used, tend to be slow, rely on operator subjectivity, and are inefficient when dealing with large-scale production. Technological advancements have opened opportunities for the adoption of machine learning (ML) based methods capable of analyzing data in real time, thereby enabling faster decision-making [2]. Simanjuntak *et al.* (2025) demonstrated that the integration of sensors in an IoT-based system can process data in real time and provide rapid responses through digital notifications [3].

The need for adaptive and accurate quality control systems serves as the main foundation of this research. The industrial sector demands a mechanism for detecting nonconforming products that can operate quickly and consistently under various production conditions. Previous studies have demonstrated that the application of ML is effective in improving quality control performance. Vorhemus *et al.* (2020) successfully enhanced the accuracy of industrial packaging defect detection using a Random Forest algorithm based on image data. Wurdianto *et al.* (2022) showed that Random Forest implementation reduced errors in steel quality evaluation to below 10% compared to conventional methods [4]. Han *et al.* (2021) demonstrated that Support Vector Machine (SVM) algorithms effectively classified numerical sensor data to predict production quality outcomes [5], whereas Zhang *et al.* (2020) found that a combination of Decision Tree and ensemble methods increased the accuracy of nonconformity detection in electronic components to more than 90% [6]. These findings collectively confirm that ML holds significant potential to support QC systems, although most studies still focus on image-based data or specific sectors such as automotive and metal industries. Research on numerical sensor data remains limited, despite its efficiency and relevance to modern industrial automation.

This study aims to apply the Random Forest, Decision Tree, and Support Vector Machine algorithms to sensor-based Quality Control (QC) systems in order to analyze numerical sensor data for product quality determination. The research focuses on evaluating the effectiveness of machine learning models in recognizing sensor data patterns to detect production outcomes that deviate from quality standards. The novelty of this work lies in applying classification algorithms to non-

image numerical sensor data, which have not been extensively explored in the Quality Control context. This study is expected to broaden the application of ML in Quality Control systems within metal manufacturing industries using numerical sensor data.

II. Literature Review

A. Decision Tree

Decision Tree (DT) is a tree-structured classification model that recursively divides data based on the attribute with the highest information gain until it forms easily interpretable decision rules [7]. The research findings indicate that the DT algorithm achieved the highest accuracy of 92% compared to Naive Bayes and K-Nearest Neighbor (K-NN) in classifying healthcare data, demonstrating its capability to capture complex relationships among variables and produce accurate, interpretable decisions. This advantage makes the DT that require a balance between predictive accuracy and result interpretability [8].

The implementation of DT on electronic production sensor data has been proven effective in distinguishing between normal and abnormal product conditions, achieving an accuracy level of up to 89% [9]. Its stability can further improve by approximately 10% when the optimal tree depth and minimum sample size are properly configured [10]. The choice of splitting criteria such as Gini or entropy not only affects accuracy but also determines the model's sensitivity to minority classes [11]. The Gini index measures the impurity level of a node, while entropy quantifies the uncertainty of class distribution within a node. Their respective values can be calculated using Equations (1) and (2) [12]. Both metrics are then utilized to determine the optimal split based on the Information Gain (IG), as defined in Equation (3) [13].

$$Gini(D) = 1 - \sum_{i=1}^{k} p_i^2 \quad (1)$$

$$Entropy(D) = -\sum_{i=1}^{k} p_i \log_2(p_i) \quad (2)$$

$$Gain(A) = Entropy(D) - \sum_{v=1}^{V} \frac{|D_v|}{|D|} Entropy(D_v) \quad (3)$$

where D denotes the data node, k is the number of target classes, and p_i is the class proportion within D. The attribute A serves as the splitting variable, while D_v denotes the subset formed according to the value of A.

On imbalanced datasets, the combination of DT with balancing techniques such as SMOTE has been proven to increase the recall of minority classes by up to 20% without reducing precision [14]. DT can also be employed to develop fast detection systems based on simple rule sets, demonstrating its capability for real-time application in quality control [15]. In general, DT remains a fundamental model that functions not only as a standalone classifier but also as the foundation for ensemble algorithms such as Random Forest and Gradient Boosting. Its advantages lie in ease of interpretation and low computational cost, making it ideal for sensor-based QC systems [16].

B. Random Forest

Random Forest (RF) is an ensemble learning method that constructs multiple decision trees using a bootstrap sampling approach, which enhances model robustness while minimizing overfitting tendencies. Each tree is trained on a random subset of features, a strategy that reduces correlation among trees and leads to more stable and consistent predictions compared to a single Decision Tree [16]. As an ensemble of decision trees, RF utilizes measures such as Gini index, entropy, and information gain in its splitting process. The final classification output is determined through a voting strategy, where each tree contributes a vote, and the class with the majority votes is selected as the final prediction, as shown in Equation (4) [17]. Increasing the number of trees up to a certain limit can reduce model variance without significantly decreasing accuracy [18], making RF a commonly used baseline model [19].

$$\hat{y} = \text{mode}\{ h_1(x), h_2(x), \dots, h_T(x)\} \quad (4)$$

where $\hat{y}$ denotes the final predicted class produced by the Random Forest model, $h_t(x)$ represents the prediction result obtained from the t^{th} decision tree for an input x, and T indicates the total number of trees included in the ensemble.

The RF algorithm is effectively used to analyze complex industrial sensor data that contain noise within quality control (QC) processes. In a surface inspection system based on pressure sensor data, RF demonstrated a performance accuracy of 92.6% [15], while other studies reported comparable results in detecting nonconformities in electronic products with an error rate below 10% [20]. Moreover, RF was able to reduce false defect detection by up to 28% compared to traditional linear models [21]. Another strength of RF lies in its capability to perform feature importance evaluation, which identifies the sensors that contribute most significantly to product quality [22-23].

Furthermore, handling imbalanced data is a crucial factor in industrial applications. The integration of RF with the Synthetic Minority Oversampling Technique (SMOTE) has been proven to enhance model sensitivity toward minority classes [24], this approach can improve recall by up to 15% without reducing precision [25]. In the manufacturing domain, RF has been applied to textile sensor data combined with oversampling and hyperparameter tuning using Randomized Search [26]. The use of ROC-AUC and PR-AUC metrics is recommended over a single accuracy measure when evaluating model performance under imbalanced class distributions [27].

Overall, Random Forest (RF) is widely regarded as a robust algorithm for detecting product anomalies in quality control systems, offering stable performance compared to Support Vector Machine (SVM) and Logistic Regression (LR), particularly for non-image sensor data. Its strong predictive ability and adaptability make it one of the most reliable models for ML-based quality control [28-29].

C. Support Vector Machine

Support Vector Machine (SVM) is a classification algorithm based on the maximum margin theory, which aims to find an optimal hyperplane that separates two or more classes with the maximum distance between decision boundaries [30]. In general, the formula for separating target classes using a hyperplane is expressed in Equation (5) [31]. Through kernel functions such as Radial Basis Function (RBF) and Polynomial, SVM can map non-linear data into a higher-dimensional space without explicit computation, making it highly effective for detecting complex patterns in industrial sensor data. Equation (6) defines the Radial Basis Function (RBF) kernel, which is commonly used in SVM applications [30].

$$f(x) = \omega \cdot x + b \tag{5}$$

$$K(x_i, x_j) = \exp(-\gamma |x_i - x_j|^2) \tag{6}$$

where ω denotes the weight vector, x represents the feature vector of the input data, and b is the bias term. The function $K(x_i, x_j)$ indicates the kernel value computed between two samples x_i and x_j, where $\| x_i - x_j \|^2$ denotes their squared Euclidean distance, and γ is the kernel parameter that controls the spread of the radial basis function.

SVM has been proven effective in classifying machine vibration sensor data with an accuracy of 96%, even under imbalanced class conditions [32]. Furthermore, SVM can identify micro-damages faster than linear algorithms in bearing fault detection [33]. It also demonstrates superior performance in predicting machine failures up to 15 minutes before the actual breakdown in sensor-based inspection systems, achieving an accuracy above 90% and an AUC of 0.97 [28].

The performance of SVM is highly influenced by the selection of the *C* and *gamma* parameters. Equation (7) represents the SVM objective function, which minimizes both the weight magnitude and the penalty for margin violations through parameter *C*, controlling the trade-off between margin width and misclassification rate in the training data. Equation (8) defines that each training sample must lie on the correct side of the margin, with ξ_i serving as the error tolerance variable, while Equation (9) explains the optimization of *C* and γ parameters through Grid Search and Cross-Validation to obtain the best combination and enhance model generalization [31-34]. Parameter optimization via grid search can improve average accuracy by approximately 8–12% compared to using default parameter settings [35].

$$\min_{w,b,\xi} \frac{1}{2}|w|^2 + C\sum_{i=1}^{n} \xi_i \tag{7}$$

subject to,

$$y_i(w^T x_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0 \tag{8}$$

$$(C^*, \gamma^*) = \arg\max_{C,\gamma} Accuracy(C, \gamma) \tag{9}$$

where w denotes the weight vector, b represents the bias term, and ξ_i is the slack variable that allows margin violations. The parameter C serves as the regularization coefficient controlling the trade-off between margin width and classification errors, while n indicates the total number of samples.

III. Research Methods

A. Type of Research

This study falls under the category of quantitative experimental research, employing a comparative approach aimed at evaluating the performance of several machine learning algorithms in classifying product quality based on sensor data. This approach is adopted because it allows for objective model performance assessment through numerical measurement of specific evaluation metrics. The research is conducted using a supervised learning framework, in which each data sample has been labeled with a binary class (defective and non-defective).

B. Data and Data Sources

The dataset used in this study is the Laser Metal Deposition (LMD) Sensor for Defect Detection and Quality Control, obtained from the Kaggle platform [36]. This dataset consists of synthetic data simulating sensor readings in the LMD process, which is one of the commonly used additive manufacturing methods in the metal manufacturing industry. Each data record represents a unique sample containing readings from five main sensors, namely Optical Sensor, Laser Power, Pressure, Temperature, and Defect Label as the target variable. The target label is binary, where a value of 1 indicates a defective product and 0 indicates a normal (non-defective) product. The dataset comprises 1,000 numerically scaled samples that represent realistic conditions of sensor-based QC systems in dynamic manufacturing environments, making it relevant for developing and evaluating defect detection algorithms.

The dataset was split into two subsets using a stratified sampling approach to preserve class distribution, where 70% of the data were assigned for training and the remaining 30% for testing. To address class imbalance, the Synthetic Minority Oversampling Technique (SMOTE) was applied to equalize the number of samples across classes in the training data.

C. Research Flowchart

The flowchart in Figure 1 illustrates the stages of the research, starting from data input and preprocessing, which include cleaning and format adjustment to make the data ready for use. The data were then divided into training and testing sets before model training was carried out using three algorithms, namely Decision Tree, Random Forest, and Support Vector Machine. In the model training stage, parameter tuning was performed to determine the configuration that produces the best performance. The resulting models were then evaluated using the testing data to assess how accurately they could perform classification. This process ensures that the models obtained are not only optimal for the training data but also possess good generalization ability toward new data. After training, analysis and comparison of the results were conducted to evaluate the performance of each model, ultimately identifying the algorithm with the best performance for classification in the QC process.

D. Model Evaluation

Model evaluation was carried out to measure how well the classification algorithms could recognize product quality conditions. The performance of each model was evaluated using the confusion matrix metrics, namely accuracy, precision, recall, F1-score, and ROC-AUC value, which are generally recommended in machine learning research for binary classification tasks [37].

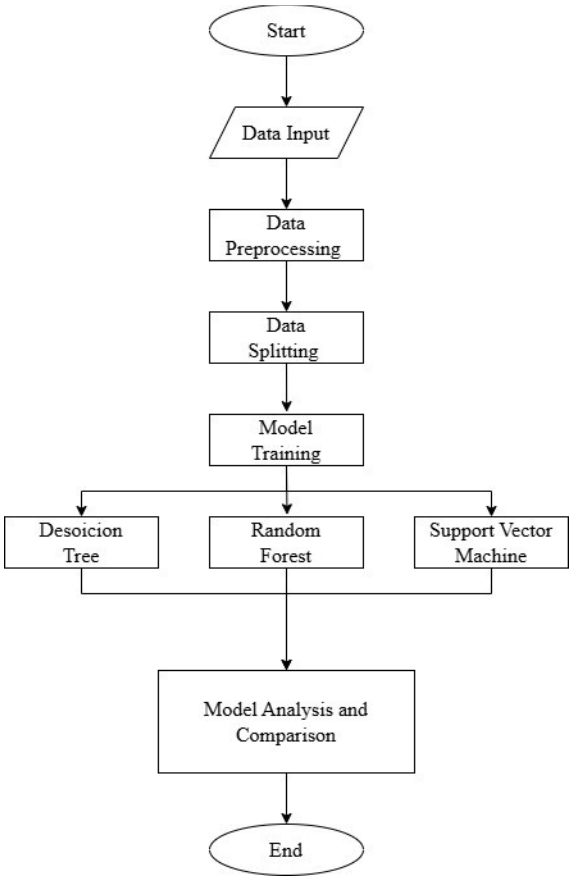


Figure 1. Research Flowchart

Table 1. Optimal Parameters of Classification Models

Model	Parameter
Decision Tree	criterion: gini
	splitter: best
	max_depth: 6
	min_samples_split: 4
	min_samples_leaf: 2
	max_features: sqrt
Random Forest	n_estimators: 600
	min_samples_split: 2
	in_samples_leaf: 2
	max_features: sqrt
	max_depth: 9
	bootstrap: True
Super Vector Machine	*C*: 10
	gamma: 0.01
	kernel: sigmoid

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (10)$$

$$Precision = \frac{TP}{TP + FP} \quad (11)$$

$$Recall = \frac{TP}{TP + FN} \quad (12)$$

$$F1\text{-}Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (13)$$

where TP and TN represent the numbers of correctly predicted positive and negative samples, respectively, while FP and FN denote the numbers of incorrectly predicted positive and negative samples. These values are used to compute the four standard classification metrics: Accuracy, Precision, Recall, and F1-Score.

Accuracy is used as a basic metric to measure the proportion of correct predictions against the overall test data, as shown in Equation (10). The higher the accuracy value, the better the model's ability to perform general classification. However, accuracy is not sufficiently representative for datasets with significant class imbalance because it may be biased toward the majority class [38]. To overcome this limitation, two additional metrics are used, namely precision and sensitivity (recall). Precision measures the model's correctness in predicting positive classes accurately among all positive predictions, as stated in Equation (11).

Meanwhile, sensitivity or recall measures the model's ability to detect all truly positive data, as shown in Equation (12). A high precision value indicates that the model rarely makes errors in predicting positive classes, while a high recall value indicates the model's capability to detect all positive cases accurately. The F1-score metric is used to obtain a balance between precision and recall, which can be calculated using Equation (13).

E. Model Parameters

The process of determining the optimal parameters was carried out using the Grid Search Cross-Validation method to obtain configuration combinations that provide the best performance on the training data. The selected parameters represent a balance between accuracy, generalization capability, and computational efficiency. The details of the tuned parameters for the three classification models are presented in Table 1.

Based on the tuning results, DT, RF, and SVM models were configured using the optimal parameters summarized in Table 1. The Decision Tree was optimized to control model complexity, the Random Forest to improve prediction stability through bootstrap aggregation, and the SVM to capture non-linear patterns using a sigmoid kernel. The model training process also incorporated StandardScaler and SMOTE to ensure balanced class distribution. The optimal parameters listed in Table 1 were applied in each respective model during the training phase to ensure consistent comparison of performance. These configurations served as the baseline for evaluating the classification results presented in the subsequent figures.

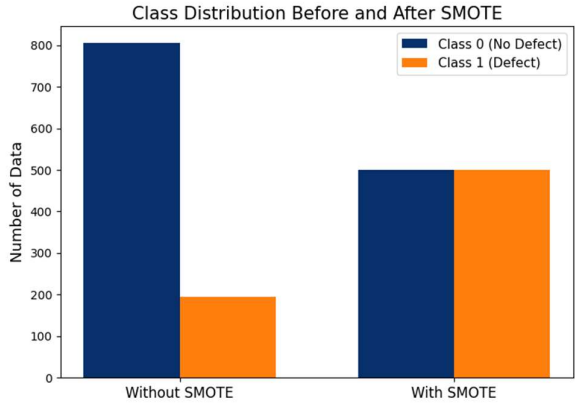


Figure 2. Class Distribution Before and After SMOTE

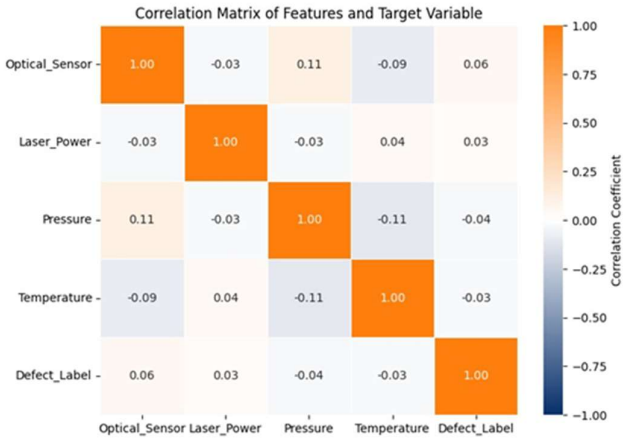


Figure 3. Correlation Matrix Among Variables

IV. Results and Discussion

The study applied three machine learning algorithms, namely Decision Tree, Random Forest, and Support Vector Machine, to classify product conditions in the Laser Metal Deposition (LMD) process. The dataset contains readings from five numerical sensors, namely Optical Sensor, Laser Power, Pressure, Temperature, and Defect Label as the binary target. The data proportion in Figure 2 shows the dominance of the no-defect class, therefore data balancing using the Synthetic Minority Oversampling Technique (SMOTE) was required for the training data.

The application of SMOTE successfully balanced the distribution between the no-defect (0) and defect (1) classes, each reaching 50%, as shown in Figure 2. The balanced data were then used during the model training process using Decision Tree, Random Forest, and Support Vector Machine.

The correlation matrix in Figure 3 shows the relationships among variables in the dataset. The findings indicate that the correlations of Temperature and Pressure with the target class are very weak, with correlation values of -0.03 and -0.04, respectively. The low correlation values indicate non-linear relationships among the variables, therefore the Decision Tree, Random Forest, and Support Vector Machine models were selected to capture non-linear patterns that could not be detected through linear relationships.

Subsequently, the initial evaluation of the Decision Tree model in Figure 4 shows an AUC value of 0.59, indicating that the classification ability is still low since the model has not yet been able to consistently distinguish between the two classes. The ROC curve in Figure 5 shows the performance of the Random Forest model in distinguishing classes with an AUC value of 0.80. This value indicates good classification capability, showing that the model can effectively distinguish between the two classes with a good balance between sensitivity and specificity.

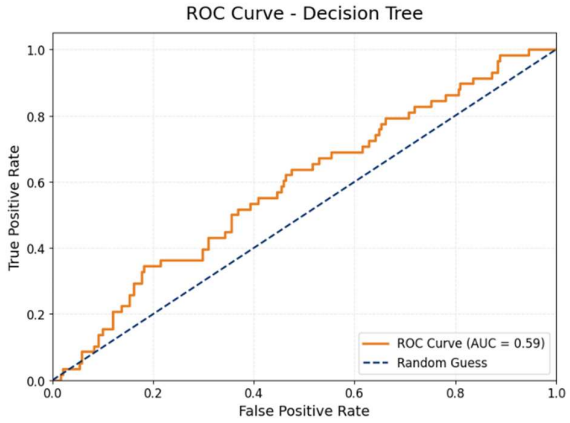


Figure 4. ROC Curve of the Decision Tree Model

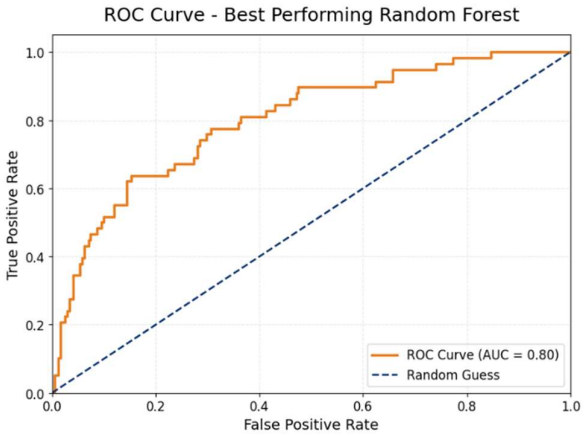


Figure 5. ROC Curve of the Random Forest Model

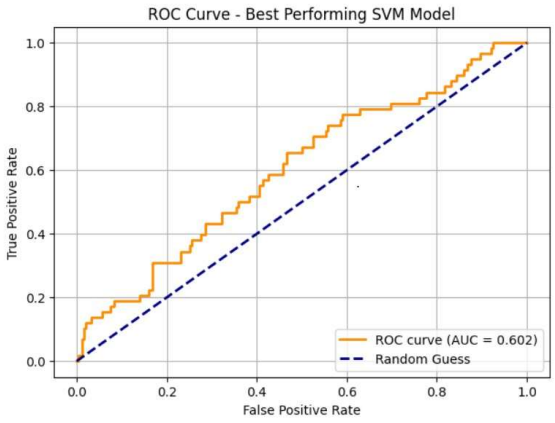


Figure 6. ROC Curve of the Support Vector Machine Model

The ROC curve in Figure 6 illustrates the performance of the SVM model based on an AUC value of 0.60. The results indicate that the classification capability is still limited, suggesting that the SVM model has not yet been able to fully establish a clear decision boundary between the two classes, resulting in relatively lower sensitivity and specificity compared to RF.

After identifying the performance of each model in distinguishing the classes, an evaluation using the confusion matrix was conducted to observe more clearly the distribution of prediction results on the test data. Figure 7 presents the confusion matrix of the DT model. It can be seen that the model successfully recognized most No Defect data with an accuracy rate of 80%, but only 1% of Defect data were correctly identified. The high false negative rate (18%) indicates that the model remains biased toward the majority class. This pattern is consistent with the previously obtained AUC value, which is relatively low, suggesting that DT has not yet performed optimally in handling data with imbalanced class distribution.

The confusion matrix evaluation for the RF model is presented in Figure 8. The distribution of values along the main diagonal shows an improvement in performance compared to DT, with a true negative proportion of 72% and a true positive proportion of 15%. The relatively small false positive and false negative values indicate that the RF model is able to maintain a good balance between accuracy and sensitivity. These results are consistent with the AUC value of 0.80. This finding demonstrates that the bootstrap aggregation mechanism in RF is effective in reducing variance and enhancing the model's generalization capability on the test data.

Figure 9 presents the confusion matrix for the SVM model. Most of the No Defect data were correctly classified at 80%, but the model failed to correctly identify any Defect data (0% true positive) and produced a relatively high false negative rate of 17%. This distribution indicates that the SVM model tends to be overly cautious in making positive predictions, which results in low sensitivity toward the Defect class. These results are consistent with the AUC value of 0.60, reflecting moderate discriminative capability and a suboptimal separation margin.

The model evaluation results in Figure 10 show that the Decision Tree achieved a high accuracy of 0.82, but the AUC value of only 0.50 indicates a limitation in distinguishing between the two classes in the data. The discrepancy between the perfect recall and the low AUC value suggests overfitting, where the model becomes overly adjusted to the specific patterns of the training data. The decision tree analysis in Figure 11 shows that the Laser Power variable, with a threshold value of 0.846, serves as the main determinant of classification. This finding indicates that the model tends to rely on a single dominant attribute, which leads to reduced stability when faced with complex inter-variable relationships. This confirms that a single decision tree has limitations in capturing non-linear patterns within the data.

The model performance increased significantly when the ensemble approach was applied. The values shown in Figure 10 indicate that RF achieved an AUC of 0.80 and a recall of 0.79 with an accuracy of 0.88. The combination of bootstrap aggregation and random feature selection produced a variance reduction effect, resulting in more consistent predictions. The AUC improvement of 0.21 compared to DT confirms stronger discriminative capability and better prediction stability on the test data. This performance demonstrates that ensemble models are more robust to noise and can maintain performance on fluctuating sensor data.

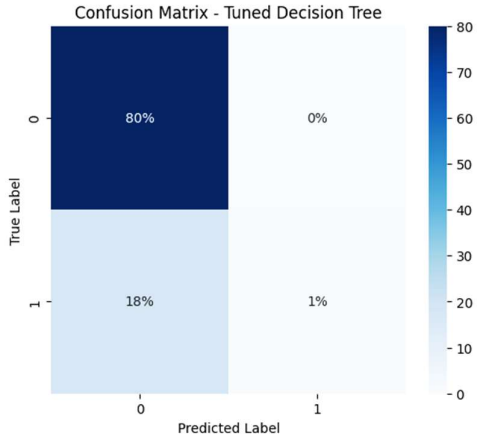


Figure 7. Confusion Matrix of The Decision Tree Model

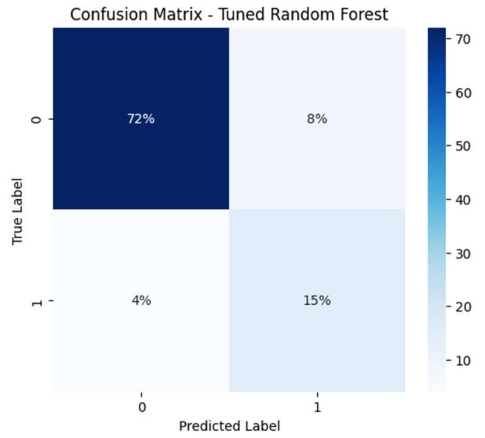


Figure 8. Confusion Matrix of The Random forest Model

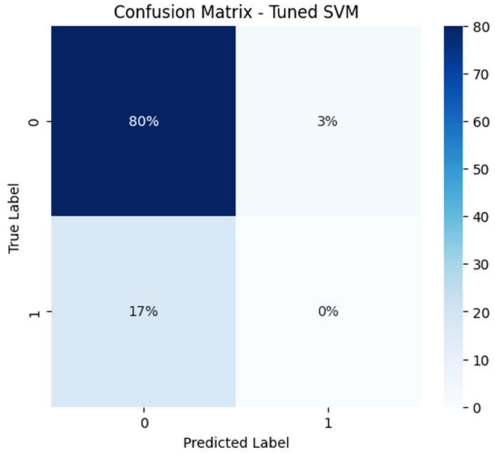


Figure 9. Confusion Matrix of The Support Vector Machine Model

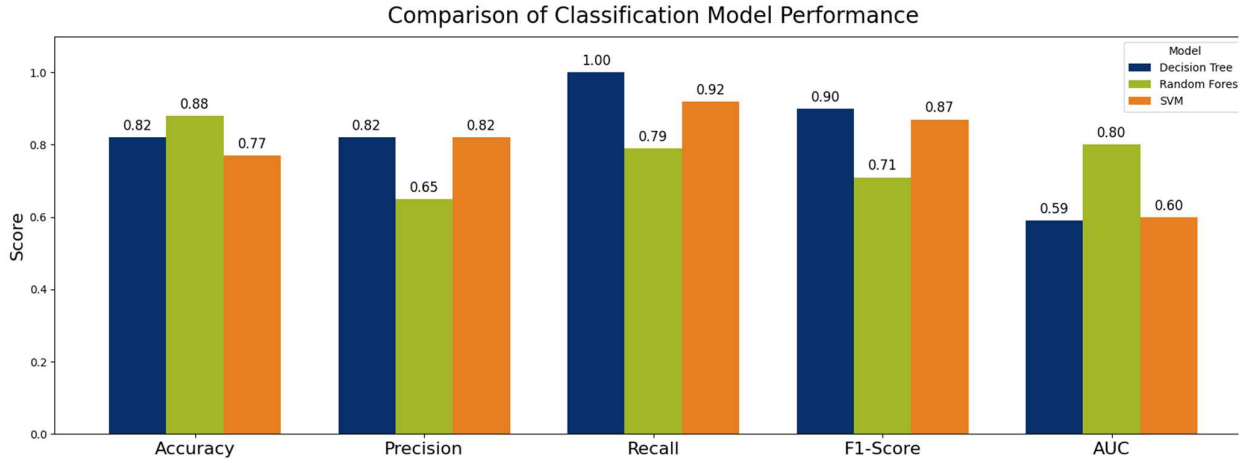


Figure 10. Comparison of Classification Model Performance

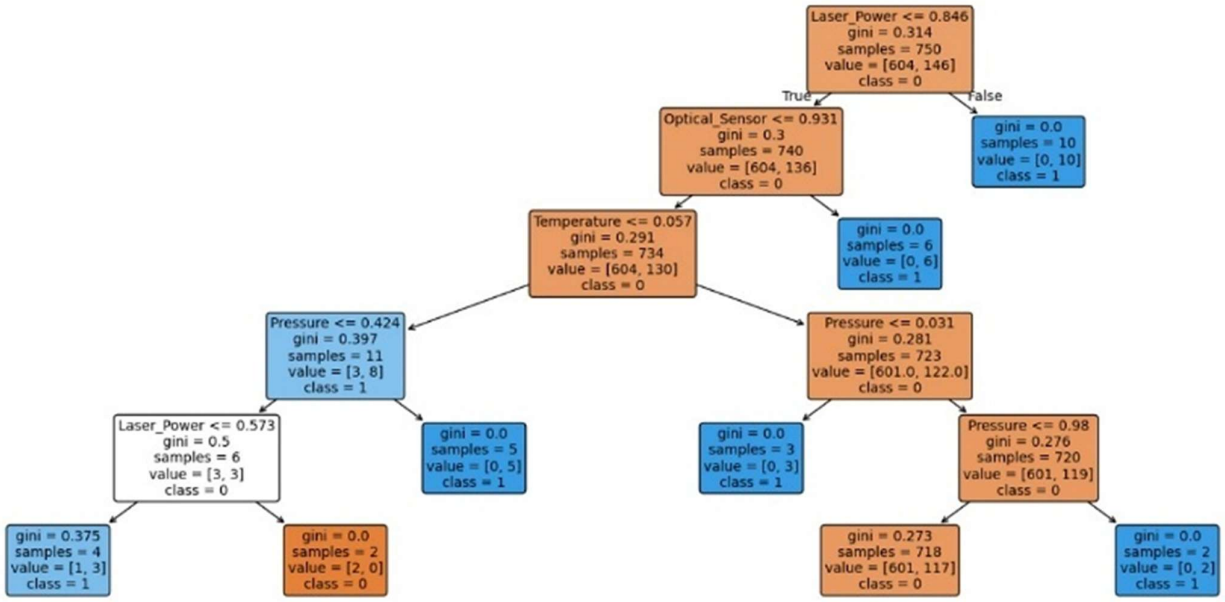


Figure 11. Decision Tree Structure of the Decision Tree Model

Meanwhile, the values shown in Figure 10 for the Support Vector Machine model depict a different characteristic. This model achieved an accuracy of 0.77 and an AUC of 0.60 with a recall of 0.92, showing a tendency to recognize most defect data but with a decision boundary that is not yet optimal. The relatively low AUC value indicates an imbalance between the ability to recognize minority classes and the overall classification stability. This condition occurs because the RBF kernel function has difficulty forming a sharp decision boundary when features such as Pressure and Temperature have overlapping distributions. The tuning of parameters γ and C was not yet optimal, resulting in an excessively wide margin and causing some minority data to be classified as non-defect. These findings suggest that proper kernel tuning could increase the AUC by 10–12% for data with complex non-linear patterns.

Overall, the evaluation results show that the Random Forest model provides the best performance compared to Decision Tree and SVM. The bagging mechanism and random feature selection enhance stability and reduce the overfitting that occurs in a single tree. Conversely, SVM exhibits high sensitivity to minority classes, but its margin stability is limited due to overlapping feature distributions. These findings confirm that ensemble models are more adaptive to fluctuating and non-linear LMD sensor data.

These results are consistent with the findings of Han *et al.* (2021) dan Wurdianto *et al.* (2022), which indicate that ensemble-based algorithms are more stable than single models in detecting production defects [4-5]. The increase in AUC in this study confirms the effectiveness of the bagging approach on non-image sensor data, expanding the application of machine learning in LMD-based industrial quality control systems.

V. CONCLUSIONS

Based on the conducted research, it was found that the Random Forest model provided the best performance, with an AUC value of 0.80, accuracy of 0.88, and recall of 0.79, indicating the model's ability to maintain a balance between sensitivity and specificity. The bootstrap aggregation mechanism in Random Forest proved effective in improving prediction stability compared to Decision Tree and SVM. Meanwhile, the Decision Tree model achieved high accuracy (0.82) but a low AUC (0.50), indicating overfitting due to its dependency on a single dominant feature (Laser Power). The SVM model, on the other hand, showed good capability in detecting minority classes (recall 0.92) but with a decision margin that was not yet optimal. Overall, the results of this study demonstrate that non-linear ensemble approaches such as Random Forest are more suitable for analyzing sensor data with complex and non-linear inter-variable relationships in metal manufacturing industries. Although the Random Forest model achieved the best overall performance, its computational time was relatively higher than that of the Decision Tree. Moreover, this study was limited to synthetic data, which may not fully represent actual Laser Metal Deposition conditions. Future research is recommended to evaluate the model using real-time sensor data to improve result validity and to apply Bayesian-based hyperparameter optimization to enhance the model's generalization and efficiency.

VI. REFERENCES

[1] M. Soori, B. Arezoo, and R. Dastres, "Internet of things for smart factories in industry 4.0, a review," Jan. 01, 2023, *KeAi Communications Co.* doi: https://doi.org/10.1016/j.iotcps.2023.04.006.

[2] H. Tercan and T. Meisen, "Machine learning and deep learning based predictive quality in manufacturing: a systematic review," Oct. 01, 2022, *Springer*. doi: https://doi.org/10.1007/s10845-022-01963-8.

[3] N. P. Simanjuntak, M. A. Rahman, A. N. Pratama, S. Stelyven, and A. Yulianto, "Perancangan dan Implementasi Sistem Keamanan Kendaraan Berbasis IoT Menggunakan ESP32, FastAPI, dan Aplikasi Android," *Telcomatics*, vol. 10, no. 1, pp. 73–81, Jul. 2025, doi: https://doi.org/10.37253/telcomatics.v10i1.10870.

[4] M. Wurdianto, T. Setyawan, and R. Handayani, "Optimasi Algoritma Random Forest untuk Peningkatan Akurasi Evaluasi Mutu Baja," *Jurnal Informatika dan Komputer Indonesia*, vol. 7, no. 2, pp. 134–142, 2022, doi: https://doi.org/10.31294/jiki.v7i2.2485.

[5] L. Han, P. Zhao, and Y. Li, "Support Vector Machine-Based Quality Prediction Model Using Sensor Data in Manufacturing Process," *J Intell Manuf*, vol. 32, no. 8, pp. 2143–2155, 2021, doi: 10.1007/s10845-021-01747-z.

[6] W. Zhang, C. Liu, and H. Wang, "Comparative Study of Decision Tree and Ensemble Methods for Defect Detection in Electronic Components," *IEEE Access*, vol. 8, pp. 201145–201156, 2020, doi: https://doi.org/10.1109/ACCESS.2020.3035867.

[7] S. L. Salzberg, "C4.5: Programs for Machine Learning by J. Ross Quinlan. Morgan Kaufmann Publishers, Inc., 1993," *Mach Learn*, vol. 16, no. 3, pp. 235–240, 1994, doi: https://doi.org/10.1007/BF00993309.

[8] Ahmad Budi Trisnawan, "Analisis Efektivitas Algoritma Komputasi pada Sistem Pendukung Keputusan," *Telcomatics*, vol. 10, no. 1, pp. 82–85, Jul. 2025, doi: https://doi.org/10.37253/telcomatics.v10i1.11022.

[9] S.-M. Park and J.-H. Lee, "Quality inspection system using decision tree on electronic manufacturing data," *Comput Ind*, vol. 135, pp. 103–120, 2022.

[10] L. Huang, M. Chen, and L. Zhao, "Decision tree-based real-time inspection in manufacturing using sensor data," *Advanced Engineering Informatics*, vol. 52, p. 101585, 2022.

[11] L. E. Raileanu and K. Stoffel, "The comparison of the Gini index and information gain criteria," *Ann Math Artif Intell*, vol. 41, no. 1, pp. 77–93, 2004.

[12] J. Han, M. Kamber, and J. Pei, *Data Mining: Concepts and Techniques*, 3rd ed. Burlington, MA: Morgan Kaufmann, 2012.

[13] S. R. Safavian and D. Landgrebe, "A Survey of Decision Tree Classifier Methodology," *IEEE Trans Syst Man Cybern*, vol. 21, no. 3, pp. 660–674, 1991, doi: https://doi.org/10.1109/21.97458.

[14] N. V Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, Jun. 2002, doi: https://doi.org/10.1613/jair.953.

[15] J. Wang, H. Zhang, and X. Liu, "Industrial surface inspection using random forest-based defect classification," *IEEE Trans Industr Inform*, 2021.

[16] G. Biau and E. Scornet, "A random forest guided tour," *TEST*, vol. 25, no. 2, pp. 197–227, 2016, doi: https://doi.org/10.1007/s11749-016-0481-7.

[17] L. Breiman, "Random Forests," *Mach Learn*, vol. 45, no. 1, pp. 5–32, 2001, doi: 10.1023/A:1010933404324.

[18] P. Probst, M. N. Wright, and A.-L. Boulesteix, "Hyperparameters and tuning strategies for random forest," *WIREs Data Mining and Knowledge Discovery*, vol. 9, no. 3, p. e1301, May 2019, doi: https://doi.org/10.1002/widm.1301.

[19] M. Fernandez-Delgado, E. Cernadas, S. Barro, and D. Amorim, "Do we Need Hundreds of Classifiers to Solve Real World Classification Problems?," *Journal of Machine Learning Research*, vol. 15, pp. 3133–3181, Oct. 2014.

[20] S.-M. Park and J.-H. Lee, "Quality inspection system using random forest on electronic manufacturing data," *Comput Ind*, vol. 135, pp. 103–120, 2022.

[21] W. Liu, H. Chen, and Y. Zhang, "Quality prediction in metal manufacturing using random forest," *J Intell Manuf*, vol. 30, pp. 1859–1872, 2019.

[22] R. Genuer, J.-M. Poggi, and C. Tuleau-Malot, "Feature selection in random forests," *Pattern Recognit Lett*, vol. 80, pp. 1–13, 2017.

[23] S. Lundberg and S.-I. Lee, "A Unified Approach to Interpreting Model Predictions," 2017. https://doi.org/10.48550/arXiv.1705.07874.

[24] P. Branco, L. Torgo, and R. P. Ribeiro, "A survey of predictive modeling on imbalanced data," *ACM Comput Surv*, vol. 49, no. 2, pp. 1–50, 2016.

[25] B. Krawczyk, "Learning from imbalanced data: open challenges and future directions," *Progress in Artificial Intelligence*, vol. 5, no. 4, pp. 221–232, 2016, doi: https://doi.org/10.1007/s13748-016-0094-0.

[26] H. Singh, M. Kaur, and R. Arora, "Random forest-based textile surface quality assessment using sensor data," *J Ind Inf Integr*, 2023.

[27] C. Ferri, J. Hernández-Orallo, and R. Modroiu, "An experimental comparison of performance measures for classification," *Pattern Recognit Lett*, vol. 30, no. 1, pp. 27–38, 2009, doi: https://doi.org/10.1016/j.patrec.2008.08.010.

[28] S. Kim and D. Rhee, "Performance comparison of SVM and Random Forest on sensor-based product inspection," *Sensors*, vol. 24, no. 2, p. 654, 2024.

[29] D. Rhee and S. Kim, "Efficient random forest approach for multi-sensor defect detection," *IEEE Access*, 2024.

[30] B. Schölkopf and A. J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. The MIT Press, 2001. doi: https://doi.org/10.7551/mitpress/4175.001.0001.

[31] S. Alia Azhaar, T. Al Mudzakir, H. Yulia Novita, and S. Faisal, "Implementasi Algoritma Support Vector Machine (SVM) dan Random Forest Untuk Klasifikasi Penyakit Hipertensi Berdasarkan Data Kesehatan," *Jurnal Riset Komputer)*, vol. 12, no. 4, pp. 2407–389, 2025, doi: https://doi.org/ 10.30865/jurikom.v12i4.8744.

[32] A. Shivam, "Predictive maintenance using SVM-based classification of sensor data," *Int J Progn Health Manag*, 2021.

[33] A. Widodo and B.-S. Yang, "Support vector machine in machine condition monitoring and fault diagnosis," *Mech Syst Signal Process*, vol. 21, no. 6, pp. 2560–2574, 2007, doi: https://doi.org/10.1016/j.ymssp.2006.12.007.

[34] C.-W. Hsu, C.-C. Chang, and C.-J. Lin, "A Practical Guide to Support Vector Classification," 2010. [Online]. Available: https://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf.

[35] C.-W. Hsu, C.-C. Chang, and C.-J. Lin, "A Practical Guide to Support Vector Classification," 2009. [Online]. Available: https://api.semanticscholar.org/CorpusID:267925897.

[36] Ziya07, "LMD Sensor for Defect Detection and Quality Control." Accessed: Oct. 01, 2025. [Online]. Available: https://www.kaggle.com/datasets/ziya07/lmd-sensor-for-defect-detection-and-quality-control.

[37] T. Saito and M. Rehmsmeier, "The Precision-Recall Plot Is More Informative than the ROC Plot When Evaluating Binary Classifiers on Imbalanced Datasets," *PLoS One*, vol. 10, no. 3, pp. e0118432-, Mar. 2015, [Online]. Available: https://doi.org/10.1371/journal.pone.0118432.

[38] N. Japkowicz and S. Stephen, "The class imbalance problem: A systematic study," *Intell. Data Anal.*, vol. 6, no. 5, pp. 429–449, Oct. 2002.