Comparative Evaluation of Machine Learning Models for UAV Network Performance Identification in Dynamic Environments

Gregorius Airlangga [1], **Oskar Ika Adi Nugroho** [2], **Lai Ferry Sugianto** [3]
[1] Department of information Systems, Atma Jaya Catholic University of Indonesia, Jakarta, Indonesia
[2] Department of Electrical Engineering, National Chung Cheng University, Chiayi, Taiwan
[3] Department of Business Administration, Fujen Catholic University, Taipei, Taiwan

ARTICLE INFORMATION

Article History:

Received 25 November 2024
Revised 30 December 2024
Published 07 January 2024

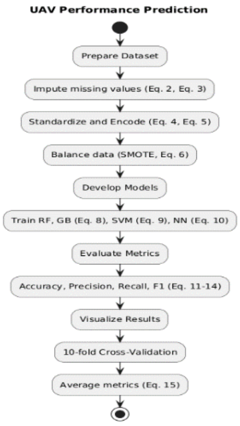
Keywords:

UAV Performance;
Machine Learning Models;
Ensemble Techniques;
Dynamic Environments;
Predictive Analytics

Corresponding Author:

Gregorius Airlangga,
Atma Jaya Catholic University of Indonesia,
Jakarta, Indonesia
Email: gregorius.airlangga@atmajaya.ac.id

ABSTRACT


UAV Performance Prediction
Prepare Dataset
Impute missing values (Eq. 2, Eq. 3)
Standardize and Encode (Eq. 4, Eq. 5)
Balance data (SMOTE, Eq. 6)
Develop Models
Train RF, GB (Eq. 8), SVM (Eq. 9), NN (Eq. 10)
Evaluate Metrics
Accuracy, Precision, Recall, F1 (Eq. 11-14)
Visualize Results
10-fold Cross-Validation
Average metrics (Eq. 15)

The rapid integration of Unmanned Aerial Vehicles (UAVs) into critical applications such as disaster management, logistics, and communication networks has brought forth significant challenges in optimizing their performance under dynamic and unpredictable conditions. This study addresses these challenges by systematically evaluating the predictive capabilities of multiple machine learning models for UAV network performance identification. Models including RandomForest, GradientBoosting, Support Vector Classifier (SVC), Multi-Layer Perceptron (MLP), AdaBoost, ExtraTrees, LogisticRegression, and DecisionTree were analyzed using comprehensive metrics such as average accuracy, macro F1-score, macro precision, and macro recall. The results demonstrated the superiority of ensemble methods, with ExtraTrees achieving the highest performance across all metrics, including an accuracy of 0.9941. Other ensemble models, such as RandomForest and GradientBoosting, also showcased strong results, emphasizing their reliability in handling complex UAV datasets. In contrast, non-ensemble approaches such as LogisticRegression and MLP exhibited comparatively lower performance, suggesting their limitations in generalization under dynamic conditions. Preprocessing techniques, including SMOTE for addressing class imbalances, were applied to enhance model reliability. This research highlights the importance of ensemble learning techniques in achieving robust and balanced UAV performance predictions. The findings provide actionable insights into model selection and optimization strategies, bridging the gap between theoretical advancements and real-world UAV deployment. The proposed methodology and results have impact for advancing UAV technologies in critical, network performance-sensitive applications.

Document Citation:

G. Airlangga, O. I. A. Nugroho, and L. F. Sugianto, “Comparative Evaluation of Machine Learning Models for UAV Network Performance Identification in Dynamic Environments,” *Buletin Ilmiah Sarjana Teknik Elektro*, vol. 6, no. 4, pp. 357-365, 2024, DOI: 10.12928/biste.v6i4.12409.

1. INTRODUCTION

Unmanned Aerial Vehicles (UAVs) have emerged as a transformative technology in modern networked systems, finding applications in fields such as surveillance, disaster management, logistics, and communication networks [1]–[3]. However, ensuring optimal performance in UAV networks remains a critical challenge due to dynamic and unpredictable factors such as environmental conditions, energy constraints, and network congestion [2],[4][5]. These challenges are further amplified in real-world applications where decision-making under uncertainty is paramount, such as disaster recovery operations and time-sensitive deliveries [6]. Current UAV systems often rely on static or heuristic-based optimization approaches that fail to adapt to rapidly changing operational environments [7]. For example, in disaster scenarios, UAVs must operate efficiently despite adverse weather conditions and fluctuating network loads, where poor performance could jeopardize mission outcomes [8]. Similarly, energy inefficiency and suboptimal latency in communication networks can limit the scalability and reliability of UAV-based systems in urban and rural deployments [9]. Addressing these practical problems requires predictive frameworks that can dynamically adapt to operational constraints and optimize key performance indicators such as latency, energy consumption, and throughput [10].

Several studies have explored the integration of machine learning (ML) techniques to address these challenges. For instance, [11] demonstrated the potential of neural networks in optimizing UAV routing, focusing on reducing energy consumption while maintaining communication quality. Similarly, [12] utilized Support Vector Machines (SVMs) to predict UAV throughput under varying environmental conditions, achieving moderate success in controlled scenarios. However, these approaches have been limited by their reliance on small datasets or assumptions about operational variability that do not reflect the complexities of real-world scenarios. Moreover, existing research often adopts a narrow focus, addressing isolated performance metrics rather than providing a holistic view of UAV network optimization. For instance, [13] emphasized latency reduction while [14] targeted energy efficiency. Such siloed approaches fail to capture the interdependencies between multiple performance metrics, such as the trade-offs between throughput, latency, and energy consumption, which are critical for real-time operations. These limitations highlight the need for a comprehensive and integrated approach to UAV network performance identification.

Another significant shortcoming in the literature is the lack of robust comparative studies of machine learning algorithms for UAV network performance prediction [15]–[17]. While some research highlights the potential of ensemble-based methods and hybrid techniques, systematic comparisons across diverse models remain scarce [18],[19]. For example, studies often default to a single algorithm without exploring the relative advantages of alternatives, such as Random Forest, Gradient Boosting, or Neural Networks, under uniform experimental conditions [20]–[22]. This lack of comparative analysis limits the identification of network optimal approaches for diverse UAV operational scenarios [2]. Additionally, class imbalance in UAV performance datasets, characterized by a disproportionate representation of low- and high-performance scenarios poses a significant challenge [23]. Without addressing this issue, model predictions become skewed, undermining their reliability and applicability in practical settings [24]. Techniques such as Synthetic Minority Oversampling (SMOTE) have demonstrated potential in mitigating this imbalance, yet their integration into UAV optimization studies remains underexplored [25]–[27].

The contribution of this research is centered on addressing these gaps through a systematic comparison of machine learning techniques for UAV network performance identification. By leveraging advanced preprocessing methods, including SMOTE for class balancing and cross-validation for robust evaluation, this study provides a rigorous framework for assessing the efficacy of multiple algorithms. Specifically, the research evaluates Random Forest, Gradient Boosting, Support Vector Machines, Neural Networks, and other models to identify their strengths and limitations in predicting UAV performance under varying conditions. Unlike prior studies, this research adopts a holistic evaluation strategy, integrating multiple performance features such as latency, energy consumption, and throughput into the model assessment process. This comprehensive approach ensures that the proposed solutions align with the multifaceted requirements of real-world UAV applications, delivering actionable insights for identifying network performance.

The urgency of this research is underscored by the growing reliance on UAVs in critical applications where performance reliability is non-negotiable. For instance, in disaster recovery operations, predictive models can enable UAVs to prioritize energy efficiency and minimize latency, ensuring timely and effective responses. Similarly, in logistics, optimizing throughput and reducing packet loss directly enhance delivery reliability, particularly in resource-constrained environments. By addressing these practical challenges, this study bridges the gap between theoretical advancements and real-world deployment, contributing to the advancement of UAV network identification. This paper is organized as follows: Section 2 outlines the experimental methodology, including dataset preparation, preprocessing techniques, and machine learning model evaluation. Section 3 presents and discusses the results, focusing on the comparative performance of the evaluated algorithms and their practical implications. Finally, Section 4 concludes with recommendations for future research directions and deployment strategies.

2. METHODS

This study employs a rigorous methodology to predict UAV performance using advanced machine learning techniques. The methodology consists of five key stages: dataset preparation, data preprocessing, model development, evaluation metrics, and experimental design. As presented in the Figure 1, each stage is described comprehensively to ensure clarity and reproducibility.

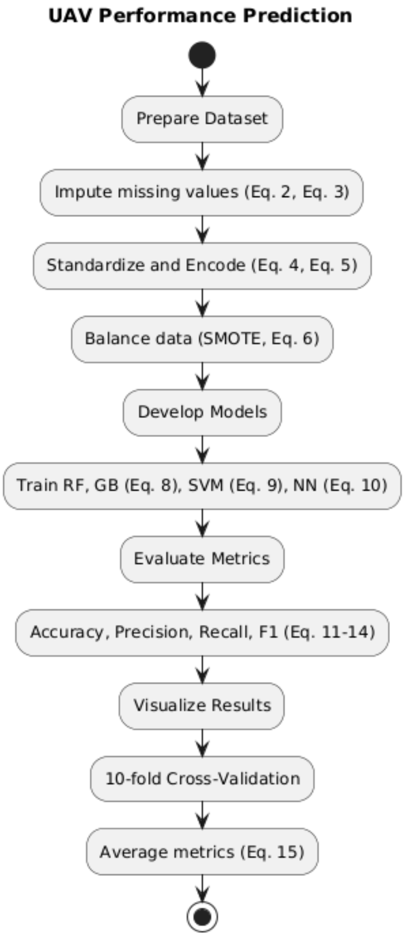

UAV Performance Prediction
Prepare Dataset
Impute missing values (Eq. 2, Eq. 3)
Standardize and Encode (Eq. 4, Eq. 5)
Balance data (SMOTE, Eq. 6)
Develop Models
Train RF, GB (Eq. 8), SVM (Eq. 9), NN (Eq. 10)
Evaluate Metrics
Accuracy, Precision, Recall, F1 (Eq. 11-14)
Visualize Results
10-fold Cross-Validation
Average metrics (Eq. 15)

Figure 1. Research Methodology

2.1. Dataset Preparation

The dataset used in this research consists of network performance metrics of UAVs operating under various environmental and operational conditions. Mathematically, the dataset is represented as (1) and can be downloaded from [28].

$$D = \{(x_i, y_i) \mid x_i \in R^n, y_i \in \{0,1\}\}, \quad i = 1,2,\dots,N \tag{1}$$

where (x_i) denotes the feature vector of the (i)-th sample, comprising (n) features such as energy consumption, latency, throughput, weather conditions, and movement patterns. The binary target variable (y_i) represents the performance category, where $(y_i = 0)$ indicates low performance and $(y_i = 1)$ indicates high performance. The dataset contains (N) samples in total. Missing values in numerical features are addressed through mean substitution. For a given feature $(x_{i,j})$, the missing values are imputed using (2).

$$x_{i,j}^{imputed} = x_{i,j}, if x_{i,j} \neq NaN; \mu_j, if x_{i,j} = NaN \tag{2}$$

where $(\mu_j = \frac{1}{N}\sum_{i=1}^{N} x_{i,j})$ represents the mean of the feature across all samples. For categorical features, missing values are imputed using the mode as presented in (3).

$$c_j^{imputed} = Mode(c_j) \tag{3}$$

where $(\text{Mode}(c_j))$ represents the most frequent category for the feature. This preprocessing step ensures that the dataset is complete, eliminating biases that could arise from incomplete data.

2.2. Data Preprocessing

The data preprocessing step prepares the dataset for effective learning by machine learning models. This involves standardization of numerical features, encoding of categorical features, and addressing class imbalance. Standardization of numerical features ensures that all features are on the same scale, which is crucial for algorithms that rely on distance or gradient-based optimization. Each numerical feature $(x_{i,j})$ is transformed using z-score normalization as presented in (4).

$$x_{i,j}^{scaled} = \frac{x_{i,j} - \mu_j}{\sigma_j} \tag{4}$$

where (μ_j) and (σ_j) represent the mean and standard deviation of the (j)-th feature, respectively. This transformation results in features with a mean of 0 and a standard deviation of 1. Categorical features are encoded using one-hot encoding to transform them into a binary representation. For a categorical variable (c) with (k) unique categories, the encoding produces as presented in (5).

$$c = [c_1, c_2, \dots, c_k], \quad c_i \in \{0,1\} \tag{5}$$

where each element of the vector corresponds to a specific category, and only one element is set to 1 to indicate the active category. Class imbalance is addressed using the Synthetic Minority Oversampling Technique (SMOTE). This technique generates synthetic samples for the minority class to achieve a balanced dataset. For a minority class sample (X_m), synthetic samples (X_s) are generated using (6).

$$X_s = X_m + \lambda \cdot (X_{m'} - X_m), \quad \lambda \sim Uniform(0,1) \tag{6}$$

Here, $(X_{m'})$ is a randomly selected neighbor of (X_m), and (λ) is a random scalar. SMOTE ensures that the machine learning models are not biased toward the majority class, improving their generalization ability.

2.3. Model Development

Several machine learning models are developed and evaluated to predict UAV performance. Each model is chosen based on its unique learning mechanism and mathematical foundations. Random Forest is an ensemble learning method that constructs multiple decision trees during training. Each tree provides a prediction, and the final output is determined through majority voting, given by equation (7).

$$\hat{y} = arg\,\underset{y}{max} \sum_{t=1}^{T} I(y_t = y) \tag{7}$$

where (I) is the indicator function that returns 1 if $(y_t = y)$, and 0 otherwise. Gradient Boosting is an iterative method that minimizes a differentiable loss function $(L(y, f(x)))$. At each iteration, the model is updated as presented in (8).

$$f_{m+1}(x) = f_m(x) + \eta \cdot \nabla L(y, f_m(x)) \tag{8}$$

Here, $(f_m(x))$ is the prediction at iteration (m), $(\nabla L(y, f_m(x)))$ is the gradient of the loss function, and (η) is the learning rate. Support Vector Machines (SVM) aim to find the optimal hyperplane $(w^T x + b = 0)$ that maximizes the margin between two classes. The optimization problem is defined as (9).

$$\underset{w,b}{min} \quad \frac{1}{2}|w|^2, \quad subject\ to\ y_i(w^T x_i + b) \geq 1 \tag{9}$$

Neural Networks use layers of interconnected nodes to model complex relationships. For a neural network with (L) layers, the prediction is computed as (10).

$$\hat{y} = \sigma\Big(W_L\big(\sigma\big(W_{L-1} \dots \big(\sigma(W_1 x + b_1)\big) \dots\big) + b_L\big)\Big) \tag{10}$$

where (W_l) and (b_l) are the weights and biases of the (l)-th layer, and (σ) is the activation function.

2.4. Evaluation Metrics

The evaluation of model performance in this study is conducted using a range of metrics to ensure a comprehensive assessment of predictive capabilities in the context of UAV performance prediction. These metrics include accuracy, precision, recall, and F1-Score, each providing unique insights into the strengths and limitations of the models. By employing these metrics, the study ensures a thorough analysis of model performance, particularly given the potential challenges of class imbalances in the dataset. Accuracy measures the proportion of UAV performance classifications that are correctly identified by the model. In this experiment, accuracy evaluates the overall correctness of predictions across both high- and low-performance UAV scenarios. Mathematically, accuracy is defined as (11).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{11}$$

where (TP) (True Positives) represents the number of high-performance UAV samples correctly identified as high performance, while (TN) (True Negatives) denotes the number of low-performance UAV samples correctly identified as low performance. (FP) (False Positives) refers to low-performance UAV samples incorrectly classified as high performance, and (FN) (False Negatives) corresponds to high-performance UAV samples incorrectly classified as low performance. While accuracy provides a general measure of predictive performance, it is sensitive to class imbalances, as the dataset may contain a disproportionate number of high- or low-performance samples. Precision is particularly important in this study, as it evaluates the proportion of UAV samples predicted as high performance that are actually high performance. This metric is defined as (12).

$$Precision = \frac{TP}{TP + FP} \tag{12}$$

In the context of UAV performance prediction, precision addresses the model's ability to avoid false positives, which occur when low-performance UAVs are misclassified as high performance. High precision is critical for applications where overestimating UAV capabilities could lead to operational failures or suboptimal resource allocation. Recall, also referred to as sensitivity or true positive rate, evaluates the model's ability to identify all high-performance UAV samples. It is defined as (13).

$$Recall = \frac{TP}{TP + FN} \tag{13}$$

Recall is crucial in applications where failing to identify high-performance UAVs could lead to missed opportunities for optimal resource utilization. For instance, in scenarios requiring precise coordination of UAVs with high communication throughput and low latency, recall ensures that all capable UAVs are correctly identified and utilized effectively. The F1-Score is particularly relevant in this experiment, as it provides a single metric that balances precision and recall. It is computed as (14).

$$F1\text{-}Score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{14}$$

This harmonic mean ensures that the model's performance in identifying true high-performance UAVs (recall) is balanced with its ability to minimize false positives (precision). The F1-Score is especially valuable in this study, given the need to optimize UAV predictions in contexts where both false positives and false negatives carry significant operational risks. The relationship among these metrics highlights their complementary nature. While accuracy provides an overarching view of model performance, it does not adequately address the challenges posed by class imbalances, as UAV datasets often contain disproportionately high numbers of low-performance samples. Precision and recall delve deeper into the specific strengths and weaknesses of the models, with precision focusing on minimizing false positives and recall emphasizing the correct identification of high-performance UAVs. The F1-Score then provides a balanced perspective, combining the insights from precision and recall into a single metric.

In the experimental context, these metrics are critical for evaluating and comparing the performance of different machine learning models, including Random Forest, Gradient Boosting, SVM, and Neural Networks. For instance, a Random Forest model may achieve high precision by minimizing false positives but could exhibit lower recall if it struggles to identify all high-performance UAV samples. Conversely, a Neural Network may excel in recall but suffer from lower precision due to false positive predictions. By analyzing these metrics collectively, the study ensures a nuanced evaluation of each model's predictive capabilities. In addition to these metrics, the experimental results are further contextualized through visualization tools such as precision-recall curves and confusion matrices. These tools provide a detailed understanding of model performance under varying thresholds, highlighting trade-offs between precision and recall. For example, adjusting the decision threshold of an SVM model may increase precision at the cost of recall or vice versa.

These insights are essential for optimizing model performance in real-world UAV applications, where the operational context often dictates the relative importance of precision and recall.

2.5. Experimental Design

The experimental design for this study employs a 10-fold cross-validation strategy, a well-established method in machine learning to ensure a thorough and unbiased evaluation of model performance. This technique is chosen due to its ability to balance the trade-off between bias and variance while providing a reliable estimate of the model's ability to generalize to unseen data. Cross-validation leverages the dataset to its fullest extent by ensuring that all data points contribute to both training and testing phases, enhancing the robustness of the evaluation process. In a 10-fold cross-validation approach, the dataset, denoted as (D), is partitioned into ten approximately equal-sized subsets, represented as ($\{D_1, D_2, \ldots, D_{10}\}$). Each subset ($D_k$) (where (k) ranges from 1 to 10) is used as a test set exactly once, while the remaining subsets, ($D \setminus D_k$), collectively form the training set for that fold. This systematic rotation ensures that every data point in (D) is utilized for testing exactly once and for training in nine of the ten iterations. This process ensures that the model is exposed to all instances in the dataset during training and testing, reducing the risk of biased evaluation due to specific data splits. For each fold, the training set is constructed by excluding (D_k), while the model is trained using the remaining data. This trained model is then evaluated on the reserved subset (D_k), which serves as the test set for that iteration. During this evaluation, performance metrics are computed, such as accuracy, precision, recall, F1-score, or mean squared error, depending on the nature of the task, whether classification or regression.

These metrics quantify how well the model performs on the test set, which it has not seen during training, thus providing an unbiased assessment of its predictive capabilities. Once all ten folds are processed, the final performance metric is calculated as the arithmetic mean of the metrics obtained from each fold. This is expressed mathematically as:

$$Metric_{final} = \frac{1}{10}\sum_{k=1}^{10} Metric_k$$

where (Metric_k) represents the performance metric computed during the (k)-th fold. This averaging process ensures that the evaluation reflects the model's performance across all portions of the dataset, reducing the influence of any anomalous results that might occur due to peculiarities in a single fold. The final metric, therefore, serves as a robust indicator of the model's generalizability and reliability. This methodology provides several critical advantages. First, it mitigates the risk of overfitting by ensuring that the model is repeatedly tested on unseen data. Second, it minimizes the variability that might arise from a single arbitrary train-test split, thereby offering a more comprehensive understanding of the model's behavior across different subsets of the dataset. Third, it ensures efficient use of the data, which is particularly beneficial when working with smaller datasets, as no data point is wasted or excluded from the evaluation process.

In practice, the implementation of 10-fold cross-validation is often supported by machine learning libraries, such as scikit-learn in Python, which provide built-in functions to automate this process. The choice of the specific performance metric (Metric_k) depends on the problem at hand. For classification tasks, metrics like accuracy, precision, recall, and F1-score are commonly used, while regression tasks might involve metrics such as mean squared error or mean absolute error. These metrics provide nuanced insights into the strengths and weaknesses of the model in terms of predictive accuracy and robustness. Overall, the adoption of 10-fold cross-validation in this study ensures that the evaluation process is both statistically sound and generalizable. By systematically rotating the roles of training and test data and aggregating results across folds, this approach provides a reliable foundation for assessing the performance of machine learning models in a rigorous and comprehensive manner.

3. RESULT AND DISCUSSION

The results of the classification models applied to the dataset demonstrate significant insights into their comparative performance across various evaluation metrics, including average accuracy, macro F1-score, macro precision, and macro recall. These metrics provide a well-rounded perspective on the models' ability to generalize effectively and handle the intricacies of the classification task. The RandomForest classifier exhibited excellent performance, achieving an average accuracy of 0.9882, which was consistent with its macro F1-score and macro recall at 0.9882. Its macro precision, slightly higher at 0.9894, reflects the model's strong capability to correctly identify positive cases across all classes. This balance between precision and recall suggests that RandomForest is highly reliable and well-suited for tasks where both false positives and false negatives are critical considerations. The stability of its performance indicates robustness and adaptability to the dataset's characteristics.

GradientBoosting delivered nearly identical results to RandomForest, with all metrics, including accuracy, F1-score, precision, and recall, converging at 0.9882. This consistency reinforces GradientBoosting's reputation as a stable and efficient algorithm for classification tasks. However, its inability to exceed RandomForest's performance highlights that both models are similarly adept under the given dataset conditions. The comparable results suggest that either model could be a reliable choice depending on other factors such as computational efficiency or interpretability requirements. The Support Vector Classifier (SVC) also produced comparable results, matching the accuracy, F1-score, and recall of RandomForest and GradientBoosting at 0.9882. However, SVC exhibited a slightly higher macro precision at 0.9894, which indicates its strong ability to reduce false positives while maintaining consistent predictions. This makes SVC a competitive option, particularly in scenarios where precision is prioritized, such as when misclassifying a specific class carries significant consequences.

On the other hand, the Multi-Layer Perceptron (MLP) demonstrated slightly lower performance compared to the ensemble-based models. It achieved an average accuracy of 0.9824, with a macro F1-score of 0.9822, macro precision of 0.9844, and macro recall of 0.9819. These results highlight the model's slight struggle to match the precision and recall balance observed in the ensemble approaches. While MLP remains a competent classifier, its lower reliability in achieving consistently high precision and recall across all classes may limit its application in highly demanding scenarios. AdaBoost achieved similar accuracy to RandomForest, GradientBoosting, and SVC, with a value of 0.9882. However, its macro F1-score was slightly lower at 0.9879, and its macro recall of 0.9875 trailed behind its macro precision of 0.9909. This indicates a slight imbalance in its predictions, favoring precision over recall. While AdaBoost remains a strong performer overall, this trade-off suggests that its suitability might depend on the specific requirements of the classification problem, particularly in cases where higher recall is necessary.

ExtraTrees outperformed all other models in this study, delivering the highest accuracy of 0.9941, alongside a macro F1-score of 0.9940, macro precision of 0.9950, and macro recall of 0.9938. These results underscore the superiority of ExtraTrees in leveraging ensemble learning to capture complex patterns and reduce overfitting effectively. The model's ability to consistently achieve high scores across all metrics highlights its robustness and adaptability, making it the most reliable choice for this dataset. LogisticRegression, while effective, fell short of the ensemble-based models. It achieved an accuracy of 0.9827, a macro F1-score of 0.9825, and macro precision and recall values of 0.9850 and 0.9819, respectively. These metrics suggest that LogisticRegression is a dependable baseline model but lacks the sophistication required to compete with advanced ensemble techniques in terms of overall performance. DecisionTree also achieved an accuracy of 0.9882, with all other metrics: macro F1-score, precision, and recall matching this value. While its performance is solid, DecisionTree does not exhibit the refinement or complexity handling capabilities of the ExtraTrees model or other ensemble-based approaches. This limitation is likely due to its tendency to overfit on training data, which ensemble methods like RandomForest and ExtraTrees effectively mitigate can be seen in Table 1.

Table 1. Performance Results

Model	Average Accuracy	Average Macro F1-Score	Average Macro Precision	Average Macro Recall
RandomForest	0.9882	0.9882	0.9894	0.9882
GradientBoosting	0.9882	0.9882	0.9882	0.9882
SVC	0.9882	0.9882	0.9894	0.9882
MLP	0.9824	0.9822	0.9844	0.9819
AdaBoost	0.9882	0.9879	0.9909	0.9875
ExtraTrees	0.9941	0.994	0.995	0.9938
LogisticRegression	0.9827	0.9825	0.985	0.9819
DecisionTree	0.9882	0.9882	0.9882	0.9882

4. CONCLUSIONS

The study evaluated the performance of several machine learning models on a classification task using four key metrics: average accuracy, macro F1-score, macro precision, and macro recall. The results demonstrated that ensemble-based models consistently outperformed other algorithms, with ExtraTrees emerging as the best-performing model across all evaluation metrics. ExtraTrees achieved an average accuracy of 0.9941, the highest among all models, and exhibited balanced performance across macro F1-score (0.9940), macro precision (0.9950), and macro recall (0.9938). This robust performance highlights the strength of ensemble methods in effectively handling classification tasks by leveraging diverse decision-making processes. RandomForest, GradientBoosting, and AdaBoost also performed exceptionally well, each achieving competitive results with slight variations in their precision and recall values. These findings suggest that these models provide a reliable alternative for scenarios requiring high accuracy and balanced class performance. The Support Vector Classifier (SVC) performed comparably to RandomForest and GradientBoosting, with an accuracy of 0.9882, showcasing its robustness and precision in classification tasks. While neural network-based models like the Multi-Layer Perceptron (MLP) delivered respectable results, their performance was slightly below that of the ensemble models, indicating potential limitations in generalization or sensitivity to hyperparameter tuning. LogisticRegression and DecisionTree, although reliable, were outperformed by more sophisticated approaches, reaffirming the benefits of advanced ensemble techniques. Future work may explore optimizing hyperparameters for underperforming models or integrating hybrid approaches to further enhance classification performance. These insights provide a strong foundation for selecting and fine-tuning models in practical applications requiring reliable predictive accuracy.

REFERENCES

[1] Khan, S. Gupta, and S. K. Gupta, “Emerging UAV technology for disaster detection, mitigation, response, and preparedness,” *Journal of Field Robotics*, vol. 39, no. 6, pp. 905–955, 2022, https://doi.org/10.1002/rob.22075.

[2] S. A. H. Mohsan, M. A. Khan, F. Noor, I. Ullah, and M. H. Alsharif, “Towards the unmanned aerial vehicles (UAVs): A comprehensive review,” *Drones*, vol. 6, no. 6, p. 147, 2022, https://doi.org/10.3390/drones6060147.

[3] F. Syed, S. K. Gupta, S. Hamood Alsamhi, M. Rashid, and X. Liu, “A survey on recent optimal techniques for securing unmanned aerial vehicles applications,” *Trans. Emerg. Telecommun. Technol*., vol. 32, no. 7, p. e4133, 2021, https://doi.org/10.1002/ett.4133.

[4] Y. Bai, H. Zhao, X. Zhang, Z. Chang, R. Jäntti and K. Yang, "Toward Autonomous Multi-UAV Wireless Network: A Survey of Reinforcement Learning-Based Approaches," in *IEEE Communications Surveys & Tutorials*, vol. 25, no. 4, pp. 3038-3067, 2023, https://doi.org/10.1109/COMST.2023.3323344.

[5] M. J. Sobouti, A. Mohajerzadeh, H. Y. Adarbah, Z. Rahimi, and H. Ahmadi, “Utilizing UAVs in wireless networks: advantages, challenges, objectives, and solution methods,” Vehicles, vol. 6, no. 4, pp. 1769–1800, 2024, https://doi.org/10.3390/vehicles6040086.

[6] X. Xu *et al*., “Optimization of reliable vehicle routing problem for medical waste collection with time windows in stochastic transportation networks,” *Engineering Optimization*, pp. 1–31, 2024, https://doi.org/10.1080/0305215X.2024.2388626.

[7] M. Hooshyar and Y.-M. Huang, “Meta-heuristic algorithms in UAV path planning optimization: A systematic review (2018--2022),” *Drones*, vol. 7, no. 12, p. 687, 2023, https://doi.org/10.3390/drones7120687.

[8] S. H. Alsamhi *et al*., “UAV computing-assisted search and rescue mission framework for disaster and harsh environment mitigation,” *Drones*, vol. 6, no. 7, p. 154, 2022, https://doi.org/10.3390/drones6070154.

[9] S. R. Sabuj, A. Ahmed, Y. Cho, K.-J. Lee, and H.-S. Jo, “Cognitive UAV-aided URLLC and mMTC services: Analyzing energy efficiency and latency,” *IEEE Access*, vol. 9, pp. 5011–5027, 2020, https://doi.org/10.1109/ACCESS.2020.3048436.

[10] M. S. Aslanpour, S. S. Gill, and A. N. Toosi, “Performance evaluation metrics for cloud, fog and edge computing: A review, taxonomy, benchmarks and standards for future research,” *Internet of Things*, vol. 12, p. 100273, 2020, https://doi.org/10.1016/j.iot.2020.100273.

[11] M. A. Sayeed, R. Kumar, V. Sharma, and M. A. Sayeed, “Efficient deployment with throughput maximization for UAVs communication networks,” *Sensors*, vol. 20, no. 22, p. 6680, 2020, https://doi.org/10.3390/s20226680.

[12] S. Zeng, X.-J. Xiang, Y.-P. Dou, J.-C. Du, and G. He, “UAV data link anti-interference via SLHS-SVM-AdaBoost algorithm: Classification prediction and route planning,” *J. Electron. Sci. Technol*., vol. 22, no. 4, p. 100279, 2024, https://doi.org/10.1016/j.jnlest.2024.100279.

[13] Y. Liu, J. Yan and X. Zhao, "Deep Reinforcement Learning Based Latency Minimization for Mobile Edge Computing With Virtualization in Maritime UAV Communication Network," in *IEEE Transactions on Vehicular Technology*, vol. 71, no. 4, pp. 4225-4236, 2022, https://doi.org/10.1109/TVT.2022.3141799.

[14] S. Hu, J. Wang, C. Hoare, Y. Li, P. Pauwels, and J. O’Donnell, “Building energy performance assessment using linked data and cross-domain semantic reasoning,” *Autom. Constr*., vol. 124, p. 103580, 2021, https://doi.org/10.1016/j.autcon.2021.103580.

[15] A. Telikani, A. Sarkar, B. Du and J. Shen, "Machine Learning for UAV-Aided ITS: A Review With Comparative Study," in *IEEE Transactions on Intelligent Transportation Systems*, vol. 25, no. 11, pp. 15388-15406, 2024, https://doi.org/10.1109/TITS.2024.3422039.

[16] H H. Yuliana, Iskandar and Hendrawan, "Comparative Analysis of Machine Learning Algorithms for 5G Coverage Prediction: Identification of Dominant Feature Parameters and Prediction Accuracy," in *IEEE Access*, vol. 12, pp. 18939-18956, 2024, https://doi.org/10.1109/ACCESS.2024.3361403.

[17] P. S. Bithas, E. T. Michailidis, N. Nomikos, D. Vouyioukas, and A. G. Kanatas, “A survey on machine-learning techniques for UAV-based communications,” *Sensors*, vol. 19, no. 23, p. 5170, 2019, https://doi.org/10.3390/s19235170.

[18] S. Abimannan, E. -S. M. El-Alfy, Y. -S. Chang, S. Hussain, S. Shukla and D. Satheesh, "Ensemble Multifeatured Deep Learning Models and Applications: A Survey," in *IEEE Access*, vol. 11, pp. 107194-107217, 2023, https://doi.org/10.1109/ACCESS.2023.3320042.

[19] U. Seidaliyeva, L. Ilipbayeva, K. Taissariyeva, N. Smailov, and E. T. Matson, "Advances and challenges in drone detection and classification techniques: A state-of-the-art review," *Sensors*, vol. 24, no. 1, p. 125, 2023, https://doi.org/10.3390/s24010125.

[20] E. Asamoah, G. B. M. Heuvelink, I. Chairi, P. S. Bindraban, and V. Logah, "Random forest machine learning for maize yield and agronomic efficiency prediction in Ghana," *Heliyon*, vol. 10, no. 17, 2024, https://doi.org/10.1016/j.heliyon.2024.e37065.

[21] A. R. Al-Aizari *et al*., "Uncertainty reduction in Flood susceptibility mapping using Random Forest and eXtreme Gradient Boosting algorithms in two Tropical Desert cities, Shibam and Marib, Yemen," *Remote Sens*., vol. 16, no. 2, p. 336, 2024, https://doi.org/10.3390/rs16020336.

[22] M. Zou *et al*., "Combining spectral and texture feature of UAV image with plant height to improve LAI estimation of winter wheat at jointing stage," *Front. Plant Sci*., vol. 14, p. 1272049, 2024, https://doi.org/10.3389/fpls.2023.1272049.

[23] A. Hussain, S. Li, T. Hussain, X. Lin, F. Ali, and A. A. AlZubi, "Computing Challenges of UAV Networks: A Comprehensive Survey.," *Computers, Materials & Continua*, vol. 81, no. 2, 2024, https://doi.org/10.32604/cmc.2024.056183.

[24] R. A. Nihal, B. Yen, K. Itoyama, and K. Nakadai, "UAV-Enhanced Combination to Application: Comprehensive Analysis and Benchmarking of a Human Detection Dataset for Disaster Scenarios," in *International Conference on Pattern Recognition*, pp. 145–162, 2024, https://doi.org/10.1007/978-3-031-78341-8_10.

[25] W. Yang, A. Acuto, Y. Zhou, and D. Wojtczak, "A Survey for Deep Reinforcement Learning Based Network Intrusion Detection," *arXiv Prepr. arXiv2410.07612*, 2024, https://doi.org/10.48550/arXiv.2410.07612.

[26] X. Lei *et al*., "An Ensemble Machine Learning Model to Estimate Urban Water Quality Parameters Using Unmanned Aerial Vehicle Multispectral Imagery," *Remote Sens*., vol. 16, no. 12, p. 2246, 2024, https://doi.org/10.3390/rs16122246.

[27] F. Zhang *et al*., "Data preparation for deep learning based code smell detection: A systematic literature review," *J. Syst. Softw*., p. 112131, 2024, https://doi.org/10.1016/j.jss.2024.112131.

[28] J. Eo, D. Lee and M. Kwon, "The Impact of Dataset on Offline Reinforcement Learning Performance in UAV-Based Emergency Network Recovery Tasks," in *IEEE Communications Letters*, vol. 28, no. 5, pp. 1058-1061, May 2024, https://doi.org/10.1109/LCOMM.2023.3339478.

AUTHOR BIOGRAPHY

Gregorius Airlangga is received the B.S. degree in information system from the Yos Sudarso Higher School of Computer Science, Purwokerto, Indonesia, in 2014, and the M.Eng. degree in informatics from Atma Jaya Yogyakarta University, Yogyakarta, Indonesia, in 2016. He got Ph.D. degree with the Department of Electrical Engineering, National Chung Cheng University, Taiwan. He is also an Assistant Professor with the Department of Information System, Atma Jaya Catholic University of Indonesia, Jakarta, Indonesia. His research interests include UAV, data science, artificial intelligence and software engineering include path planning, machine learning, natural language processing, deep learning, software requirements, software design pattern and software architecture.