RADAR Performance Prediction for Autonomous Driving Using Machine Learning and Process Data Analysis

Mingyu Cha
Industrial Engineering Dept.
Inha University
Incheon, Korea
Email: mincha2@inha.edu

***Abstract*—This paper employs a multioutput regression technique to predict the performance of RADAR sensors used in autonomous driving. The study aims to enhance prediction accuracy by utilizing real-world RADAR feature process data obtained from the manufacturing process. The analysis is conducted through the core algorithm, the catboost regressor, along with two additional standard models and their corresponding multioutput versions. The research findings demonstrate that the modified NRMSE(MNRMSE) of the multioutput regressor models are up to 0.006 lower than that of individual target models, indicating superior performance of the multioutput approach. The primary algorithm, the multioutput catboost regressor, achieved a MNRMSE of 1.9307. This result validates that multioutput models effectively capture correlations between outputs, reduce redundant data learning, and mitigate overfitting. The findings suggest that multioutput regression models, particularly catboost, can be effectively applied not only to RADAR but also to other industrial process datasets to improve yield. Future research will aim to integrate more comprehensive process data and further refine the model to maximize prediction accuracy. Additionally, applying these models to various industrial fields is expected to provide valuable insights for improving production yield and reducing costs.**

I. Introduction

A. Background and Motivation

Demand for various manufacturing products, including semiconductors, continues to grow, and this trend is expected to continue going forward [1]. This increase has led to a significant increase in the emphasis on product quality and performance across industries, driven by advances in technologies such as artificial intelligence (AI). As a result, the complexity of manufacturing processes has also been enhanced to meet this demand [2]. However, companies are facing ongoing challenges in reducing production costs. To address these issues, many businesses are focusing on improving production yields. On average, defective products cause significant financial losses every year [3]. By improving yield, companies can improve their competitive advantage and increase profitability [4]. This highlights the important role of yield optimization across various manufacturing industries, where utilizes process data to improve production outcomes has become a central strategy.

This paper focuses on radio detection and ranging(RADAR) sensor data, a key technology that emits radio waves and measures the time taken for their return after reflecting off objects, thereby determining their distance and velocity. The role of RADAR technology has grown significantly, especially in sectors such as autonomous driving, robotics, and military applications. These fields exhibit rapid technological advancements, with RADAR playing a pivotal role in autonomous vehicles. In this domain, reliable detection of vehicle speed, distance, and other critical factors are essential for ensuring safety—a paramount concern in the industry. Thus, this paper seeks to develop a predictive model for RADAR sensor performance based on process data. The dataset employed originates from actual RADAR production data provided by a corporate research laboratory, specifically related to RADAR systems utilized in autonomous vehicles [5].

B. Contributions

- **Handling the correlation between multiple outputs** This model effectively handles the correlation between multiple outputs, allowing it to make more accurate predictions by considering the correlation among variable features.
- **Mitigation of Data Leakage** By using ordered boosting in catboost, the model reduces the risk of data leakage, especially when handling multioutput tasks where multiple outputs are predicted simultaneously.
- **The Real World Data** This model is based on data from the actual manufacturing process. Therefore, applying it in real industrial settings can make more realistic results.

C. Organization

The rest of the paper is organized as follows. Sec. II reviews the existing research in the field and explains the objectives of this study. Sec. III discusses the theory and features of the multioutput catboost regressor, which is the core algorithm of this paper. Sec. IV evaluates the performance of the proposed algorithm, providing comparisons with other benchmarks. Lastly, Sec. V concludes the paper based on the findings and suggests directions for future research.

II. Preliminaries

A. Related Work

Catboost regressor, being a tree-based algorithm, demonstrates strengths not only with categorical data but also with numerical data [6]. It excels in predicting complex numerical data with high accuracy and efficiency. While models like xgboost and catboost are commonly used for predicting numerical data, catboost often outperforms prediction accuracy [7]. Catboost is particularly effective in predicting numerical features within real-world process data and handles data processing tasks with ease [8]. Additionally, Catboost's ordered boosting technique helps reduce overfitting, thus enhancing the model's generalization capabilities [9]. Catboost also supports multioutput regression, making it especially advantageous for RADAR sensor process data with multiple parameters and their interdependencies.In models that attempted to mitigate input and output noises through multioutput regression, significant results were achieved [10].

B. The objective of this paper

This paper aims to develop a model that predicts the performance of RADAR using real-world process data. The objective is to create a predictive model based on process data that minimizes normalized root mean squared error(NRMSE) by considering the importance of features. Ultimately, the derived performance prediction model is expected to be applied across various industries, contributing to improved yield while addressing economic and environmental challenges.

III. Algorithm Design

Modified NRMSE (MNRMSE) is used as the evaluation metric for the proposed algorithm. NRMSE is one of the metrics used to assess prediction performance in regression models, representing the root means squared error (RMSE) normalized to the scale of the data. The reason for using NRMSE in this model is that it allows for fair and consistent comparison of prediction errors across various variables among several evaluation metrics. Standard metrics like RMSE or mean absolute error (MAE) are sensitive to the scale and range of each variable, making them less suitable when dealing with variables of different units. In situations involving multioutput prediction, the scales of the individual variables differ, which makes it challenging to evaluate errors uniformly across variables. Normalizing the error adjusts it to the range or variability of the predicted values, enabling more accurate comparisons across different models.

The formula for NRMSE can be expressed as,

$$\text{NRMSE} = \frac{\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}}{\max(y) - \min(y)} \tag{1}$$

where n is the number of data points, y_i is the actual value, $\hat{y}_i$ is the predicted value, $\max(y)$ and $\min(y)$ are the maximum and minimum actual values.

This paper utilizes MNRMSE instead of the general NRMSE. MNRMSE applies an additional 20% weight to the first to seventh evaluation criteria, and the sum of all NRMSE values are computed.The MNRMSE can be expressed as,

$$\text{MNRMSE} = \frac{\sum_{k=1}^{7} 1.2 \cdot \text{NRMSE}_k + \sum_{k=8}^{n} \text{NRMSE}_k}{n} \tag{2}$$

where n is the total number of evaluation criteria, NRMSE_k is the NRMSE value for the k-th criterion, The first 7 criteria are given an additional weight of 20% to emphasize their importance in RADAR performance. The reason for employing this method is that, as shown in III, multiple parameters exist, and it was not appropriate to evaluate each error separately for predicting RADAR performance. Therefore, weights are added to the first to seventh evaluation criteria, which significantly impact performance in process data. Before explaining the core algorithm proposed in this paper, the multioutput catboost regression model, we will describe the basic concept of the catboost model. The catboost regressor uses ordered boosting, which is based on Gradient Boosting. Gradient Boosting is an algorithm that sequentially combines several weak learners to improve prediction performance. It improves the model iteratively by learning the residuals from the previous model at each step. The update of Gradient Boosting is performed as,

$$F_m(x) = F_{m-1}(x) + \eta \cdot \sum_{i=1}^{n} \frac{\partial L(y, F(x))}{\partial F(x_i)} \tag{3}$$

where $F_m(x)$ is the model at the m-th boosting iteration, η is the learning rate, $\frac{\partial L(y,F(x))}{\partial F(x_i)}$ is the gradient of the loss function. Catboost algorithm uses ordered boosting, which is based on Gradient Boosting. While traditional boosting algorithms learn from all the data at once, ordered boosting modifies this by creating predictions only from previously observed data in each step instead of the model's previous predictions. This approach ensures that the actual values of the current data point are not included in the training data, thus preventing overfitting and data leakage. The formula for ordered boosting can be expressed as,

$$F_m(x_i) = F_{m-1}(x_i) + \eta \cdot \sum_{j=1}^{i-1} \frac{\partial L(y_j, F_{m-1}(x_j))}{\partial F(x_j)} \tag{4}$$

where x_i is the i-th data point, $F_m(x_i)$ is the model at the m-th boosting iteration for the i-th data point, η is the learning rate, $\frac{\partial L(y_j,F_{m-1}(x_j))}{\partial F(x_j)}$ is the gradient of the loss function.

The core algorithm of this study, the multioutput catboost regressor, is an extension of the catboost model, capable of predicting multioutput simultaneously instead of a single output. This model is helpful in multioutput regression problems, as it allows learning the correlations between multiple outputs. Rather than learning individual models for each output, the multioutput catboost regressor is designed to learn multiple outputs together, enabling the model to capture shared patterns. The model can be structured to train separate models for each target or to build a model that simultaneously predicts multiple outputs, reflecting the interrelationships between the outputs.

The model predicts one output at a time in traditional single-output regression. The general formulation for a single-output model can be expressed as,

$$y_i = f(X) + \epsilon_i \tag{5}$$

where y_i is the single output, X represents the input data, $f(X)$ is the model function, ϵ_i denotes the prediction error.

On the other hand, the multioutput regression model predicts multiple outputs simultaneously, capturing the correlations between them. The formulation for a multioutput model can be expressed as,

$$\mathbf{Y} = f(X) + \epsilon \tag{6}$$

where $\mathbf{Y} = [y_1, y_2, \ldots, y_k]$ represents multiple outputs, X represents the input data (shared across the outputs), $f(X)$ is the model function, and $\epsilon = [\epsilon_1, \epsilon_2, \ldots, \epsilon_k]$ denotes the prediction errors for each output.

When applying multioutput, the loss function based on NRMSE is calculated for each output y_1, y_2, …, y_k and the total loss is defined by averaging these NRMSE values. The multioutput NRMSE loss function is expressed as,

$$L(\mathbf{y}, \mathbf{F}(x)) = \frac{1}{k} \sum_{j=1}^{k} \frac{\sqrt{\frac{1}{n} \sum_{i=1}^{n} (y_{i,j} - \hat{y}_{i,j})^2}}{\max(y_j) - \min(y_j)} \tag{7}$$

where k is the number of outputs, $y_{i,j}$ is the actual j-th target value of the i-th data point, $\hat{y}_{i,j}$ is the predicted value for the j-th target, $\max(y_j)$ and $\min(y_j)$ are the maximum and minimum values for the j-th target.

The gradient update of multioutput catboost for each output can be expressed as,

$$F_{m,j}(x) = F_{m-1,j}(x) + \eta \cdot \sum_{i=1}^{n} \frac{\partial L(\mathbf{y}, \mathbf{F}(x))}{\partial F_j(x_i)} \tag{8}$$

where $F_{m,j}(x)$ is the model for the j-th target at the m-th boosting iteration, $\frac{\partial L(\mathbf{y}, \mathbf{F}(x))}{\partial F_j(x_i)}$ is the gradient of the NRMSE loss function for the j-th target.

The multioutput regressor predicts multiple outputs simultaneously using a single model, learning while considering the relationships between the variables. In catboost, the internal algorithm is extended to handle multidimensional outputs. The multioutput loss function is the sum of the losses for each output.

In conclusion, the critical difference between the traditional catboost regressor and the multioutput catboost regressor is that the former predicts a single output. In contrast, the latter predicts multiple outputs, accounting for their correlations during learning. This results in higher computational efficiency than processing multiple outputs individually and reduces training and prediction times. This advantage is beneficial in environments where computational cost is critical, such as large-scale RADAR manufacturing datasets.

In this paper, optuna is utilized to optimize the model's hyperparameters. Optuna is a framework employed for hyperparameter optimization that has played a crucial role in improving the performance of multioutput catboost. Specifically,

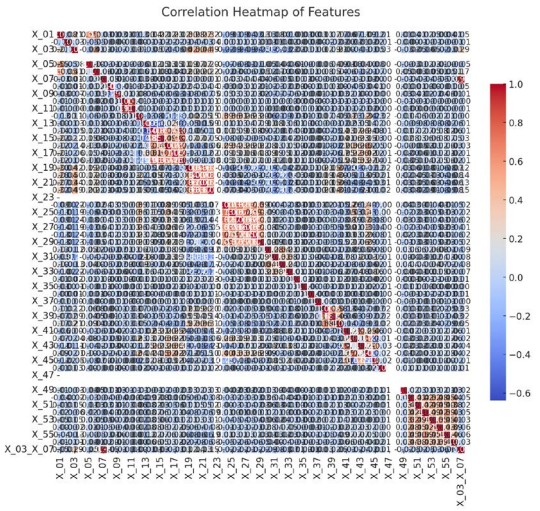

Fig. 1. Correlation Heatmap of Features

Optuna's Bayesian Optimization is utilized, which determines the following hyperparameter values to try based on the performance of previously tested values.Optuna optimizes the model's hyperparameters, focusing on the learning rate, maximum tree depth, number of estimators, and L2 regularization coefficient. This process aims to minimize the MNRMSE, allowing faster and more efficient model optimization than manual tuning.

In this study, the RADAR dataset involves numerous features. As illustrated in Fig. 1, a correlation heatmap is used to visualize the relationships between these features, aiding in the identification of highly correlated or redundant variables. By addressing multi-collinearity, the study employs Principal Component Analysis (PCA) to reduce highly correlated features while preserving key information. This step assists in streamlining the dataset and avoiding multi-collinearity issues, contributing to more robust model performance.

To identify and remove insignificant features, including categorical variables, the select_features function of catboost and shapley additive explanations (SHAP) values is used to evaluate and select important features. This process allows the model to retain only the most relevant features, improving prediction accuracy. Fig. 2 shows the top 15 feature importances according to catboost, demonstrating which features had the greatest impact on the model's predictions. The core principle of the select_features function is to iteratively retrain the model while removing features and assessing their impact on model performance. At each step, a feature is removed, the model is retrained, and features with minimal impact on performance are eliminated. This process is repeated until only the most significant features remain.

SHAP values are based on game theory and quantify the contribution of each feature to the model's predictions. SHAP calculates how each feature contributes across all possible feature combinations, allowing the average contribution of each feature to be determined. The formula for calculating

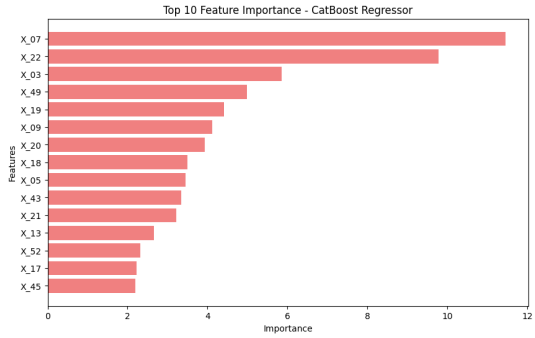

Fig. 2. Top 15 CATBOOST Feature Importance

SHAP values can be expressed as,

$$\phi_i = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(|N| - |S| - 1)!}{|N|!} \left(v(S \cup \{i\}) - v(S)\right) \quad (9)$$

where ϕ_i is the SHAP value for feature i, N is the set of all features, S is a subset of features excluding i, $v(S)$ is the value function, which represents the prediction made by the model with the feature set S. SHAP values allow for a clear interpretation of how much each feature contributes to the model's predictions. Features with lower average SHAP values are considered less important and can be removed. By doing so, features that do not significantly impact performance can be eliminated, thereby reducing the complexity of the model.

IV. Performance Evaluation

A. Experimental Setup

The dataset used in this study is based on manufacturing process data collected to predict the performance of autonomous driving RADAR systems. It consists of 39,607 samples, and each sample includes several process variables. Key variables include the pressure applied during PCB assembly, the weight and surface area of heat dissipation materials in the RADAR system, the insertion depth of screws, and the dimensions of the radome at antenna installation locations. These variables are structured to evaluate the performance of the RADAR system based on each measurement. The performance evaluation items for the process data are composed of 56 categories, as shown in Table I.

While individual variables in dataset measure specific performance factors, derived variables are essential to better capture the correlations between variables. Derived variables reflect the interactions between the base variables and contribute to improving the predictive performance of the model. This allows the model to predict RADAR performance.

Table II presents the specific derived variables used in the dataset. For example, the derived variable "weight/area" better reflects heat management by considering both the weight and surface area of the heat dissipation material, which plays a crucial role in the RADAR system's temperature management. The weight-to-area ratio represents heat dissipation efficiency, making it useful for evaluating the system's performance under heat-related stress during the manufacturing process. In addition, the derived variable "total pressing force" sums the stepwise pressure values applied during PCB assembly, clearly showing the impact of pressure on performance during assembly. These derived variables reflect significant characteristics of the data and provide insights that cannot be captured using only the base variables. Based on the domain knowledge of RADAR, several derived variables are generated from these evaluation criteria. Additionally, categorical variables, such as pass/fail results that do not influence the learning process and features with insignificant correlations, are dropped. Furthermore, the RADAR's performance is determined according to the acceptance criteria outlined in Table III, classifying products as acceptable or defective.

NO	Description
1~2	Stepwise pressing force during PCB assembly (Step 1, 2)
3	Weight of thermal material 1 [g]
4	Pass/fail result of first inspection (0/1)
5~6	Stepwise pressing force during PCB assembly (Step 3, 4)
7~9	Surface area of thermal materials 1, 2, 3 [cm²]
10~11	Weight of thermal materials 2, 3
12	Reference coordinate of connector position
13	Height difference between antenna pads [cm]
14~18	Positions of antenna pads 1, 2, 3, 4, 5
19~22	Insertion depth of screws 1, 2, 3, 4
23	Pass/fail result of second inspection
24~29	Pin dimensions of connectors 1, 2, 3, 4, 5, 6
30~33	Insertion depth of screws 1, 2, 3, 4
34~37	Rotation speed [RPM] during screw fastening 1, 2, 3, 4
38~40	Dimensions of housing PCB mounting parts 1, 2, 3
41~44	Radome dimensions at antenna locations 1, 2, 3, 4
45	Radome inclination at antenna section
46	Required amount of sealant bond
47	Pass/fail result of third inspection
48	Pass/fail result of fourth inspection
49	Waiting time before Cal procedure
50~56	Solder amount at SMT locations of RF sections 1, 2, 3, 4, 5, 6, 7

TABLE I
RADAR Sensor Process Evaluation Parameters

Derived Variable	Description
X_{03}/X_07	Weight/Area
$X_{01} + X_{02} + X_{05} + X_{06}$	Total pressing force
$\max(X_{41} \ldots X_{44})$	Radome dimension difference
$X_7 + X_8 + X_9$	Total area
$X_3/(X_{19} + X_{20} + X_{21} + X_{22})$	Weight/Screw depth
$X_{12}/(X_{24}, X_{25})$	coordinates and pin

TABLE II
Derived Variables for Feature Engineering

B. Benchmarks

The following benchmarks are adopted to evaluate and compare the performance of the proposed algorithm.

1) Random Forest (RF): RF is an ensemble learning technique that improves predictive performance by combining multiple decision trees using the bagging method. The prediction of each tree is defined as,

$$T_i(x) = f(x, D_i), \quad (10)$$

where $T_i(x)$ is the prediction from the i-th tree, x represents the input feature values, D_i refers to the random sample of data used to train the i-th tree. The final prediction is computed

NO	Description	Acceptance Criteria
1	Antenna Gain Average (Angle 1)	0.2˜2
2	Antenna 1 Gain Deviation	0.2˜2.1
3	Antenna 2 Gain Deviation	0.2˜2.1
4	Average Signal-to-Noise Ratio	7˜19
5	Antenna Gain Average (Angle 2)	22˜36.5
6	Signal-to-Noise Ratio (Angle 1)	-19.2˜19
7	Antenna Gain Average (Angle 3)	2.4˜4
8	Signal-to-Noise Ratio (Angle 2)	-29.2˜-24
9	Signal-to-Noise Ratio (Angle 3)	-29.2˜-24
10	Signal-to-Noise Ratio (Angle 4)	-30.6˜-20
11	Antenna Gain Average (Angle 4)	19.6˜26.6
12	Signal-to-Noise Ratio (Angle 5)	-29.2˜-24
13	Signal-to-Noise Ratio (Angle 6)	-29.2˜-24
14	Signal-to-Noise Ratio (Angle 7)	-29.2˜-24

TABLE III
RADAR PERFORMANCE CRITERIA

by averaging the predictions from decision trees, which can be expressed as,

$$\hat{y} = \frac{1}{T}\sum_{t=1}^{T} f_t(x), \tag{11}$$

where T is the number of trees, $f_t(x)$ is the prediction from the t-th tree. RF trains each tree using random subsets of the training data and random feature subsets, leading to diverse learning outcomes for each tree. This process reduces the correlation between the trees, thereby preventing overfitting and improving the stability of predictions. However, since it does not account for the correlations between outputs, it demonstrates limited performance in multioutput problems such as the RADAR performance prediction used in this study.

2) Multioutput Random Forest (Multi RF): The Multi RF model, an extension of the traditional RF, has the advantage of predicting multiple outputs simultaneously. However, since each tree handles outputs independently, it fails to capture the correlations between them. As a result, it also shows limited performance in addressing multioutput problems, such as the RADAR performance prediction in this study, leading to suboptimal performance.

3) Xgboost (XGB): XGB is an improved version of the Gradient Boosting algorithm. Traditional Gradient Boosting lacks regularization techniques to control model complexity. In contrast, XGB incorporates Lasso (L1) and Ridge (L2) regularization to control complexity and prevent overfitting, thus improving performance. XGB operates by learning from the residuals at each stage, progressively refining the prediction model. The fundamental loss function of XGB can be expressed as,

$$L(y, \hat{y}) = \sum_{i=1}^{n} \ell(y_i, \hat{y}_i) + \sum_{t=1}^{T} \Omega(f_t) \tag{12}$$

where $L(y, \hat{y})$ represents the loss function between the actual values y and the predicted values $\hat{y}$, and $\ell(y_i, \hat{y}_i)$ is the loss function (e.g., NRMSE). The regularization term $\Omega(f_t)$ can be defined as,

$$\Omega(f_t) = \gamma T + \frac{1}{2}\lambda ||w||^2 \tag{13}$$

where γT is a penalty for the number of leaf nodes in the tree, $\frac{1}{2}\lambda||w||^2$ is the L2 regularization term on the weights w. This regularization term controls model complexity and prevents overfitting. XGB improves the prediction iteratively by adding the output of each tree to the previous predictions:

$$\hat{y}_i^{(m)} = \hat{y}_i^{(m-1)} + f_m(x_i) \tag{14}$$

where $\hat{y}_i^{(m)}$ is the prediction at the m-th boosting iteration, and $f_m(x_i)$ is the prediction model learned from the m-th tree. Through this iterative process, the model's performance is continuously refined. XGB controls model complexity through regularization and prevents overfitting, thereby providing better predictive performance. However, XGB independently learns each output, which means it does not capture the correlations between the outputs. While XGB reduces residual errors by adding new trees at each stage, this process is carried out only for a single output at a time. Consequently, in multioutput problems, it fails to learn the interactions between outputs and does not account for complex relationships among them. This limitation results in reduced performance in the case of the RADAR performance prediction model, as it cannot effectively reflect the relationships between the multiple outputs.

4) Multioutput XGBoost (Multi XGB): The Multi XGB model is an extension of the original XGB, offering the advantage of predicting multiple outputs simultaneously. By combining XGB's robust performance with multioutput handling, this model eliminates the need to learn each output separately, allowing for efficient batch prediction. However, Multi XGB still does not explicitly learn the correlations between outputs and processes each target independently. This independent structure limits the model's performance when variables need to be considered interdependently. Such a limitation can lead to suboptimal performance, particularly in multioutput problems like RADAR performance prediction, where interactions and correlations between outputs are crucial.As a result, while Multi XGB excels in efficiently handling multiple outputs, its inability to capture relationships between outputs can restrict its predictive performance in scenarios where such relationships play a significant role.

C. Results

In this study, a total of six models are used to evaluate performance. These models are divided into two groups: models with and without the application of multioutput regression, each comprising RF, XGB, and CAT models. By comparing the models equipped with multioutput capabilities to those that handled outputs individually, the impact of predicting multiple outputs simultaneously is assessed in contrast to individual predictions. This approach enables a comprehensive evaluation

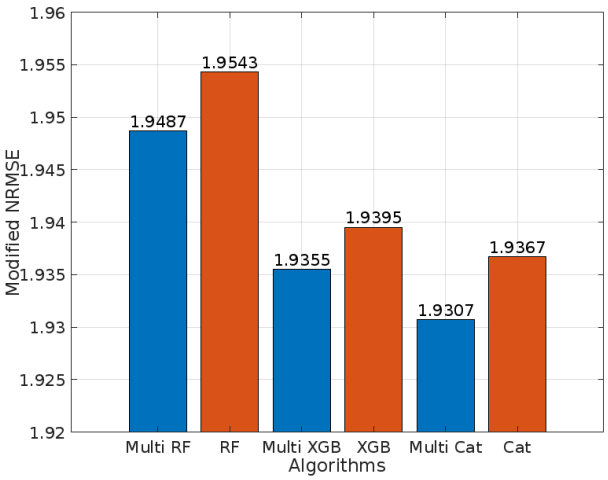

Fig. 3. MNRMSE Comparison of Models with Values

of how multioutput regression improves predictive accuracy across various model architectures.

As shown in Fig. 3, MNRMSE values for each model are 1.9487 for Multi RF, 1.9543 for RF, 1.9355 for Multi XGB, 1.9395 for XGB, 1.9307 for Multi Cat, and 1.9367 for Cat. The models with multioutput regression show MNRMSE values up to 0.006 smaller than their corresponding single-output models. There are several reasons for this performance improvement. First, multioutput regression effectively captures potential correlations between outputs by learning them simultaneously. Single-output models predict each target independently, thus ignoring the interactions between targets. However, Multioutput regression captures these relationships, leading to improved predictive performance, which is reflected in the smaller NRMSE values. Another key factor contributing to this performance improvement is the reduction of data redundancy. Instead of training separate models for each output, multioutput regression processes the shared information across multiple outputs, reducing redundant data learning. By streamlining the learning process and integrating diverse information and features, the model can avoid overfitting and achieve better generalization. This reduction in data redundancy contributes significantly to lower the MNRMSE values.

Among the RF, XGB, and CAT models, CAT consistently demonstrates superior performance. This can be attributed to differences in the algorithm's methods. RF, which employs a bagging approach, trains several independent decision trees but likely struggles to capture the correlations between targets in the RADAR dataset. Both XGB and CAT utilize the Gradient Boosting algorithm; however, catboost's use of ordered boosting helps prevent data leakage, which contributed to its more pronounced performance improvement.

V. Concluding Remark

The purpose of this study is to apply multioutput machine learning models to predict and analyze RADAR defects using actual process data, ultimately maximizing yield. The analysis of various regression models demonstrates that multi-output regression effectively reduces errors by learning the correlations between outputs. In advanced technologies like RADAR, which feature diverse characteristics, the benefits of multioutput regression are particularly pronounced. This is because multioutput regression captures the relationships between targets more effectively, reduces redundant data learning, and integrates various information, preventing overfitting. These advantages are most evident in catboost, as its algorithm structure is well-suited for handling correlated outputs. Additionally, the use of the ordered boosting technique, which prevents data leakage, further contributes to catboost's superior performance in RADAR performance prediction tasks. As shown in fig. 3, the MNRMSE of the multioutput catboost regressor is 1.9307, which is 0.0236 smaller compared to other models, underscoring the importance of effectively learning target relationships. Future research will focus on incorporating more comprehensive process data and improving model accuracy through hyperparameter optimization. Moreover, applying these models across various industries is recommended to ultimately enhance production yields and reduce costs.

References

[1] A. Sekiguchi, "The requirement for a bright future in semiconductor device manufacturing is… innovation, collaboration and care for the environment," in *Proc. IEEE Electron Devices Technology & Manufacturing Conference (EDTM)*, Oita, Japan, March 2022, pp. 6–7.

[2] R. Busch, M. Wahl, P. Czerner, and B. Choubey, "Yield prediction with machine learning and parameter limits in semiconductor production," in *Proc. IEEE International Symposium on Semiconductor Manufacturing (ISSM)*, Tokyo, Japan, December 2022, pp. 1–4.

[3] J. H. Park, H. Chung, K. H. Kim, J. J. Kim, and C. Lee, "The impact of technological capability on financial performance in the semiconductor industry," *Sustainability*, vol. 13, no. 2, p. 489, 'January' 2021.

[4] Y. Lee and Y. Roh, "An expandable yield prediction framework using explainable artificial intelligence for semiconductor manufacturing," *Applied Sciences*, vol. 13, no. 4, p. 2660, February 2023.

[5] LG AI Research, "Smart factory product quality status classification ai online hackathon," Online dataset, LG AI Research, 2023, dacon. Available at: https://dacon.io/competitions/official/236055/data.

[6] S. Jaiswal and P. Gupta, "Ensemble approach: Xgboost, catboost, and lightgbm for diabetes mellitus risk prediction," in *Proc. IEEE Second International Conference on Computer Science, Engineering and Applications (ICCSEA)*, Gunupur, India, September 2022, pp. 1–6.

[7] D. Wang, X. Xu, X. Xia, and H. Jia, "Interactive 3d vase design based on gradient boosting decision trees," *Algorithms*, vol. 17, no. 9, September 2024. [Online]. Available: https://www.mdpi.com/1999-4893/17/9/407

[8] Q. Zhou, Y. Guo, K. Xu, B. Chai, G. Li, K. Wang, and Y. Dong, "Research on the prediction algorithm of aero engine lubricating oil consumption based on multi-feature information fusion," *Applied Intelligence*, pp. 1–31, September 2024.

[9] A. Sysoev, "On factors selection in catboost models construction," in *Proc. IEEE International Conference on Control Systems, Mathematical Modeling, Automation and Energy Efficiency (SUMMA)*, Lipetsk, Russian Federation, November 2023, pp. 338–340.

[10] X. Li, Y. Wang, Z. Zhang, R. Hong, Z. Li, and M. Wang, "Rmor-aion: Robust multioutput regression by simultaneously alleviating input and output noises," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 3, pp. 1351–1364, 'April' 2020.