International Journal of Research in Engineering and Modern Technology (IJREMTE)
Volume 2, No 1, April 2025, Pages 8-15
ISSN 3089-4077

Resistor Demand Prediction Using Manual Gradient Boosting Implementation for Inventory Optimization

F.X. Wisnu Yudo Untoro[1,*]

[1] Department of Informatics, Universitas Wijaya Kusuma Surabaya, East Java, Indonesia; Email: wisnuyudo@uwks.ac.id
*Correspondence: wisnuyudo@uwks.ac.id.

Abstract—This study proposes and evaluates a manual implementation of the Gradient Boosting algorithm for monthly resistor demand forecasting to optimize inventory in electronic component stores. Efficient inventory management is critical in a competitive market to ensure product availability and minimize costs. While conventional forecasting methods often struggle with demand instability in diverse resistor SKUs, Gradient Boosting offers robust capabilities for handling complex, non-linear patterns. Our methodology involves training the model on a small, simulated historical dataset (5 unique resistor IDs), using past demand data as features. The model's performance is evaluated using Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE), which consistently decreased during 20 training iterations (final MAE 0.2529, RMSE 0.2938). The model successfully predicted demand for Month-4 and Month-5 using a sliding window strategy. These predictions have significant implications for reducing overstocking costs, preventing understocking, and optimizing purchasing decisions. However, the use of a very small simulated dataset is a major limitation, leading to overfitting and limiting the model's generalizability for real-world applications. This study primarily serves as a methodological illustration of the core principles of Gradient Boosting. Future work should focus on larger, real-world datasets and leveraging optimized libraries for enhanced accuracy and practical reliability.
Keywords— Demand Prediction; Resistor; Gradient Boosting; Inventory Optimization; Time Series

I. Introduction

Efficient inventory management forms the operational backbone for electronic component suppliers, particularly in the handling of resistors. In a rapidly evolving and competitive component market, the timely availability of resistors with precise values and tolerances is paramount for business continuity and meeting diverse customer demands [1].

Resistor overstocking can lead to inflated storage costs, risk of component obsolescence, and unproductive tied-up capital. Conversely, understocking can result in lost sales opportunities, inability to fulfill direct customer orders, and damage to the store's reputation [3]. Therefore, the ability to accurately predict resistor demand is crucial for optimizing inventory levels and maintaining operational balance, ensuring efficient capital turnover and minimizing losses [4], [5], [6].

Resistors, as indispensable fundamental elements in every electronic circuit, serve as a relevant case study for this research. As resistor vendors, stores face the challenge of managing thousands of *Stock-Keeping Units* (SKUs), which vary by value, tolerance, power rating, and other characteristics. Their demand patterns can exhibit significant instability depending on resistor values, tolerances, pricing, and broader electronics market trends [7], [8]. Precisely understanding and predicting this instability is vital to ensuring stable stock and preventing supply chain disruptions, thereby fulfilling all customer needs [9].

Despite the advancements in demand forecasting, a gap often exists in applying advanced machine learning techniques to specific electronic component inventory challenges, especially when detailed historical data might be limited or exhibit complex non-linear patterns.

To address the problem of demand instability and inventory management challenges, this research offers a solution in the form of integrating *machine learning* models into the demand forecasting process. Specifically, the *Gradient Boosting* algorithm, as the foundation of *Extreme Gradient Boosting* (XGBoost), is chosen for its superior ability to handle tabular data, model non-linear relationships, and provide built-in regularization features that prevent *overfitting* [10], [11], [12], [13], [14].

The main objective of this research is to develop and evaluate a manual implementation of a *Gradient Boosting* model capable of predicting monthly resistor demand. These accurate predictions are expected to serve as a guide for component vendors in making more efficient procurement and stock management decisions, reducing operational costs, and ultimately increasing customer satisfaction.

This research provides several important contributions: (1) Demonstration of Manual Implementation: Presents a manual implementation of *Gradient Boosting*, providing in-depth insights into how this *ensemble* algorithm works and its role in iteratively reducing prediction errors, (2) Application in Resistor Inventory Optimization: Applies the predictive model to a resistor inventory optimization scenario, demonstrating how demand predictions can directly support operational cost reduction and improved product availability, and (3) Analysis of Small Data Limitations: Highlights the challenges of *overfitting* and generalization limitations when dealing with very small datasets, providing important lessons for future research and applications.

II. Literature Review

The importance of accurate demand forecasting in inventory management has been widely recognized across various industries. Studies highlight that effective demand prediction is a cornerstone for optimizing inventory levels, leading to reduced holding costs and improved service levels. It is emphasized that robust forecasting underpins efficient inventory operations [2], while the negative repercussions of both overstocking and understocking, linking them directly to financial inefficiencies and customer dissatisfaction, are delved into [3]. The economic benefits of precise inventory management, ensuring efficient capital turnover and minimizing losses, are further underscored by research [4], [5]. Furthermore, the application of advanced forecasting methods in retail contexts, similar to an electronic component store, has shown promise in improving sales predictions [6].

Specifically concerning components like resistors, their demand can be inherently complex and exhibit significant instability due to various factors. The intricacies of time series analysis are essential for understanding such fluctuating demand patterns [7], while the role of supply chain reliability and seasonality as key performance indicators for demand forecasting in inventory is also emphasized [8]. The criticality of predicting these instabilities to ensure supply chain stability and avoid disruptions is also a recurring theme in the literature [9].

The integration of *machine learning* technologies, particularly advanced *ensemble* methods, offers substantial potential to overcome the limitations of conventional forecasting methods. *Extreme Gradient Boosting* (XGBoost), a highly optimized implementation of *Gradient Boosting*, has demonstrated superior performance in various prediction tasks involving tabular data [10]. The foundational principles of *Gradient Boosting*, which sequentially build models to correct errors from previous ones, allow it to handle heterogeneous data and identify complex non-linear relationships with high accuracy [11], [12], [13]. The selection of *Gradient Boosting*, and by extension the principles of XGBoost, is justified by its proven ability to achieve high accuracy with structured data, its capacity to model non-linear relationships between variables, and its built-in regularization features that effectively prevent *overfitting* [11], [12], [14], [10]. These capabilities are highly relevant given the often complex and unstable characteristics of component sales data.

III. Materials and Methods

3.1. Research Design

This research adopts a quantitative approach with a focus on predictive modeling. This research design follows the Input-Process-Output (IPO) flow, where historical resistor demand data serves as input, the development and evaluation of the *Gradient Boosting* model as the process, and resistor demand prediction for inventory optimization as the output. For a clearer visualization, Figure 1 presents the block diagram of this research design.

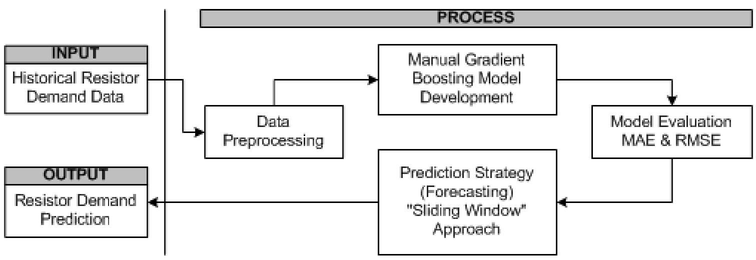

Figure 1. Block Diagram of Research Design

Figure 1 illustrates the Input-Process-Output (IPO) flow of the research methodology, detailing the data input, the Gradient Boosting model development and evaluation, and the resulting demand predictions for inventory optimization.

The objective is to develop and evaluate a *machine learning* model capable of predicting monthly resistor demand for future periods (Month-4 and Month-5). These predictions are expected to serve as a basis for supporting resistor inventory optimization. The *Gradient Boosting* algorithm was chosen for its ability to handle non-linear relationships and produce accurate predictions on tabular data [11], [12], [13]. This manual implementation serves as a demonstration of the fundamental principles underlying optimized algorithms like XGBoost [10].

2. Data Source

The dataset used in this study is simulated historical resistor demand data, consisting of 5 unique entries (`ID`). These entries represent common types of resistors, such as E12, E24, E48, E96, and E192 series, each with standard resistance values and tolerances [15], [16], [17], [18]. Each entry includes the following attributes:

- `ID`: A unique identifier for each resistor type (e.g., a specific resistor value within an E series).
- `Value`: A numerical attribute representing the resistance value of the resistor (e.g., in Ohms) relevant to the specified E series.
- `Tolerance`: A numerical attribute representing the tolerance level of the resistor (e.g., 0.10, 0.05) according to the E series standard.
- `Month-3`: Actual resistor demand three months prior to the target month.
- `Month-2`: Actual resistor demand two months prior to the target month.
- `Month-1`: Actual resistor demand one month prior to the target month.
- `Demand`: Actual resistor demand in the target month (the month after `Month-1`), which serves as the dependent variable or target to be predicted.

To provide a concrete overview of the data structure used, this dataset structure is represented as follows:

```
data = {
    'ID': [1, 2, 3, 4, 5],
    'Value': [10, 22, 47, 100, 220],
    'Tolerance': [0.10, 0.05, 0.02, 0.01, 0.10],
    'Month-3': [10, 11, 12, 14, 7],
    'Month-2': [9, 10, 11, 13, 8],
    'Month-1': [8, 9, 13, 12, 6],
    'Demand': [9, 10, 14, 12, 7]
}
```

Although this dataset is small, it is used specifically for demonstration purposes and for an in-depth understanding of how the *Gradient Boosting* algorithm works iteratively. In real-world applications, datasets would include a wider variety of E12, E24, E48, E96, and E192 series resistors, as well as longer historical periods. Larger and more representative datasets are highly necessary to achieve reliable and generalizable prediction accuracy.

3. Data Preprocessing

The data preprocessing stage involves identifying features and targets, as well as preparing data for model training:

- Feature Identification (Independent Variables): The features used to train the model are historical resistor demand data: `Month-1`, `Month-2`, and `Month-3`. Although `Value` and `Tolerance` are available in the dataset and are relevant to resistor characteristics (including E series), they are not directly used as input for the decision trees in this *Gradient Boosting* implementation, but they remain part of the context for each resistor `ID`.
- Target Identification (Dependent Variable): The `Demand` variable is set as the prediction target, representing the actual resistor demand in the month after `Month-1`.
- Data Splitting: Due to the very small size of the dataset, the entire dataset is used for model training to maximize the learning of existing patterns. In large-scale research, splitting data into *training*, *validation*, and *test sets* is standard practice.

4. Model Development

The predictive model is built using a manual implementation of the *Gradient Boosting* Regression algorithm. This implementation aims to demonstrate the core principles of *Gradient Boosting*, which form the basis for more advanced and optimized algorithms like *Extreme Gradient Boosting* (XGBoost). This model development process includes the following steps:

- Basic Algorithm: *Gradient Boosting* is an *ensemble* technique that builds models sequentially, where each new model attempts to correct errors made by the previous models.
- Weak Learner: Each individual model in this *ensemble* is a decision tree (`DecisionTreeRegressor`) from the `scikit-learn` library.
- Model Parameters:
    1. `n_estimators`: The number of decision trees built in the *ensemble* is set to 20. More trees lead to a more complex model but also a higher risk of *overfitting*.

International Journal of Research in Engineering and Modern Technology (IJREMTE)
Volume 2, No 1, April 2025, pages 8-15
ISSN 3089-4077

2. `learning_rate`: The learning rate is set at 0.1. This parameter controls how much each new tree contributes to the overall prediction. Smaller values require more trees but can result in a more robust model [14].
3. `max_depth`: The maximum depth of each decision tree is limited to 3. This depth restriction helps prevent each tree from becoming too complex and reduces the risk of *overfitting*.
4. `random_state`: Set to 42 to ensure reproducibility of tree training results.

- Iterative Training Process:
  1. Initial Prediction Initialization: The initial prediction (`F_train`) is initialized as the average of all target values (`Demand`) in the training dataset. The average formula is calculated as:

$$\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i \quad (1)$$

where:
$\bar{y}$: average target value.
n : number of training data samples.
y_i : individual target value

  2. Boosting Iterations: For each of the `n_estimators` iterations:
     - Residual Calculation: Residual (error) is calculated as the difference between the actual `Demand` value and the current `F_train` prediction. This residual serves as the negative gradient of the *loss* function.
     - Tree Training on Residuals: A new decision tree is trained to predict these residuals, not the original target. This tree learns from the remaining errors.
     - Prediction Update: The prediction from the newly trained tree is multiplied by the `learning_rate` and added to `F_train`. This process gradually "pushes" the model's prediction closer to the actual value.

5. Model Evaluation

Model evaluation is performed to measure the model's performance and accuracy on the training data. The evaluation metrics used are:

- *Mean Absolute Error* (MAE): Measures the average absolute difference between predicted and actual values. MAE provides a direct indication of the average prediction error in the same units as the target variable [19], [20].

$$MAE = \frac{1}{n}\sum \left| y_i - \hat{y}_i \right| \quad (2)$$

where:
n : number of data samples
y_i : i-th actual value (observation)
$\hat{y}_i$: i-th predicted value

- *Root Mean Squared Error* (RMSE): Measures the square root of the average of the squared differences between predicted and actual values. RMSE is more sensitive to large errors (*outliers*) than MAE [19], [20].

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n} \left(y_i - \hat{y}_i \right)^2} \quad (3)$$

where:
n : number of data samples
y_i : i-th actual value (observation)

International Journal of Research in Engineering and Modern Technology (IJREMTE)
Volume 2, No 1, April 2025, pages 8-15
ISSN 3089-4077

$\hat{y}_i$: i-th predicted value

- Iterative Evaluation Visualization: Line graphs are used to visualize changes in MAE and RMSE values at each training iteration. These graphs help monitor model convergence and identify potential *overfitting* (if validation error starts to increase).
- Decision Tree Structure: The structure of each trained decision tree is displayed to provide insights into how the model makes decisions and which features are most influential in predicting residuals.

6. Prediction Strategy (Forecasting)

The trained model is used to predict demand for future months (Month-4 and Month-5) using a "sliding window" strategy. Accurate demand predictions are crucial for informed decision-making in resistor inventory optimization.

- Month-4 Demand Prediction: To predict `Demand` for Month-4, input features are prepared by shifting historical data:
  - `Month-3` (for Month-4) is taken from the original `Month-2`.
  - `Month-2` (for Month-4) is taken from the original `Month-1`.
  - `Month-1` (for Month-4) is taken from the original `Demand` (which was the training target).

  The model then runs this input data through all trained trees to generate `Predicted_Demand_Month_4`.

- Month-5 Demand Prediction: There are two scenarios for predicting `Demand` for Month-5:
  There are two scenarios for predicting `Demand` for Month-5:

  1. Using Month-4 Prediction: If actual Month-4 data is not yet available, the `Predicted_Demand_Month_4` generated by the model will be used as the latest historical input (`new Month-1`) to predict Month-5. This approach is susceptible to *error propagation*, where errors from `Predicted_Demand_Month_4` can affect the accuracy of `Predicted_Demand_Month_5` [21], [22].

  2. Using Actual Month-4 Data (If Available): If actual `Demand` data for Month-4 is available, that actual data will be used as the latest historical input (`new Month-1`) for Month-5 prediction. This is the more recommended method as it reduces the risk of error accumulation and yields more accurate predictions.

7. Development Environment

This research was implemented using the Python programming language. Key *libraries* used include:

- NumPy: For numerical operations and arrays.
- Pandas: For data manipulation and analysis (DataFrames).
- Scikit-learn: Provides `DecisionTreeRegressor` implementation and `export_text` function.
- Matplotlib: For visualizing model evaluation graphs.

IV. Results and Discussion

1. Model Training Results

The *Gradient Boosting* model training was conducted over 20 iterations, with each iteration adding a new decision tree that attempts to reduce the residuals from previous predictions. Table 1 shows the final prediction results of the model on the resistor demand training data.

Table 1. Prediction Results on Training Data

ID	Demand	Predicted_Training	Residual_Training
1	9	9.170207	-0.170207
2	10	10.048631	-0.048631
3	14	13.562324	0.437676
4	12	11.805477	0.194523
5	7	7.413361	-0.413361

`Predicted_Training` represents the resistor demand values predicted by the model after 20 iterations on the same data used for training. `Residual_Training` shows the difference between the actual resistor demand values and the

International Journal of Research in Engineering and Modern Technology (IJREMTE)
Volume 2, No 1, April 2025, pages 8-15
ISSN 3089-4077

model's predictions. Residual values close to zero indicate that the model has learned well to adapt to the patterns in the available training data.

The model's performance progression during the training process can be observed through MAE and RMSE values per iteration. In Iteration 1, the model started with an MAE of 1.8720 and an RMSE of 2.1749. Over the iterations, both error metrics showed a consistent decrease, reaching an MAE of 0.2529 and an RMSE of 0.2938 at Iteration 20. This decrease indicates that each added decision tree successfully reduced the overall prediction error of the model.

The visual progression of model performance is also shown in Fig. 2.

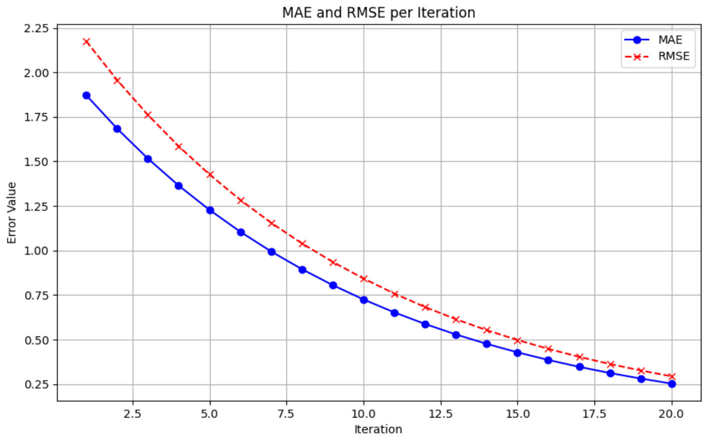

Figure 2. MAE and RMSE Graph per Training Iteration

Figure 2 shows that the MAE (blue line) and RMSE (red line) values consistently decrease as the number of iterations increases. RMSE is consistently higher than MAE, which is expected as RMSE gives more weight to larger errors (due to squaring). This stable downward trend indicates that the *Gradient Boosting* algorithm successfully converged and improved the model's accuracy on the training data.

To understand how the model makes decisions and identifies patterns in the data, it is important to review the structure of the decision tree formed in each iteration. As an illustration, here is the textual representation of the decision tree structure formed in Iteration 1:

```
|--- Bulan-2 <= 10.50
|   |--- Bulan-3 <= 8.50
|   |   |--- value: [-3.40]
|   |--- Bulan-3 >  8.50
|   |   |--- Bulan-2 <= 9.50
|   |   |   |--- value: [-1.40]
|   |   |--- Bulan-2 >  9.50
|   |   |   |--- value: [-0.40]
|--- Bulan-2 >  10.50
|   |--- Bulan-3 <= 13.00
|   |   |--- value: [3.60]
|   |--- Bulan-3 >  13.00
|   |   |--- value: [1.60]
```

This tree structure shows how the model makes decisions based on `Month-2` and `Month-3` values to predict residuals. The `value` at each leaf of the tree represents the predicted residual contribution by that tree. Over iterations, the `value` at the tree leaves will change (decrease) as subsequent trees focus on progressively smaller residuals.

2. Month-4 Demand Prediction Results

After the model was trained, it was used to predict resistor demand for Month-4. The input for this prediction was prepared by shifting historical data, where the actual `Demand` values from the training dataset served as the new `Month-1` for Month-4 prediction. Table 2 presents the resistor demand prediction results for Month-4.

International Journal of Research in Engineering and Modern Technology (IJREMTE)
Volume 2, No 1, April 2025, pages 8-15
ISSN 3089-4077

Table 2. Month-4 Demand Prediction Results

ID	Value	Tolerance	Month-3 for_Month_4	Month-2 for_Month_4	Month-1 for_Month_4	Predicted_Demand_Month_4
1	10	0.10	9	8	9	9.170207
2	22	0.05	10	9	10	10.048631
3	47	0.02	11	13	14	13.562324
4	100	0.01	13	12	12	11.805477
5	220	0.10	8	6	7	7.413361

`Predicted_Demand_Month_4` shows the estimated **resistor** demand for each `ID` based on patterns learned from historical data. For example, for `ID=1`, the model predicts a resistor demand of 9.170207 for Month-4. It should be noted that, due to the very limited size of the training dataset, these predicted values tend to be very similar to the predictions on the training data, indicating the model's tendency to memorize existing patterns.

3. Month-5 Demand Prediction Results

Resistor demand prediction for Month-5 is explored in two scenarios:

3.1. Scenario 1: Using Month-4 Prediction

In this scenario, `Predicted_Demand_Month_4` is used as the latest historical input (the new `Month-1`) to predict Month-5. Table 3 shows the prediction results under this scenario.

Table 3. Month-5 Demand Prediction Results (Using Month-4 Prediction)

ID	Value	Tolerance	Month-3 for_Month_5	Month-2 for_Month_5	Month-1 for_Month_5	Predicted_Demand_Month_5
1	10	0.10	8	9	9.170207	9.170207
2	22	0.05	9	10	10.048631	10.048631
3	47	0.02	13	14	13.562324	13.562324
4	100	0.01	12	12	11.805477	11.805477
5	220	0.10	6	7	7.413361	7.413361

These prediction results show how the model continues to estimate **resistor** demand for further periods. However, this approach is susceptible to *error propagation*, where errors from `Predicted_Demand_Month_4` can affect the accuracy of `Predicted_Demand_Month_5` [21], [22].

3.2. Scenario 2: Using Actual Month-4 Data (Simulated)

To demonstrate the impact of using actual data, a simulation of actual Month-4 data is used as input to predict Month-5. Table 4 presents the prediction results under this scenario.

Table 4. Month-5 Demand Prediction Results (Using Actual Month-4 Simulated Data)

ID	Value	Tolerance	Month-3 for_Month_5	Month-2 for_Month_5	Month-1 for_Month_5	Predicted_Demand_Month_5
1	10	0.10	8	9	9.500000	9.170207
2	22	0.05	9	10	10.200000	10.048631
3	47	0.02	13	14	13.800000	13.562324
4	100	0.01	12	12	12.100000	11.805477
5	220	0.10	6	7	7.300000	7.413361

The comparison between Table 3 and Table 4 illustrates the importance of using actual data if available. Using actual Month-4 data (simulated) as the latest historical input for Month-5 prediction is expected to yield more accurate predictions compared to using previous Month-4 predictions, as it reduces error propagation.

4. General Discussion and Implications

This research successfully demonstrates a manual implementation of the *Gradient Boosting* algorithm for time series prediction. The model shows the ability to learn patterns from historical data and iteratively reduce errors on training data, as indicated by the consistent decrease in MAE and RMSE. This manual implementation serves as a conceptual foundation for understanding how more advanced and optimized *Extreme Gradient Boosting* (XGBoost) algorithms work [10], [11], [12], [13].

The resistor demand predictions generated by this model have significant implications for resistor inventory optimization. By knowing the estimated demand for Month-4 and Month-5, companies can:

- Reduce Storage Costs: Minimize *overstocking* of resistors, which reduces storage costs, risk of damage, or obsolescence.
- Improve Product Availability: Prevent *understocking* of resistors, ensuring product availability when needed, and avoiding lost sales and customer dissatisfaction.

- Optimize Purchasing and Production Decisions: Enable inventory managers to plan raw material purchases and resistor production schedules more timely and efficiently, thereby reducing operational costs and increasing supply chain responsiveness.

However, it is important to acknowledge a significant limitation of this research due to the very small dataset size (only 5 rows of data). With limited data, the model tends to *overfit*, meaning it memorizes specific patterns in the training data rather than learning generalizable relationships [11], [14]. This is evident from the similarity between predictions on the training data and predictions for Month-4 and Month-5. Therefore, the results obtained in this study primarily serve as a methodological illustration and basic concept validation of the *Gradient Boosting* algorithm rather than a reliable prediction for real-world applications.

V. Conclusion

This research successfully developed and demonstrated a manual implementation of the *Gradient Boosting* algorithm to predict monthly resistor demand. The model showed the ability to learn patterns from historical data from the preceding three months, with consistent evaluation performance showing a decrease in MAE and RMSE values at each training iteration. Demand predictions for Month-4 and Month-5 were also successfully generated using a "sliding window" strategy.

These prediction results have significant positive implications for resistor inventory optimization. With more accurate demand estimates, electronic component supply stores can:

- Reduce storage costs due to *overstocking*.
- Improve product availability and avoid lost sales due to *understocking*.
- Optimize purchasing and production planning.

Nevertheless, it must be acknowledged that this research has substantial limitations, namely the use of a very small simulated dataset (only 5 rows). This dataset size causes the model to tend to *overfit*, limiting its generalizability, and making the prediction results serve more as a methodological illustration than as a reliable prediction tool for real-world applications.

For future development, it is highly recommended to:

- Use larger and more varied resistor demand datasets to train the model, thereby improving generalization.
- Implement more robust cross-validation techniques and data splitting (*training*, *validation*, *test set*).
- Utilize optimized *Gradient Boosting libraries* (e.g., XGBoost, LightGBM) for better scalability and performance.
- Perform systematic *hyperparameter tuning* to achieve optimal accuracy.

Conflict of Interest

The authors declare no conflict of interest.

References

[1] D. K. Vaka, "Integrating Inventory Management and Distribution: a Holistic Supply Chain Strategy," *International Journal of Managing Value and Supply Chains (IJMVSC)*, vol. 15, no. 2, p. 13, Jun. 2024.

[2] V. Pasupuleti, B. Thuraka, C. S. Kodete, and S. Malisetty, "Enhancing Supply Chain Agility and Sustainability through Machine Learning: Optimization Techniques for Logistics and Inventory Management," *Logistics*, vol. 8, no. 3, p. 73, 2024, doi: 10.3390/logistics8030073.

[3] G. K. Mweshi, "Effects of Overstocking and Stockouts on the Manufacturing Sector," *International Journal of Advances in Engineering and Management (IJAEM)*, vol. 4, no. 9, pp. 1054–1064, Sep. 2022, doi: 10.35629/5252-040910541064.

[4] R. Winurputra and D. E. Ratnawati, "Peramalan Penjualan Produk Menggunakan Extreme Gradient Boosting (XGBOOST) Dan Kerangka Kerja Crisp-Dm untuk Pengoptimalan Manajemen Persediaan (Studi Kasus: Ub Mart)," *JTIIK*, vol. 12, no. 2, pp. 417–428, Apr. 2025, doi: 10.25126/jtiik.2025129451.

[5] S. B. Dalimunthe, R. Ginting, and S. Sinulingga, "The Implementation Of Machine Learning In Demand Forecasting : A Review Of Method Used In Demand Forecasting With Machine Learning," *j.sist.teknik.industri.*, vol. 25, no. 1, pp. 41–49, Jan. 2023, doi: https://doi.org/10.32734/jsti.v25i1.9290.

[6] P. Dankorpho, "Sales Forecasting for Retail Business Using XGBoost Algorithm," *Journal of Computer Science and Technology Studies*, vol. 6, no. 2, pp. 136–141, Jun. 2024, doi: https://doi.org/10.32996/jcsts.2024.6.2.15.

[7] B. Auffarth, *Machine Learning for Time-Series with Python: Forecast, Predict, and Detect Anomalies with State-of-the-Art Machine Learning Methods*. Packt Publishing Ltd, 2021.

[8] Y. Tadayonrad and A. B. Ndiaye, "A New Key Performance Indicator Model for Demand Forecasting in Inventory Management Considering Supply Chain Reliability and Seasonality," *Supply Chain Analytics*, vol. 3, p. 100026, 2023, doi: https://doi.org/10.1016/j.sca.2023.100026.

[9] K. Kanike, “Factors Disrupting Supply Chain Management in Manufacturing Industries,” *Journal of Supply Chain Management Science*, vol. 4, no. 1-2, pp. 1–24, 2023, doi: https://doi.org/10.18757/jscms.2023.6986.

[10] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2016, pp. 785–794, doi: 10.1145/2939672.2939785.

[11] J. H. Friedman, “Greedy Function Approximation: A Gradient Boosting Machine,” *Ann. Stat.*, vol. 29, no. 5, pp. 1189–1232, 2001.

[12] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed. New York, NY, USA: Springer, 2009.

[13] Z. He, D. Lin, T. Lau, and M. Wu, “Gradient Boosting Machine: A Survey,” arXiv:1908.06951 [cs.LG], 2019, doi: 10.48550/arXiv.1908.06951.

[14] A. Géron, *Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. O’Reilly Media, Inc., 2022.

[15] “Standard Resistor Values,” Logwell.com. [Online]. Available: https://www.logwell.com/tech/components/resistor_values.html. [Accessed: Jun. 6, 2025].

[16] “Standard Resistor Values E24,” Asenergi.com. [Online]. Available: https://asenergi.com/pdf/resistors/standard-resistor-values-e24.pdf. [Accessed: Jun. 6, 2025].

[17] “Standard Resistor Series Values: E3, E6, E12, E24, E48, E96, E192,” ElectricalVolt.com. [Online]. Available: https://www.electricalvolt.com/standard-resistor-series-values-e3-e6-e12-e24-e48-e96/#h-resistors-e-series-values-and-tolerance-band. [Accessed: Jun. 6, 2025].

[18] “Resistor Chart: Comprehensive Guide to Resistor Values, E-Series, and Color Codes,” *Wevolver*. [Online]. Available: https://www.wevolver.com/article/resistor-chart-comprehensive-guide-to-resistor-values-e-series-and-color-codes [Accessed: Jun. 7, 2025].

[19] A. V. Tatachar, “Comparative Assessment of Regression Models Based On Model Evaluation Metrics,” *International Research Journal of Engineering and Technology (IRJET)*, vol. 8, no. 09, pp. 853–860, Sep. 2021.

[20] D. Chicco, M. J. Warrens, and G. Jurman, “The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation,” *PeerJ Computer Science.*, vol. 7, p. e623, 2021.

[21] R. J. Hyndman and G. Athanasopoulos, *Forecasting: Principles and Practice*, 3rd ed. Melbourne: OTexts, 2021. [Online]. Available: https://otexts.com/fpp3/.

[22] S. Ben Taieb, G. Bontempi, A. F. Atiya, and A. Sorjamaa, “A review and comparison of strategies for multi-step ahead time series forecasting based on the NN5 forecasting competition,” *Neurocomputing*, vol. 73, no. 10-12, pp. 3428–3438, 2010, doi: 10.1016/j.neucom.2010.01.032