Journal of Information System Research (JOSH)
Volume 6, No. 1, Oktober 2024, pp 566−574
ISSN 2686-228X (media online)
https://ejurnal.seminar-id.com/index.php/josh/
DOI 10.47065/josh.v6i1.6095

Robust Fan Actuator Prediction in Smart Greenhouses Using Machine Learning: A Comparative Analysis of Ensemble and Linear Models

Gregorius Airlangga

Engineering Faculty, Information System Study Program, Atma Jaya Catholic University of Indonesia, Jakarta
Jl. Jend. Sudirman No.51 5, RT.004/RW.4, Karet Semanggi, Setiabudi District, South Jakarta City, Special Capital Region of Jakarta, Indonesia
Email: gregorius.airlangga@atmajaya.ac.id
Correspondence Author Email: gregorius.airlangga@atmajaya.ac.id
Submitted: **18/10/2024;** Accepted: **29/10/2024;** Published: **31/10/2024**

Abstract−The increasing demand for sustainable agriculture has driven the development of smart greenhouses equipped with automated systems for climate control. A critical component of these systems is the fan actuator, which regulates airflow and stabilizes the internal climate. This study explores the use of machine learning models for predicting the activation status of fan actuators based on environmental data collected from a smart greenhouse. We evaluate several machine learning models, including Support Vector Machine (SVM), Random Forest, Gradient Boosting, XGBoost, and Logistic Regression, under real-world conditions simulated by adding noise and label corruption to the dataset. The dataset was augmented and balanced using the Synthetic Minority Over-sampling Technique (SMOTE) to address class imbalances. Results indicate that ensemble methods, particularly XGBoost and Random Forest, outperform simpler models in terms of accuracy, precision, recall, and F1 score. XGBoost achieved the highest accuracy at 94.47%, while Random Forest followed closely with 94.29%. The study demonstrates that these models are robust to data imperfections and can be effectively employed for real-time fan actuator control. However, further validation is needed to generalize the findings to different greenhouse environments. The research highlights the potential of machine learning models to improve operational efficiency in smart farming, offering insights into how these technologies can support more sustainable agricultural practices.

Keywords: Smart Greenhouse; Machine Learning; Fan Actuator Prediction; XGBoost; Ensemble Methods

1. INTRODUCTION

The growing demand for sustainable agricultural practices has led to significant advancements in smart farming, particularly in controlled-environment systems such as smart greenhouses [1]–[3]. These greenhouses rely on sensors and actuators to regulate critical environmental factors like temperature, humidity, and soil moisture, enabling farmers to optimize crop growth and resource use efficiently [4]–[6]. A key component in these systems is the fan actuator, which regulates airflow to maintain a stable internal climate essential for optimal plant health and productivity [7]. However, despite the potential of smart greenhouses, many current control systems rely on static, rule-based mechanisms [8]. These systems activate actuators based on pre-defined thresholds, which often fail to adapt effectively to real-time environmental changes or account for the unpredictability of sensor data [9]. In particular, sensor data can be noisy or incomplete due to malfunctions, communication errors, or external disturbances [10]. This limitation highlights the need for adaptive control methods that can dynamically respond to evolving conditions. Machine learning models offer a promising solution by predicting actuator behavior based on historical and real-time data, enabling more flexible and accurate climate control [11].

Machine learning has been widely applied to various aspects of smart agriculture, including irrigation and lighting control. For instance, [12] used Random Forest models to optimize irrigation schedules based on soil moisture and weather forecasts, improving water efficiency. Similarly, [13] applied Support Vector Machine (SVM) models to control greenhouse lighting, achieving significant energy savings while maintaining optimal light levels for plant growth. In another relevant study, [14] used deep learning techniques to predict the growth of lettuce in a greenhouse environment by analyzing temperature and light data. While these applications demonstrate the effectiveness of machine learning in optimizing specific aspects of smart agriculture, the control of fan actuators crucial for regulating temperature remains largely unexplored in the literature [15]–[17].

Moreover, most existing studies assume that the data used to train machine learning models is clean and structured [18]–[20]. In reality, data collected from smart greenhouses is often subject to noise, missing values, or inconsistencies due to sensor malfunctions or transmission errors [21]. Models trained on ideal datasets may perform well in controlled environments but struggle when deployed in real-world settings where data imperfections are common [22]. Thus, it is essential to evaluate how machine learning models perform when faced with noisy and incomplete data, as this better reflects the operational conditions in commercial greenhouses [23]–[25]. This study aims to address these gaps by focusing on fan actuator prediction in smart greenhouses. It evaluates the performance of machine learning models using a dataset that has been augmented and intentionally corrupted to simulate the real-world challenges of noisy and imperfect data. Several machine learning algorithms: SVM, Random Forest, Gradient Boosting, XGBoost, and Logistic Regression are compared to identify the most effective model for predicting fan actuator status in a dynamic greenhouse environment.

The contributions of this research are threefold. First, it addresses the underexplored area of fan actuator control in greenhouses, which plays a critical role in maintaining temperature stability a key factor in crop health. Second, the study introduces a novel approach by deliberately augmenting and corrupting the dataset, providing a

Journal of Information System Research (JOSH)
Volume 6, No. 1, Oktober 2024, pp 566−574
ISSN 2686-228X (media online)
https://ejurnal.seminar-id.com/index.php/josh/
DOI 10.47065/josh.v6i1.6095

more realistic evaluation of model performance under practical conditions. This ensures that the developed models are robust enough to handle noisy and incomplete data, making them more applicable in real-world settings. Finally, the comparative analysis of multiple machine learning models provides valuable insights into the strengths and limitations of each approach, offering guidance for future research and practical applications in smart agriculture. The remainder of this article is organized as follows. The next section outlines the materials and methods used in this study, including a description of the dataset, data preprocessing steps, data augmentation techniques, and the machine learning models applied. The results section presents the performance metrics of each model, focusing on accuracy, precision, recall, and F1 score. This is followed by a discussion that interprets the results in the context of real-world greenhouse operations and highlights the practical implications of the findings. Finally, the conclusion summarizes the key contributions, acknowledges the limitations of the study, and suggests directions for future research.

2. RESEARCH METHODOLOGY

This section details the materials and methods employed in this study, including an in-depth explanation of the dataset, preprocessing techniques, data augmentation methods, and the machine learning models applied to predict fan actuator status in a smart greenhouse as presented in figure 1.

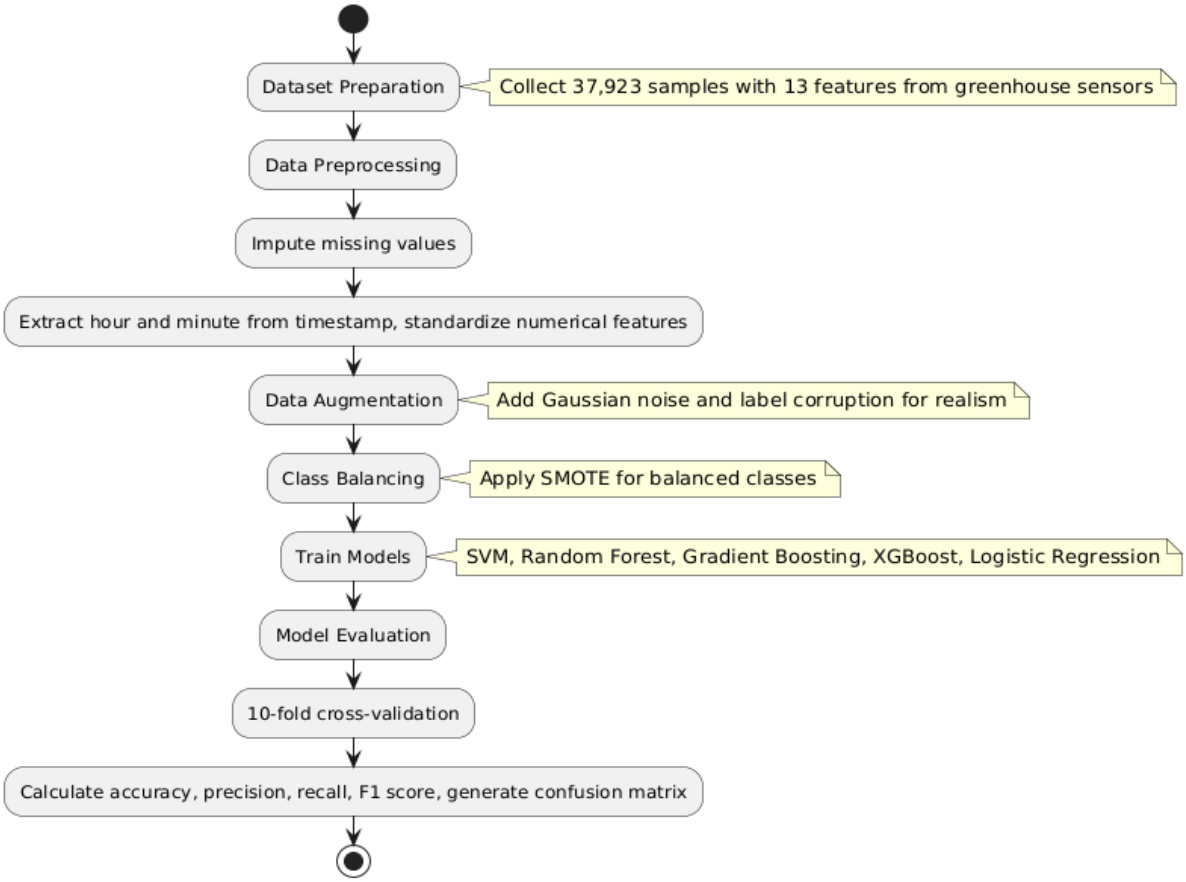

Figure 1. Research Methodology

2.1 Dataset Preparation

The dataset used in this study was obtained from a smart greenhouse equipped with advanced sensors for environmental monitoring [26]. The dataset comprised 37,923 samples with 13 features representing various environmental conditions. These features included temperature (measured in degrees Celsius), humidity (as a percentage), water level (as a percentage), nitrogen (N), phosphorus (P), and potassium (K) levels in the soil (scaled between 0 and 255). Additionally, the dataset contained binary indicators for fan actuator, plant watering pump, and water pump states. The target variable for this study was the Fan_actuator_ON feature, a binary indicator representing whether the fan actuator was activated.

2.2 Data Preprocessing

The initial step in data preprocessing involved handling missing values. Missing data was imputed using different strategies depending on the type of feature. For numeric features, the mean value was used to replace missing

Journal of Information System Research (JOSH)
Volume 6, No. 1, Oktober 2024, pp 566−574
ISSN 2686-228X (media online)
https://ejurnal.seminar-id.com/index.php/josh/
DOI 10.47065/josh.v6i1.6095

entries, ensuring that the overall distribution of the data was maintained. Categorical features or binary indicators were imputed using the most frequent value, preserving the inherent distribution of actuator states. After imputing missing values, the date feature, which represented the timestamp of each sensor reading, was transformed into two additional features: hour and minute, both of which were extracted from the timestamp. These new time-based features captured temporal trends that might influence fan actuator behavior. The original date feature subsequently dropped, as it did not provide further useful information for prediction. To prepare the dataset for machine learning models, numerical features were standardized using a StandardScaler, which ensured that each feature had a mean of zero and a standard deviation of one. This step was crucial for models like Support Vector Machines (SVM) and Gradient Boosting, which are sensitive to feature scaling. Standardization helped align the features on a comparable scale, improving the performance of the models.

Data augmentation was applied to simulate real-world noise and imperfections commonly encountered in greenhouse operations. To achieve this, Gaussian noise was injected into the numeric environmental features (excluding the target variable). The noise was generated from a normal distribution with zero mean and a standard deviation of 0.05, representing a 5% noise factor. This method perturbed the feature values slightly, mimicking the inaccuracies often observed in sensor measurements. The target variable Fan_actuator_ON was also modified by introducing label corruption. Approximately 5% of the labels were randomly flipped, changing 0s to 1s and vice versa. This simulated potential errors in the labeling process, adding complexity to the dataset and testing the models' robustness against mislabeled data. Additionally, the dataset exhibited slight class imbalance, with more samples indicating the fan actuator was OFF than ON. To address this imbalance, the Synthetic Minority Over-sampling Technique (SMOTE) was applied. SMOTE generated synthetic samples for the minority class by interpolating between existing minority class samples, effectively balancing the dataset. This ensured that the machine learning models were not biased toward the majority class during training, improving their ability to correctly predict the fan actuator ON state.

2.3 Machine Learning

Several machine learning models were trained and evaluated on the preprocessed and augmented dataset. The first model was a Support Vector Machine (SVM) with a linear kernel, which is well-suited for binary classification tasks where the data can be linearly separable in the feature space. The SVM optimization problem aimed to minimize the hinge loss while maximizing the margin between the two classes (fan actuator ON and OFF), expressed mathematically as

$$\min_{w,b} \quad \frac{1}{2}|w|^2 + C\sum_{i=1}^{n} \quad \max\left(0, 1 - y_i(w^T x_i + b)\right) \tag{1}$$

where (w) is the weight vector, (b) is the bias term, (C) is the regularization parameter, and $((x_i, y_i))$ are the feature vectors and labels. Random Forest Classifier was also employed, leveraging an ensemble of decision trees to reduce variance and improve generalization. The Random Forest model trained multiple decision trees on random subsets of the data and combined their predictions through majority voting. The algorithm's final decision was computed as

$$\hat{y} = \text{mode}\{T_1(x), T_2(x), \ldots, T_B(x)\} \tag{2}$$

where $(T_b(x))$ is the prediction of the (b)-th tree and (B) is the total number of trees in the forest. Gradient Boosting and XGBoost were also applied to the dataset. These algorithms iteratively build models by focusing on the residual errors of previous models. Gradient Boosting minimizes the following loss function by updating the model weights with respect to the negative gradient of the loss function

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

where (η) is the learning rate, and $(F_m(x))$ represents the ensemble model at iteration (m). XGBoost is an efficient implementation of Gradient Boosting that incorporates regularization to prevent overfitting. Logistic Regression was used as a baseline model, offering a simpler approach for binary classification. The Logistic Regression model uses the sigmoid function to map predicted values to probabilities

$$P(y = 1|\mathbf{x}) = \frac{1}{1+e^{-(\mathbf{w^T x}+b)}}. \tag{4}$$

2.4 Evaluation

Finally, model evaluation was conducted using 10-fold cross-validation, where the dataset was split into 10 subsets. For each iteration, one subset was used as the validation set, and the remaining nine subsets were used for training. This process was repeated for all 10 subsets, and the average performance metrics were recorded. Evaluation metrics included accuracy, precision, recall, and F1 score, computed as

$$\text{Precision} = \frac{TP}{TP+FP} \tag{5}$$

Journal of Information System Research (JOSH)
Volume 6, No. 1, Oktober 2024, pp 566−574
ISSN 2686-228X (media online)
https://ejurnal.seminar-id.com/index.php/josh/
DOI 10.47065/josh.v6i1.6095

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}} \tag{6}$$

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{7}$$

where TP, FP, and FN represent true positives, false positives, and false negatives, respectively. Confusion matrices were also used to visualize the models' classification performance, providing insights into how well the models distinguished between fan actuator ON and OFF states.

2.5 Implementation

To implement the machine learning models for predicting fan actuator activation in a smart greenhouse, we structure the data processing and model training pipeline to achieve accurate predictions. This explanation provides a continuous flow through each phase, focusing on the mathematical foundations and model-specific training steps. In the initial phase, data processing involves preparing the dataset for machine learning. The primary goal here is to clean and format the data for consistency across all features. First, handle any missing values in the dataset by imputing the mean for numerical features, ensuring these values maintain the overall distribution, and for binary or categorical features, use the most frequent value to preserve the existing patterns in the actuator states. After this imputation, focus on time-based data: transform timestamp information by extracting relevant features like hour and minute, which can reveal temporal patterns affecting fan activator behavior, then discard the original timestamp column to streamline the dataset. Standardize numerical features using a StandardScaler to maintain a mean of zero and a standard deviation of one, which is essential for algorithms like SVM and Gradient Boosting that are sensitive to feature scaling. Standardization aligns feature values across a comparable scale, enhancing model performance and stability.

Data augmentation follows, adding robustness to the dataset by simulating real-world sensor imperfections. Gaussian noise is injected into the environmental numeric features, generating noise from a normal distribution with a zero mean and a 5\% standard deviation, mimicking potential inaccuracies in sensor readings. This step is crucial for testing each model's resilience to imperfect data. In addition, introduce label corruption by randomly flipping 5\% of the \texttt{Fan_actuator_ON} labels, which simulates possible labeling errors, making the models more robust to real-world misclassifications. Address class imbalance using SMOTE (Synthetic Minority Over-sampling Technique), which generates synthetic samples for the minority class by interpolating between existing samples. This step balances the dataset, ensuring that models do not become biased towards predicting the majority class, enhancing their ability to accurately detect the fan actuator's "ON" state. Once the data is preprocessed and augmented, move on to the model training phase, starting with Support Vector Machine (SVM). The SVM model, equipped with a linear kernel, aims to find a hyperplane that maximally separates the two classes, "ON" and "OFF." The objective here is to minimize hinge loss while maximizing the margin between the classes. The optimization problem for SVM is defined as minimizing the function

$$\frac{1}{2}|w|^2 + C\sum_{i=1}^{n} \quad \max\left(0, 1 - y_i(w^T x_i + b)\right) \tag{8}$$

where (w) represents the weight vector, (b) is the bias, (C) is a regularization parameter balancing margin size and error, and (x_i) and (y_i) are the feature vectors and corresponding labels. The model optimizes (w) and (b) to maximize this margin, ensuring that the two classes are distinctly separated.

Next, the Random Forest model uses an ensemble approach, training multiple decision trees on random subsets of the data and aggregating their predictions through majority voting. Each decision tree is trained by recursively splitting the data based on criteria such as Gini impurity or entropy to reach optimal leaf nodes. For example, Gini impurity, used to evaluate the purity of a split, is calculated as

$$G = 1 - \sum_{k=A,B} p_k^2 \tag{9}$$

where (p_k) represents the probability of a sample belonging to class \(k \). After training all trees, Random Forest predicts a sample's label by taking the mode of the predictions from all individual trees, expressed as

$$\hat{y} = \text{mode}\{T_1(x), T_2(x), \ldots, T_B(x)\} \tag{10}$$

where $(T_b(x))$ is the prediction from the (b)-th tree, and (B) is the number of trees in the forest. This ensemble approach helps reduce overfitting, making Random Forest more robust against noisy data. The Gradient Boosting model iteratively builds a sequence of models that correct the errors of the previous models by focusing on residuals. The process begins with a simple model, such as the mean prediction, then each subsequent model, $(h_m(x))$, is trained to fit the residual errors of the previous model. The update formula for Gradient Boosting is

$$F_m(x) = F_{m-1}(x) + \eta h_m(x) \tag{11}$$

where $(F_{m-1}(x))$ represents the ensemble model at the previous iteration, \(h_m(x) \) is the new model, and (η) is the learning rate, which controls the contribution of each model. Gradient Boosting minimizes the loss function $(L(y, F(x)))$ by fitting $(h_m(x))$ to the negative gradient of the loss, ensuring that each new model

Journal of Information System Research (JOSH)
Volume 6, No. 1, Oktober 2024, pp 566−574
ISSN 2686-228X (media online)
https://ejurnal.seminar-id.com/index.php/josh/
DOI 10.47065/josh.v6i1.6095

improves the overall performance. XGBoost, a more optimized form of Gradient Boosting, includes regularization to prevent overfitting, allowing it to perform well on large datasets. The objective function in XGBoost is defined as

$$\text{Obj} = \sum_{i=1}^{n} L(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(T_k) \quad (11)$$

where (L) is the primary loss function (for instance, log loss for binary classification) and $(\Omega(T_k))$ is a regularization term that penalizes model complexity. The regularization function is

$$\Omega(T) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2 \quad (12)$$

with (T) representing the number of leaves in the tree, (γ) controlling tree complexity, (λ) as the L2 regularization parameter, and (w_j) the weights of each leaf. By iteratively improving the model and applying regularization, XGBoost produces highly accurate predictions while avoiding overfitting, making it well-suited for the complex greenhouse dataset. Finally, Logistic Regression, a baseline model for comparison, employs a simpler probabilistic approach to binary classification by using the sigmoid function to map feature inputs to probabilities. The sigmoid function is defined as

$$P(y = 1|x) = \frac{1}{1+e^{-(w^T x+b)}} \quad (13)$$

where (w) is the weight vector and (b) is the bias term. Logistic Regression minimizes the binary cross-entropy loss, expressed as

$$\text{Loss} = -\frac{1}{n}\sum_{i=1}^{n}[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)] \quad (14)$$

with $(\hat{y}_i)$ representing the predicted probability for the (i)-th sample. While this model is less complex, it provides a useful baseline for performance comparison. In this setup, each model is trained using the structured, augmented dataset, focusing on achieving a high level of accuracy and resilience against data imperfections typical in greenhouse environments. Through these mathematical approaches, each model builds a unique decision boundary or ensemble framework to distinguish between the fan actuator's "ON" and "OFF" states, preparing them for deployment in predictive greenhouse systems.

3. RESULT AND DISCUSSION

This section contains the results and discussion of the research topic, which can be made especially the application of the method used.

3.1 Result Analysis

The results of the machine learning models applied to predict the activation status of fan actuators in a smart greenhouse are presented and discussed in detail. The models evaluated include Support Vector Machine (SVM), Random Forest, Gradient Boosting, XGBoost, and Logistic Regression. These models were assessed based on four primary performance metrics: accuracy, precision, recall, and F1 score. Each of these metrics was computed using 10-fold cross-validation to ensure robust evaluation by dividing the dataset into 10 subsets, training the model on nine subsets, and testing it on the remaining one. This process was repeated across all subsets, and the average performance was recorded. The SVM model demonstrated an accuracy of 91.22%, indicating that around 91% of its predictions were correct. The precision score was 91.94%, reflecting that the model correctly predicted nearly 92% of the instances where it anticipated the fan actuator would be activated. The recall score of 90.37% showed that the model identified 90.37% of the true positive instances, meaning that it successfully predicted the actual "ON" states of the fan actuator most of the time. The F1 score of 91.15% indicated a good balance between precision and recall, making it an effective measure of the model's overall performance.

Table 1. Machine Learning Performance

Model	Accuracy	Precision	Recall	F1 Score
SVM	0.9122	0.9194	0.9037	0.9115
Random Forest	0.9429	0.9744	0.9098	0.941
Gradient Boosting	0.9384	0.974	0.9007	0.936
XGBoost	**0.9447**	**0.9713**	**0.9165**	**0.9431**
Logistic Regression	0.9097	0.914	0.9045	0.9092

Random Forest outperformed SVM, with an accuracy of 94.29%, which suggests that it made correct predictions for more than 94% of the instances. The precision score of 97.44% reflects the model's highly precise predictions of the fan actuator being "ON," with very few false positives. The recall score was 90.98%, slightly lower than the precision, indicating that the model identified most of the true positives, but a small portion of

Journal of Information System Research (JOSH)
Volume 6, No. 1, Oktober 2024, pp 566−574
ISSN 2686-228X (media online)
https://ejurnal.seminar-id.com/index.php/josh/
DOI 10.47065/josh.v6i1.6095

actual "ON" instances were missed. The F1 score was 94.10%, highlighting a strong balance between precision and recall, which confirmed that Random Forest is an effective model for predicting fan actuator status in this context. Gradient Boosting also delivered strong performance, with an accuracy of 93.84%, marginally lower than Random Forest. The precision score was 97.40%, reflecting its accuracy in predicting the fan actuator activation. However, the recall score of 90.07% was slightly lower, indicating a minor tendency to miss true positives compared to Random Forest. The F1 score of 93.60% demonstrated the overall effectiveness of Gradient Boosting, though it may have a marginal bias toward missing some true "ON" states due to its slightly lower recall.

XGBoost, a more optimized version of Gradient Boosting, exhibited the best performance among all models, achieving an accuracy of 94.47%, indicating the highest proportion of correct predictions. Its precision score of 97.13% shows its ability to accurately identify fan actuator activation, though marginally lower than Random Forest's precision. However, its recall score of 91.65% was the highest across all models, demonstrating XGBoost's superior ability to capture the actual "ON" instances of the fan actuator. The F1 score of 94.31% reflected the excellent balance XGBoost achieved between precision and recall, making it the most reliable model for this prediction task. Logistic Regression, a simpler linear model, served as a baseline for comparison. It achieved an accuracy of 90.97%, lower than the more complex models such as Random Forest and XGBoost. The precision score was 91.40%, showing that Logistic Regression had a reasonable ability to predict true positives but performed less precisely than the ensemble models. The recall score of 90.45% indicated a slightly lower ability to capture all true positive instances compared to other models. The F1 score was 90.92%, which, while balanced, was consistently lower than the ensemble models, confirming the limitations of Logistic Regression in handling the complexities of the dataset.

3.2 Discussion

The results indicate that ensemble methods, particularly Random Forest and XGBoost, outperformed the simpler models in predicting the activation status of fan actuators in a smart greenhouse. These ensemble methods demonstrated superior accuracy, precision, recall, and F1 scores, suggesting that they are better suited for handling the complexities of this dataset, which included noisy and corrupted data. XGBoost, which achieved the highest overall accuracy and F1 score, outperformed the other models. Its success can be attributed to its gradient boosting approach, which builds successive models to correct the errors of previous iterations. This feature made XGBoost highly effective for datasets with imperfections, such as the noise and label corruption introduced in this study. By iteratively improving on prior errors, XGBoost fine-tuned its predictions to capture complex patterns in the data, enabling it to excel in this task. Random Forest also performed exceptionally well, with slightly lower accuracy and recall than XGBoost. Random Forest's strength lies in its ensemble of decision trees, each built on random subsets of the data. By aggregating the predictions of multiple decision trees, Random Forest mitigated variance and overfitting, which enabled it to perform robustly even with noisy and corrupted data. The strong performance of these ensemble methods suggests that they are particularly suitable for predictive tasks in environments like smart greenhouses, where sensor data may be imperfect. Gradient Boosting, though slightly less effective than XGBoost, also demonstrated robust performance. Its ability to iteratively correct residual errors makes it a valuable model for predictive tasks involving complex data. However, its recall score suggests that it may miss some true positives, particularly when data noise is prevalent, as was the case in this study.

The SVM model performed reasonably well but did not match the ensemble methods in accuracy or F1 score. SVMs are often effective for binary classification tasks, but they tend to be more susceptible to the effects of noisy data, as evidenced by the lower recall and slightly lower F1 score compared to the ensemble models. The linear kernel used in this study may not have been flexible enough to capture the non-linear relationships in the data, contributing to the model's lower performance. Logistic Regression, while useful as a baseline, performed lower than the more advanced models. Its linear nature likely limited its ability to handle the complex interactions between features in the dataset. While it provided a balanced performance, it was consistently outperformed by the ensemble models in accuracy, precision, recall, and F1 score. The lower performance of Logistic Regression underscores the importance of using more advanced models, such as Random Forest and XGBoost, for predictive tasks in complex environments like smart greenhouses. The data augmentation and noise injection techniques applied in this study were designed to simulate real-world conditions, where sensor data is often noisy or incomplete. The ensemble models, particularly XGBoost and Random Forest, demonstrated strong resilience to these imperfections, as evidenced by their high precision and recall scores. These models are capable of managing variance through the aggregation of predictions from multiple trees or iterations, which reduces the negative impact of noisy data on their overall performance. In contrast, SVM and Logistic Regression were more affected by the introduction of noise, as reflected in their lower recall and F1 scores. These results suggest that ensemble methods are better suited for real-world applications in smart greenhouses, where data imperfections are common.

3.3 Threats to validity

Several potential threats to the validity of the results in this study must be acknowledged, as they could impact the interpretation and generalization of the findings. These threats fall into four main categories: internal validity, external validity, construct validity, and conclusion validity. Each of these aspects poses unique challenges that could influence the reliability of the conclusions drawn from the research. One of the primary concerns regarding

Journal of Information System Research (JOSH)
Volume 6, No. 1, Oktober 2024, pp 566−574
ISSN 2686-228X (media online)
https://ejurnal.seminar-id.com/index.php/josh/
DOI 10.47065/josh.v6i1.6095

internal validity is the impact of data augmentation and label corruption. These techniques were deliberately applied to simulate real-world conditions where sensor data is often noisy or incomplete. However, the introduction of noise may have influenced the models' performance in unintended ways. For instance, corrupting 5% of the labels to mimic erroneous data collection might have disproportionately affected the models' ability to learn from the data, leading to potential overfitting or underfitting in certain instances. Similarly, the application of SMOTE (Synthetic Minority Over-sampling Technique) to balance the classes might have altered the natural distribution of the dataset, which could result in models performing differently when applied to unaltered, imbalanced datasets. Consequently, the effectiveness of the models might vary when they are used on datasets that have not been subject to synthetic alterations, raising questions about the robustness of the models under different real-world conditions. External validity is another significant concern, particularly regarding the generalizability of the study's findings. The dataset used in this research was collected from a specific smart greenhouse setup, which may limit the applicability of the results to other agricultural environments. Although data augmentation and label corruption were introduced to simulate real-world scenarios, the unique characteristics of the dataset such as environmental factors, sensor configurations, and crop management practices may not be representative of other greenhouses. As a result, the models developed in this study might not perform as effectively in different contexts, such as greenhouses operating under varying environmental conditions or using different sensor technologies. To address this issue, further validation on a more diverse range of datasets is necessary to assess the broader applicability of the proposed machine learning models.

This would provide a more accurate understanding of how these models might perform in different agricultural settings, ensuring that the findings are not confined to the specific greenhouse environment examined in this study. Another important consideration is construct validity, which relates to whether the chosen performance metrics accurately reflect the real-world effectiveness of the models. In this study, accuracy, precision, recall, and F1 score were used as primary metrics for evaluating model performance. While these metrics provide valuable insights into the models' ability to predict fan actuator activation, they may not fully capture the operational implications of the models' predictions in a live greenhouse setting. For example, misclassifying the fan actuator as "ON" when it is not needed could lead to unnecessary energy consumption, while failing to activate the fan when required could harm crop health. The use of these traditional machine learning metrics does not account for the economic or environmental impact of such misclassifications, which could be critical in evaluating the practical effectiveness of the models. Future research could benefit from incorporating additional metrics that assess the real-world consequences of prediction errors, providing a more comprehensive evaluation of the models' performance in an operational context.

3.4 Practical Implementation

The practical application of the results from this study indicates that ensemble models, specifically XGBoost and Random Forest, show strong potential for real-world deployment in predicting fan actuator activation in smart greenhouse environments. These models demonstrated high resilience to noisy data, a common occurrence in sensor-based systems, and their superior accuracy, precision, and recall make them viable choices for such implementations. To integrate these models effectively, a robust deployment framework must be established. This framework should begin with a data preprocessing pipeline designed to replicate the resilience observed during model testing. In practice, this pipeline would include data augmentation and noise-handling techniques. This approach would ensure the consistency of data formatting, standardizing incoming sensor data to align with the training data distribution used in this study. Further, a method for handling class imbalances, such as an in-situ SMOTE implementation or equivalent technique, would be critical, especially if the data naturally skews towards one actuator status more frequently than another.

Model selection and tuning are equally important for real-world applications. XGBoost and Random Forest should be the primary models considered for deployment due to their demonstrated robustness with noisy datasets. Hyperparameter tuning should occur periodically to allow models to adapt to any seasonal variations in data or environmental conditions. Tuning methods such as Bayesian optimization or grid search can help in identifying optimal parameter configurations to maintain model accuracy over time. Additionally, version control of the models, with tracking for each iteration, would help in assessing improvements, capturing details on model evolution, and maintaining a well-documented record for reproducibility. The prediction process within the greenhouse control system should function in real time, leveraging model inference to predict the actuator status based on live sensor data. For practical performance, decision thresholds for model predictions may need calibration in response to environmental factors, such as seasonal temperature changes, ensuring that the models maintain a high recall rate and reduce instances of missed fan activations, particularly during critical periods. Integrating the model output with the greenhouse’s actuator control system allows automation of the fan activation process, making predictive control actionable. To enhance operational efficiency, the control logic could factor in economic and environmental impacts, potentially allowing models to adjust fan speeds or activation frequency based on predictive needs rather than using a binary "ON/OFF" control, optimizing energy consumption.

Monitoring and evaluating model performance is essential to maintain efficacy over time, especially as environmental and operational conditions evolve. Precision and recall tracking should remain a priority to avoid the costs associated with false positives, such as unnecessary fan activations, or false negatives, where the system

Journal of Information System Research (JOSH)
Volume 6, No. 1, Oktober 2024, pp 566−574
ISSN 2686-228X (media online)
https://ejurnal.seminar-id.com/index.php/josh/
DOI 10.47065/josh.v6i1.6095

fails to activate the fan when needed, potentially harming crop health. To support continuous improvement, a feedback loop should be implemented to log instances of false predictions, helping refine model training for specific error types over time. Dynamic adaptation of models is critical in these conditions; as sensor data may change with seasonal or environmental variations, retraining models at regular intervals or implementing a continuous learning approach ensures they stay effective and accurate. Limitations identified in the study, such as the impact of data augmentation and label corruption, should be addressed in the deployment phase to enhance the robustness of the system. Testing models on unaltered, imbalanced datasets can provide insight into their performance under more natural conditions, and this testing phase can inform necessary adjustments in training data preprocessing for future models. Additionally, assessing the operational impact of the models' predictions on energy consumption and environmental conditions provides a broader perspective on their practical effectiveness. Misclassifications that lead to unnecessary activations or missed activations may have significant consequences in greenhouse environments, making energy management and environmental impact critical metrics for ongoing model evaluation. Finally, broader deployment considerations should include testing models across varied greenhouse environments with different crop types, climates, and sensor configurations to generalize performance effectively. This cross-context validation would ensure the models' utility is not confined to a specific environment but is adaptable to other greenhouse setups. For settings requiring low-latency decisions, deploying models on edge devices within the greenhouse can further reduce response times, allowing actuator controls to respond swiftly to real-time data, enhancing the system's ability to maintain optimal conditions.

4. CONCLUSION

This study evaluated machine learning models, including SVM, Random Forest, Gradient Boosting, XGBoost, and Logistic Regression, for predicting fan actuator activation in smart greenhouses. XGBoost and Random Forest outperformed the others, demonstrating high accuracy and a strong balance between precision and recall, making them ideal for real-time applications. Data augmentation techniques, such as noise injection and label corruption, were used to simulate real-world conditions, offering a realistic evaluation of model performance. Additionally, SMOTE was applied to address class imbalances, improving prediction accuracy. While the results are promising, the generalizability of these models to other greenhouse environments remains uncertain. Future research should validate these findings with diverse datasets and explore more advanced models, such as hybrid or deep learning approaches, to further enhance predictive accuracy and system resilience.

REFERENCES

[1] M. S. Farooq, R. Javid, S. Riaz, and Z. Atal, "IoT based smart greenhouse framework and control strategies for sustainable agriculture," IEEE Access, vol. 10, pp. 99394–99420, 2022.
[2] F. K. Shaikh, S. Karim, S. Zeadally, and J. Nebhen, "Recent trends in internet-of-things-enabled sensor technologies for smart agriculture," IEEE Internet Things J., vol. 9, no. 23, pp. 23583–23598, 2022.
[3] R. Rayhana, G. Xiao, and Z. Liu, "Internet of things empowered smart greenhouse farming," IEEE J. radio Freq. Identif., vol. 4, no. 3, pp. 195–211, 2020.
[4] C. Maraveas, D. Piromalis, K. G. Arvanitis, T. Bartzanas, and D. Loukatos, "Applications of IoT for optimized greenhouse environment and resources management," Comput. Electron. Agric., vol. 198, p. 106993, 2022.
[5] M. S. Farooq, S. Riaz, M. A. Helou, F. S. Khan, A. Abid, and A. Alvi, "Internet of things in greenhouse agriculture: a survey on enabling technologies, applications, and protocols," IEEE Access, vol. 10, pp. 53374–53397, 2022.
[6] C. Maraveas and T. Bartzanas, "Application of Internet of Things (IoT) for optimized greenhouse environments," AgriEngineering, vol. 3, no. 4, pp. 954–970, 2021.
[7] M. Soussi, M. T. Chaibi, M. Buchholz, and Z. Saghrouni, "Comprehensive review on climate control and cooling systems in greenhouses under hot and arid conditions," Agronomy, vol. 12, no. 3, p. 626, 2022.
[8] M. Guesbaya, "Intelligent control of agriculture production in greenhouses," Université Mohamed Khider Biskra, 2022.
[9] H. Luo, X. Wang, Z. Xu, C. Liu, and J.-S. Pan, "A software-defined multi-modal wireless sensor network for ocean monitoring," Int. J. Distrib. Sens. Networks, vol. 18, no. 1, p. 15501477211068388, 2022.
[10] D. Li, Y. Wang, J. Wang, C. Wang, and Y. Duan, "Recent advances in sensor fault diagnosis: A review," Sensors Actuators A Phys., vol. 309, p. 111990, 2020.
[11] L. Zhang et al., "A review of machine learning in building load prediction," Appl. Energy, vol. 285, p. 116452, 2021.
[12] R. Togneri et al., "Soil moisture forecast for smart irrigation: The primetime for machine learning," Expert Syst. Appl., vol. 207, p. 117653, 2022.
[13] L. I. Chenyang, S. Huiyong, C. ZHANG, L. I. U. Huiqin, G. U. O. Xucun, and X. U. Mengze, "Optimal regulation model of Greenhouse light under limited light resources," in IOP Conference Series: Earth and Environmental Science, 2021, vol. 792, no. 1, p. 12025.
[14] C.-L. Chang, S.-C. Chung, W.-L. Fu, and C.-C. Huang, "Artificial intelligence approaches to predict growth, harvest day, and quality of lettuce (Lactuca sativa L.) in a IoT-enabled greenhouse system," Biosyst. Eng., vol. 212, pp. 77–105, 2021.
[15] A. Escamilla-Garc\'\ia, G. M. Soto-Zarazúa, M. Toledano-Ayala, E. Rivas-Araiza, and A. Gastélum-Barrios, "Applications of artificial neural networks in greenhouse technology and overview for smart agriculture development," Appl. Sci., vol. 10, no. 11, p. 3835, 2020.
[16] D. Xie, L. Chen, L. Liu, L. Chen, and H. Wang, "Actuators and sensors for application in agricultural robots: A review," Machines, vol. 10, no. 10, p. 913, 2022.

Journal of Information System Research (JOSH)
Volume 6, No. 1, Oktober 2024, pp 566−574
ISSN 2686-228X (media online)
https://ejurnal.seminar-id.com/index.php/josh/
DOI 10.47065/josh.v6i1.6095

[17] A. Rokade, M. Singh, P. K. Malik, R. Singh, and T. Alsuwian, "Intelligent data analytics framework for precision farming using IoT and regressor machine learning algorithms," Appl. Sci., vol. 12, no. 19, p. 9992, 2022.
[18] S. E. Whang and J.-G. Lee, "Data collection and quality challenges for deep learning," Proc. VLDB Endow., vol. 13, no. 12, pp. 3429–3432, 2020.
[19] A. M. Rahmani et al., "Machine learning (ML) in medicine: Review, applications, and challenges," Mathematics, vol. 9, no. 22, p. 2970, 2021.
[20] K. Maharana, S. Mondal, and B. Nemade, "A review: Data pre-processing and data augmentation techniques," Glob. Transitions Proc., vol. 3, no. 1, pp. 91–99, 2022.
[21] A. Z. Bayih, J. Morales, Y. Assabie, and R. A. de By, "Utilization of internet of things and wireless sensor networks for sustainable smallholder agriculture," Sensors, vol. 22, no. 9, p. 3273, 2022.
[22] T. PlÖtz, "Applying machine learning for sensor data analysis in interactive systems: Common pitfalls of pragmatic use and ways to avoid them," ACM Comput. Surv., vol. 54, no. 6, pp. 1–25, 2021.
[23] A. Cravero, S. Pardo, S. Sepúlveda, and L. Muñoz, "Challenges to use machine learning in agricultural big data: a systematic literature review," Agronomy, vol. 12, no. 3, p. 748, 2022.
[24] S. R. Melal, M. Aminian, and S. M. Shekarian, "A machine learning method based on stacking heterogeneous ensemble learning for prediction of indoor humidity of greenhouse," J. Agric. Food Res., vol. 16, p. 101107, 2024.
[25] A. Cravero, S. Pardo, P. Galeas, J. López Fenner, and M. Caniupán, "Data type and data sources for agricultural big data and machine learning," Sustainability, vol. 14, no. 23, p. 16131, 2022.