Terbit online pada laman web jurnal: **http://journal.isas.or.id/index.php/JACOST**

JOURNAL OF APPLIED COMPUTER SCIENCE AND TECHNOLOGY (JACOST)

Vol. 7 No. 1 (2026) 9 – 21 | ISSN: 2723-1453 (Media Online)

A Comparative Study of Random Forest, K-Nearest Neighbors, and XGBoost Models for Weather-Aware Smart Office Building Automation

Erwin Yonata[1], Maya Anggun Beer[2], Ni Nyoman Putri Shopia[3], Emilia Loho[4], Gilang Raka Rayuda Dewa[5]
[1,2,3,4,5]Computer Science, Faculty of Engineering and Technology, Sampoerna University
[1]erwin.yonata@my.sampoernauniversity.ac.id, [2]maya.beer@my.sampoernauniversity.ac.id,
[3]putri.yuandari@my.sampoernauniversity.ac.id, [4]emilia.loho@my.sampoernauniversity.ac.id,
[5]gilang.dewa@sampoernauniversity.ac.id*

Abstract

The intelligent control of lighting and HVAC systems plays a critical role in reducing energy consumption in smart buildings. However, many existing automation systems rely on static scheduling strategies that fail to adapt to dynamic environmental conditions. Although machine learning has been widely applied to weather-based building automation, inconsistent feature selection, model configuration, and evaluation procedures limit the validity of comparative performance claims. This study aims to develop and evaluate a machine-learning-based weather classification framework for smart building automation. The proposed methodology follows a structured pipeline comprising data acquisition and preprocessing, model training and testing, parameter tuning, and performance evaluation. A publicly available Weather Type Classification dataset is used, consisting of numerical weather parameters, which are encoded prior to training. Feature selection is applied to identify the most influential predictors. Three machine learning models, Random Forest, K Nearest Neighbors, and XGBoost, are trained using an 80:20 stratified split, with hyperparameters optimized through grid search to ensure an optimized model. Model performance is evaluated using accuracy, precision, recall, F1 score, and a confusion matrix. Experimental results demonstrate that Random Forest achieves the highest accuracy of 97.50 percent, followed by XGBoost at 96.90 percent and K Nearest Neighbors at 95.73 percent, with balanced performance across all weather categories. The findings indicate that ensemble-based classifiers are well-suited for robust weather recognition. The classified weather outputs can be directly mapped to real-time control strategies for lighting and HVAC systems, enabling adaptive automation and improved energy efficiency in smart buildings.

Keywords: building, K-Nearest Neighbor, Random Forest, XGBoost, weather

1. Introduction

The smart city concept has been rapidly developed as a global priority to support an efficient and sustainable lifestyle. According to data from Statista, the global smart city market is projected to experience substantial growth, with revenue expected to reach $79.94 billion by 2025 [1]. This increasing trend is anticipated to continue with a compound annual growth rate (CAGR) of 9.60% from 2025 to 2029 [1], [2]. These numbers demonstrate the growing global interest in leveraging technology to enhance the quality of urban life [3].

A critical aspect of smart cities is the use of intelligent systems in office buildings, which are responsible for a significant portion of energy usage. According to a 2018 survey conducted by the U.S. Energy Information Administration, office buildings in the United States consumed approximately 1,025 trillion British thermal units (TBtu) of electricity and natural gas energy combined, primarily due to space heating, ventilation, lighting, and cooling systems [2]. Weather conditions have a direct impact on these energy needs, either in the number of lighting intensity, heating system utilization, or air conditioning; recent work shows that integrating weather forecasts and ML-based predictive control can substantially improve HVAC and lighting efficiency. For example, tree-based and reinforcement-learning ML approaches have been applied to HVAC demand-response and have shown promising results in reducing energy use and improving control responsiveness [4]. However, numerous current systems in office buildings still rely on fixed schedules. Several systems utilize external weather forecast data; however, they often do not provide real-time responses and may not accurately reflect local conditions. Most existing weather-related applications and systems are designed to provide users with general information, using time-series data in specific areas to generate forecast results. These systems

often rely on third-party data and do not utilize weather sensors that may already be available in office environments, such as temperature, humidity, and wind speed sensors. This not only complicates the algorithm but also results in less accurate predictions for a particular office building due to the lack of localized data [5], [6]. Recent surveys and IoT/systematic reviews emphasize the growing importance of on-site sensor networks and IoT integration for accurate, localized building control and energy management, and highlight practical barriers and solutions for adopting sensor-driven approaches in buildings [7], [8].

As a result, there is a clear need for a machine learning–based solution capable of classifying current weather conditions from real-time sensor data to support responsive and localized building automation. Accurate weather classification is a critical component of adaptive control strategies in smart buildings, enabling heating, cooling, and ventilation systems to dynamically adjust their operations based on directly observed environmental conditions. Several studies have explored weather-aware control using machine learning techniques, such as decision trees [9], support vector machines [10], and neural networks [11]. However, these works typically focus on improving prediction accuracy for a single selected model, often through extensive hyperparameter tuning [12], while neglecting systematic comparisons across multiple models under equivalent experimental conditions.

This practice introduces a significant research gap: the lack of fair and reproducible model comparisons for real-time weather classification in building automation contexts. When only one model is optimized while others are evaluated using default or suboptimal settings, the resulting conclusions are biased and provide limited guidance for practical deployment. Furthermore, most existing studies emphasize empirical performance without providing a clear mathematical formulation of the classification problem, including definitions of input feature spaces, class boundaries, and decision functions as they relate to physical weather phenomena and sensor behavior [9-11]. Consequently, it remains unclear how these models generalize across buildings, sensor configurations, or climatic conditions.

To address these limitations, this study formulates weather condition classification as a supervised learning problem. Unlike prior work, this research adopts a methodologically fair evaluation framework, in which multiple machine learning classifiers are trained and tested under consistent data preprocessing, feature sets, and parameter selection strategies. This allows performance differences to be attributed to model characteristics rather than tuning bias.

The main objective of this study is to develop and evaluate a robust weather classification framework suitable for real-time building automation while ensuring fairness and transparency in model comparison. The novelty of this work lies in three aspects: (1) a precise mathematical formulation of the weather classification problem tailored to building automation needs, (2) a balanced comparative analysis of multiple machine learning models without selective parameter tuning, and (3) a focus on deployment-relevant decision support, where classification outputs directly inform energy-saving control actions. By addressing these gaps, the study contributes both methodological clarity and practical insight to the design of weather-aware intelligent building systems

Therefore, to achieve research objectives, our main contribution is summarized as follows:

Mathematical Formalization of Weather-Aware Building Automation: This paper models weather-aware building automation system automation as a supervised multi-class classification problem. The system utilizes six ecological features as input to predict discrete weather classes. This formalization enables the integration of real-time environmental sensing into adaptive and data-driven automation strategies.

Non-biased Hyperparameter Setting Across All Models: To ensure an unbiased comparison, all machine learning models are individually tuned using hyperparameter search strategies that correspond to each algorithm. Here, we employ a grid search to identify the optimal configuration for each model. This ensures that each model operates under its optimal condition, maximizing performance without favoring any particular approach. As a result, our comparison accurately reflects the model's true capability, rather than differences in tuning effort.

Empirical Evaluation of Machine Learning Classifiers: This paper conducts a controlled empirical comparison of three popular classifiers: K-Nearest Neighbors (KNN), Random Forest (RF), and XGBoost. Each model is evaluated using identical input features, with metrics such as accuracy, precision, recall, and F1-score. Our results show that Random Forest achieves the best performance with 98% accuracy, outperforming XGBoost at 97% and KNN at 96%. These findings provide evidence-based insights for selecting reliable and efficient models in innovative building applications.

The reminder paper is organized as follows. Section 2 the methodological aspects, respectively. Section 3 presents a simulation result and the discussion. Finally, the conclusion is summarized in Section 4.

In recent years, intelligent building technology has been rapidly evolving, including lighting, heating, and other potential solutions. These systems aim to reduce electricity consumption by automatically managing electronic devices based on environmental conditions, user behavior, and operational needs. As demand for intelligent infrastructure grows, electronic automation

DOI : https://doi.org/10.52158/7925qh24

has evolved from simple timers to context-aware solutions that prioritize both efficiency and user comfort.

Timed-Based Model: The typical implementation model in office environments is the time-scheduled lighting system, as described by Rahman et al. [13]. In this setup, lights are programmed to switch on and off according to fixed working hours, typically turning on at 8:00 AM and off at 6:00 PM. This method is straightforward, cost-effective, and works properly in buildings with regular and predictable schedules. It minimizes the need for complex hardware. However, the simple model also incur overutilization and underutilization, because it does not account for fluctuating daylight levels or variations in occupancy.

Motion-Based Model Comparison: Another technique in building automation is motion-based or occupancy-sensing systems, which rely on sensors, e.g., passive infrared (PIR) detectors, to activate lighting in response to movement. This model is particularly effective in intermittently used spaces such as meeting rooms, corridors, restrooms, and storage areas. It ensures that electricity is only active when the area is in use, thereby preventing waste in seldom-occupied zones. However, while occupancy sensors provide clear benefits in shared or transitional spaces, they encounter significant challenges in individual workspaces where occupants may remain still for extended periods. In such cases, a lack of movement can falsely signal that the space is unoccupied. Furthermore, like time-scheduled systems, motion-based lighting generally does not consider daylight variability, resulting in redundant lighting even when natural light is sufficient.

Machine Learning for Weather Classification: To implement automatic office appliance control based on weather classification, a wide range of machine learning models has been introduced recently, including tree-based methods and ensemble techniques [14]. Instance-based approaches, such as k-Nearest Neighbors (KNN), have been widely applied in early and recent studies on weather prediction and classification using historical meteorological data [15], [16]. Empirical results generally indicate that KNN achieves reasonable accuracy on small- to medium-scale datasets with well-structured features. However, multiple studies identify inherent limitations that restrict its suitability for real-time or large-scale deployment. Specifically, KNN exhibits high computational complexity during inference as the dataset size grows, making it less practical for continuous and sensor-driven systems [17]. Furthermore, its sensitivity to irrelevant or redundant features and to issues with feature scaling can significantly degrade classification performance, particularly in heterogeneous IoT-based sensing environments [18], [19]. These findings suggest that while KNN is useful as a baseline method, it may not be ideal for dynamic building automation scenarios.

Tree-based ensemble methods have gained increasing attention in recent years due to their robustness and scalability. In particular, XGBoost has been extensively adopted for short-term weather-related prediction tasks, including wind speed estimation, rainfall classification, and energy-related environmental modeling. In the study by Zheng et al. [6], XGBoost is employed for short-term wind power forecasting, demonstrating its capability to model complex and nonlinear relationships [20]. This research highlights XGBoost's robustness and superior performance compared to traditional models, contributing its success to advanced features such as regularization, parallel processing, and tree pruning. These characteristics enable XGBoost to effectively mitigate overfitting and improve generalization, making it suitable for applications involving dynamic, high-dimensional datasets. However, the model's complexity can lead to increased computational demands, particularly during hyperparameter tuning and when dealing with large-scale data [21]. Additionally, XGBoost's ensemble nature can make it more complex compared to simpler models [22]. Moreover, its sensitivity to noisy data and outliers can significantly impact on the overall performance [8], [23]. Recent applied work also demonstrates XGBoost's practical strength when used within predictive HVAC control pipelines and other building energy applications [3], [4], [24].

In the aforementioned study by Ayankemi et al. [5], the Random Forest classifier outperformed both the Decision Tree and Extra Tree classifiers, achieving the highest accuracy of 66% on the weather prediction dataset. The ensemble nature of Random Forests allows them to capture complex patterns in the data while maintaining robustness against noise and overfitting. More generally, reviews of ML for smart buildings show that Random Forest and ensemble methods are often favored for their interpretability and robustness in building energy tasks [25]. Feature-selection and IoT-sensor design studies also emphasize that careful selection of a compact sensor feature set (temperature, UV, visibility, precipitation, pressure, cloud cover in our case) improves deployment feasibility while retaining predictive power [26], [27].

Identified Gaps and Study Contribution: Despite the promising developments in both rule-based and machine learning-driven automation systems, several limitations remain. Time-based and motion-based models, while simple and widely adopted, often fail to account for real-time environmental conditions and natural light variability. Meanwhile, existing machine learning approaches for weather prediction typically focus on forecasting using historical data or third-party sources, with limited emphasis on real-time sensor-based classification. Moreover, previous studies often conduct selective or uneven hyperparameter tuning, leading to

DOI : https://doi.org/10.52158/7925qh24

biased comparisons and unclear guidance for practical deployment.

To address these gaps, this study presents a mathematically formulated sensor-driven weather classification framework designed explicitly for intelligent office automation. This paper performs an unbiased and head-to-head comparison of multiple machine learning models, specifically KNN, XGBoost, and Random Forest. Each algorithm is optimized using consistent and model-specific hyperparameter tuning strategies. By leveraging real-time ecological data from local sensors, the proposed system enables context-aware automation that reduces energy consumption while maintaining occupant comfort. This integrated approach ensures both fair model evaluation and practical relevance in innovative building environments.

This section describes the mathematical procedures used in machine learning models. Consider a multi-class classification problem involving multiple features. Suppose $\mathcal{D} = \{(\boldsymbol{x}^{(i)}, y^{(i)})\}_{i=1}^{N}$ is a dataset that consists of N numbers. Here, $\boldsymbol{x}^{(i)} \in \mathbb{R}^m$ denotes a feature vector that contains m ecological measurement. In addition, $y^{(i)} \in \{1,2,\dots,K\}$ represents discrete weather class labels for the sample i, where K is the total number of weather conditions. [26], [28]

Several machine learning models, i.e., K-Nearest Neighbor, Random Forest, and XGBoost, are applied as part of the evaluation, which are defined as Equation 1.

$$F = \{f_{KNN}, f_{RF}, f_{XGB}\}. \quad (1)$$

Each model utilizes a function $f_i \in F$ that maps input features to a predicted class. Hence, the prediction class is computed as Equation 2.

$$y^{(i)} = f(\boldsymbol{x}^{(i)}). \quad (2)$$

In this model, we aim to minimize the number of unmatched mappings between features and classes. The objective function is formulated as Equation 3.

$$\min_{f_i \in F} \frac{1}{N} \sum_{i=1}^{N} \dagger \left[f(\boldsymbol{x}^{(i)}) \neq y^{(i)} \right], \quad (3)$$

where † denotes a binary indicator function, which takes the value 1 if true and 0 if otherwise. Therefore, accuracy can be simply defined as the correct prediction of all the total prediction. Here, the accuracy can be defined as Equation 4.

$$Accuracy(f) = \frac{1}{N} \sum_{i=1}^{N} \dagger \left[f(\boldsymbol{x}^{(i)}) = y^{(i)} \right] \quad (4)$$

Finally, this study defines the best-performing model as Equation 5.

$$f^* = arg \max_{f_i \in F} Accuracy(f_i) \quad (5)$$

2. Methods

Developing a real-time weather classification system using sensor data requires a machine learning framework that can systematically process heterogeneous environmental inputs. Accordingly, this study adopts an experimental research design based on supervised machine learning. The proposed framework is designed to transform raw multi-sensor weather observations into discrete weather classes that can support weather-aware building automation decisions. The methodology follows several structured stages: data acquisition and preprocessing, model training and testing, model parameter setting, and evaluation, ensuring methodological transparency and reproducibility.

2.1. Data Acquisition and Preprocessing

The dataset utilized in this study is obtained from the Weather Type Classification dataset available on Kaggle [29]. This dataset is selected due to its relevance to weather-aware building automation and its inclusion of variables commonly measured by IoT-based environmental sensors. The dataset comprises numerical parameters, including Temperature, Humidity, Wind Speed, Precipitation (%), Atmospheric Pressure, UV Index, and Visibility (in kilometers), as well as categorical attributes such as Cloud Cover, Season, and Location. These variables collectively represent the environmental conditions required for real-time weather characterization in building control scenarios. The target variable represents discrete weather types.

Prior to model training, an initial data quality assessment is conducted to identify missing or inconsistent values. As no significant missing data is observed, no imputation is required. Categorical variables are transformed using label encoding, as presented in Table 1, where each category is assigned a unique integer value. This encoding approach is computationally efficient and appropriate for categorical variables with a limited number of levels, particularly when used with tree-based classifiers. Label encoding also provides a consistent and reproducible feature representation across training and testing phases. Consistent with recent surveys on feature-selection techniques for IoT and smart-building machine learning, this study emphasizes reproducible feature representation and combines domain knowledge with automated selection procedures to prevent irrelevant predictors from degrading classifier performance [28]. Table 1 provides the mappings from categorical to numerical values used in this study to ensure clarity and reproducibility.

Table 1. Label Encoder Categorical Data

Feature	Original Value	Encoded
Cloud Cover	clear, cloudy, overcast, partly cloudy	0, 1, 2, 3
Season	Autumn, Spring, Summer, Winter	0, 1, 2, 3
Location	coastal, inland, mountain	0, 1, 2
Weather Types	Cloudy, Rainy, Snowy, Sunny	0, 1, 2, 3

DOI : https://doi.org/10.52158/7925qh24

Numerical features are examined for outliers using the Interquartile Range (IQR) method [22]. This approach identifies data points that fall significantly outside the interquartile range, ensuring that the dataset is free from anomalies. This statistical technique is selected due to its robustness to skewed distributions and its suitability for environmental sensor data.

It should be noted that normalization or standardization of numerical features is not applied to the Random Forest model. This decision is based on the nature of tree-based algorithms, which are inherently insensitive to the scale of input features. Random Forest operates by partitioning the data using feature thresholds. Each decision tree in the forest makes splits based on feature values without considering feature scale. Therefore, normalization does not impact the performance of Random Forest models.

2.2. Model Training and Testing

All experiments are conducted using a Kaggle Jupyter Notebook environment to facilitate reproducibility. The dataset is partitioned into training and test sets using an 80:20 split, with stratification to preserve the original class distribution across both sets. Stratified sampling ensures that minority weather classes are adequately represented during both training and evaluation.

The training subset is used for model learning and parameter optimization, while the testing subset is reserved exclusively for performance evaluation, ensuring an unbiased assessment of model generalization.

2.3. Model Parameter Setting

An initial model comparison is performed using all ten available features to establish a baseline performance. Subsequently, feature selection is applied to identify the predictors that contribute most significantly to the classification outcome. Based on this process, the six most influential features are selected for further analysis. These selected features are then used for hyperparameter tuning to improve model stability and identify the optimal parameter configuration for the best-performing model.

Hyperparameter optimization is conducted using Grid Search with 5-fold cross-validation [18], which systematically evaluates candidate parameter combinations while mitigating overfitting. The optimal hyperparameters identified through cross-validation are used to train the final model. This approach aligns with established best practices in building-focused machine learning pipelines, where cross-validated grid search combined with feature reduction and IoT sensor considerations has been shown to yield reliable and deployable control models [4], [27], [30].

2. 4. Evaluation

The performance of the final model is evaluated using standard classification metrics, including accuracy, precision, recall, and F1-score. These metrics provide complementary perspectives on overall correctness, class-specific performance, and robustness under potential class imbalance. In addition, a confusion matrix is generated to visualize classification outcomes across the four weather categories, enabling detailed analysis of misclassification patterns and model behavior.

3. Results and Discussions

The weather classification model is developed as a preliminary step toward building automation. Although the long-term goal of this study is to develop a machine learning-based solution that utilizes real-time sensor data to support energy-efficient automation, the current work focuses on creating and testing a machine learning model using a publicly available weather dataset. As part of the experiment, the trained model is also tested in a basic automation scenario, where it is used to control a lighting system based on the predicted weather conditions.

3.1. Random Forest

The initial Random Forest model is trained using two selected hyperparameters, as shown in Table 2.

Table 2. Initial Random Forest Hyperparameter

Hyperparameter	Value
n_estimators	200
max_features	‘sqrt’

By implementing these hyperparameters, the overall results are obtained in Table 3. The model successfully achieves an overall test accuracy of 97.28%. In addition, the model also achieves high results for Precision, Recall, and F1-Score across all classes, with a macro average of each metric at 97%. In terms of per-classification results, it generally achieves high performance with a minimum of 94%.

Table 3. Initial Random Forest Classification Report

Class	Precision	Recall	F1-score	Support
Cloudy	0.94	0.98	0.96	592
Rainy	0.98	0.97	0.97	562
Snowy	0.99	0.98	0.99	575
Sunny	0.99	0.96	0.98	589
Overall Accuracy				0.9728

While all classes achieve a high overall performance, “Cloudy” exhibits slightly lower precision, specifically 94%, indicating that the model occasionally misclassifies cases as other classes. This is further illustrated in the confusion matrix, as shown in Figure 1, which reveals patterns of misclassification: 9 cases of “Cloudy” are predicted as “Rainy,” 14 cases of “Rainy” are predicted as “Cloudy,” and 19 cases of “Sunny” are predicted as “Cloudy.” These misclassifications indicate

DOI : https://doi.org/10.52158/7925qh24

feature overlaps between these classes, likely due to similar atmospheric conditions that challenge strict visual or sensor-based distinctions.

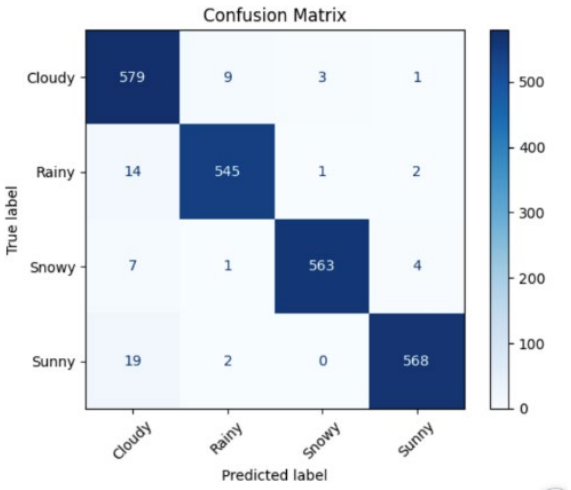

Figure 1. Initial Random Forest Confusion Matrix

A key strength of Random Forest lies in its built-in feature importance scoring, which enables the interpretation of the model to be more generalized. Based on the importance values as shown in Figure 2, the top contributors to weather classification are Temperature, UV Index, Visibility (km), Precipitation (%), Atmospheric Pressure, and Cloud Cover.

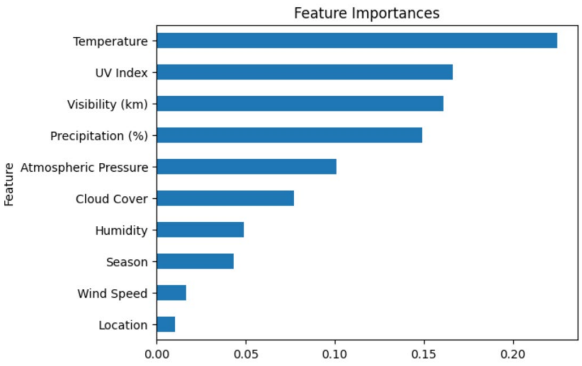

Figure 2. Initial Random Forest Feature Importance

To explore potential model simplification, experiments are conducted by selecting only the most essential features. Using the top 5 features, accuracy drops to 96.07%, suggesting some loss of critical information.

However, as shown by Figure 3, adding the 6th most crucial feature (adding the additional feature 'Cloud Cover') restored accuracy to 97.33%, matching the full-feature baseline with a very slight improvement. Despite adding a seventh feature, the model's accuracy remains at 97.28%. This demonstrates the model's robustness and confirms that these six features capture most of the discriminatory power needed for accurate classification, which is particularly valuable for real-time, innovative office applications where computational efficiency is essential.

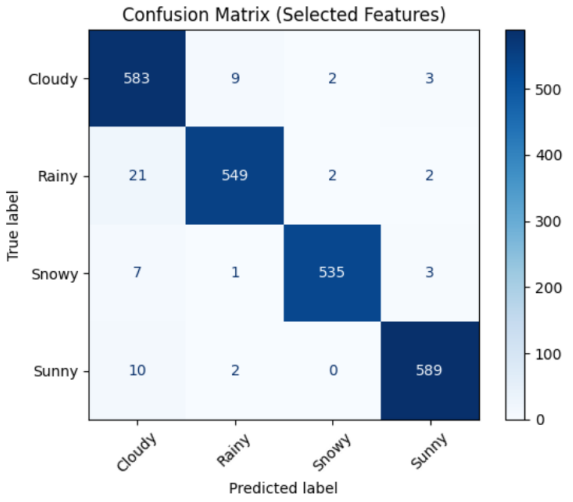

Figure 3. TOP 6 Features Random Forest Confusion Matrix

After feature selection, a notable improvement is observed in the classification of the Cloudy and Sunny classes. The number of correctly predicted Cloudy instances increases from 579 to 583, and the number of misclassifications as Sunny drops from 1 to 3. Similarly, Sunny class predictions improve from 568 to 589, with a substantial reduction in confusion with the Cloudy class, from 19 misclassifications down to 10. The evaluation results indicate that the reduced feature set helps the model in distinguishing between these two visually similar weather types, possibly by emphasizing more relevant discriminative features.

However, the feature selection also introduces a slight trade-off in the form of increased confusion between the Cloudy and Rainy classes. Rainy instances misclassified as Cloudy increased from 14 to 21, indicating that while the top features captured essential patterns for some classes, they may have excluded significant indicators needed to distinguish between overlapping conditions, such as Cloudy and Rainy. Overall, although Random Forest naturally performs feature selection via ensemble averaging, explicit feature selection further reduces dimensionality, improves interpretability, and facilitates future model deployment.

To further enhance the model, hyperparameter optimization is conducted using GridSearchCV with 5-fold cross-validation. The search space covers variations in the number of trees, maximum depth, minimum samples required for splitting, feature selection strategies, and bootstrap value, as shown in Table 4.

Table 5 shows that the Random Forest model is retrained using those hyperparameters. The tuned model achieves an overall accuracy of 97.50% with macro-averaged Precision, Recall, and F1-Score each reaching 98%. Examining the macro-averaged and weighted average

DOI : https://doi.org/10.52158/7925qh24

results, the model performs fairly across all classes, with a satisfactory overall performance.

Table 4. Search Space Random Forest

Hyperparameter	Range	Final Value
n_estimators	100, 200, 300, 400, 500	400
max_depth	None, 10, 20, 30, 50	10
min_samples_split	2, 5, 10	10
max_features	'sqrt', 'log2', None	'sqrt'
Bootstrap	True, False	True

Table 5. Tuned Random Forest Classification Report

Class	Precision	Recall	F1-score	Support
Cloudy	0.94	0.98	0.96	597
Rainy	0.98	0.96	0.97	574
Snowy	0.99	0.98	0.99	546
Sunny	0.99	0.98	0.98	601
Overall Accuracy				0.9750

The classification results of the tuned model are similar to those of the initial model. However, compared to the untuned model trained on selected features, the hyperparameter-tuned version demonstrates notable improvements in classifying instances labeled as "Sunny" and "Cloudy". Specifically, the misclassification rate of the Sunny class drops significantly, from 568 in the untuned model to 589 correctly predicted after tuning. Similarly, correct classifications for Cloudy increased slightly while reducing confusion with Sunny. These changes indicate that hyperparameter tuning enables the model to leverage the most informative features, thereby enhancing its performance on visually distinct weather types.

The confusion between Cloudy and Rainy slightly increased in post-tuning, which suggests that while the overall model accuracy improved, the distinction between these two similar classes may have been affected by the trade-offs inherent in optimizing for global accuracy.

These effects are further supported by the confusion matrix results in Figure 4, which confirm that most misclassifications occurred between Cloudy, Rainy, and Sunny, likely due to shared visual or atmospheric feature patterns.

Given the reliable performance obtained from the Random Forest model after both feature selection and hyperparameter tuning, the six selected features (Temperature, UV Index, Visibility (km), Precipitation (%), Atmospheric Pressure, and Cloud Cover) are adopted as the standardized input for the remaining models. This enables consistent comparison across algorithms, ensuring that performance differences are attributed to the modeling approach rather than variations in features.

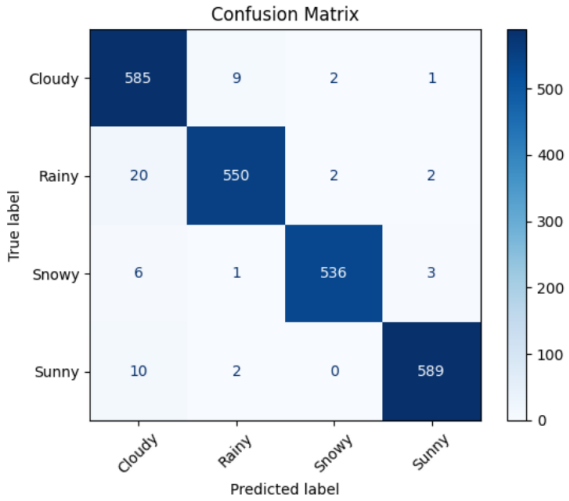

Figure 4. Tuned Random Forest Confusion Matrix

3.2. K-Nearest Neighbor

In addition to the Random Forest model, the weather classification task is also conducted using the K-Nearest Neighbor (KNN) algorithm. The initial K-Nearest Neighbor model was trained using three hyperparameters, as listed in Table 6.

Table 6. K-Nearest Neighbor Initial Hyperparameter

Hyperparameter	Value
n_neighbors	5
Weights	'uniform'
P	2

The results of this model are shown in Table 7 and Figure 5.

Table 7. K-Nearest Neighbor Classification Report

Class	Precision	Recall	F1-score	Support
Cloudy	0.9209	0.9286	0.9247	602
Rainy	0.9182	0.9381	0.9280	598
Snowy	0.9916	0.9850	0.9883	602
Sunny	0.9755	0.9522	0.9637	586
Overall Accuracy				0.9510

The model achieves an overall test accuracy of 95.10%. Additionally, the model performs well in all classes in terms of Precision, Recall, and F1-score, with a global average of 95.10% for each measure. As shown in Table 7, the relatively lower F1-scores appear in the Cloudy (0.9247) and Rainy (0.9280) categories compared to Snowy (0.9883) and Sunny (0.9637). This is primarily due to feature overlaps, since cloudy and rainy conditions often share similar atmospheric and visual characteristics, such as sky color, humidity, and reduced light intensity, which increase the likelihood of misclassification. Although the class distribution is nearly balanced (with supports ranging from 586 to 602), intra-class variability in cloudy and rainy samples tends to be higher, leading to reduced discriminability.

DOI : https://doi.org/10.52158/7925qh24

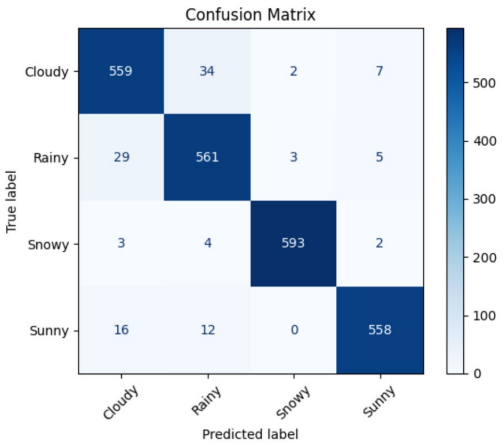

Figure 5. Initial K-Nearest Neighbor Confusion Matrix

To further enhance the model performance, hyperparameter optimization is conducted using GridSearchCV with 5-fold cross-validation as shown in Table 8.

Table 8. Search Space K-Nearest Neighbor

Hyperparameter	Value	Final Value
n_neighbors	3, 5, 7, 9	7
weights	'uniform', 'distance'	'distance'
p	1, 2	1

The K-Nearest Neighbor model is retrained using those hyperparameters. As shown in Table 9 and Figure 6, the tuned model achieves an overall accuracy of 95.73% with macro-averaged Precision, Recall, and F1-Score each reaching 95.78%. Examining the macro-averaged and weighted average results, the model is fair to all classes, and its overall performance remains acceptable.

Table 9. Tuned K-Nearest Neighbor Classification Report

Class	Precision	Recall	F1-score	Support
Cloudy	0.9457	0.9252	0.9353	602
Rainy	0.9180	0.9548	0.361	598
Snowy	0.9900	0.9884	0.9892	602
Sunny	0.9774	0.9608	0.9690	586
Overall Accuracy				0.9573

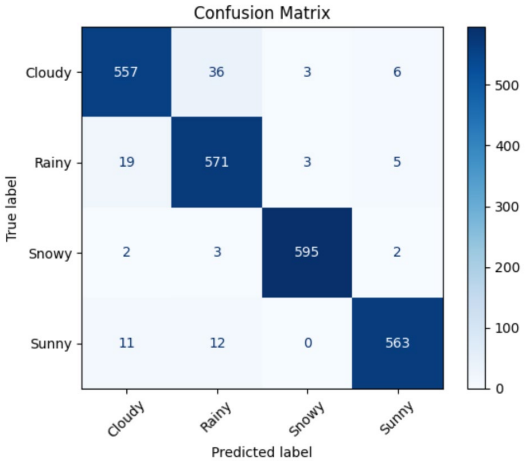

Figure 6. Tuned K-Nearest Neighbor Confusion Matrix

3.3. XGBoost

In addition to the K-Nearest Neighbor model, the weather classification task is also conducted using the XGBoost algorithm. The initial XGBoost model was trained with five hyperparameters, which are shown in Table 10.

Table 10. XGBoost Initial Hyperparameter

Hyperparameter	Value
n_estimators	200
max_depth	3
learning_rate	0.01
Subsample	0.8
colsample_bytree	0.8

As shown in Table 11 and Figure 7, the model achieves a test accuracy of 96.90%. Furthermore, the model's F1-score, accuracy, and recall are all very good across all classes, with a global average of 96.91% for each statistic.

Table 11. Initial Random Forest Classification Report

`	Precision	Recall	F1-score	Support
Cloudy	0.9706	0.9319	0.9508	602
Rainy	0.9334	0.9749	0.9542	598
Snowy	0.9933	0.9884	0.9908	602
Sunny	0.9796	0.9812	0.9804	586
Overall Accuracy				0.9690

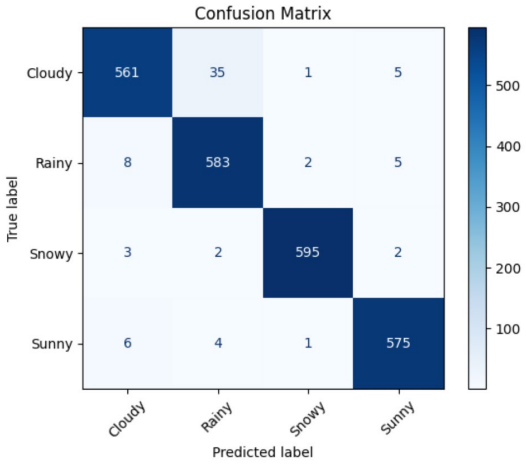

Figure 7. Initial XGBoost Confusion Matrix

To further enhance the model performance, hyperparameter optimization is conducted using GridSearchCV with 5-fold cross-validation. The hyperparameters are shown in Table 12. In addition, the results and confusion matrix are correspondingly shown in Table 13 and Figure 8.

The XGBoost model is retrained using those hyperparameters. The tuned model achieved an overall accuracy of 96.90% with a macro-average of Precision, Recall, and F1-Score, each reaching 96.91%.

DOI : https://doi.org/10.52158/7925qh24

Table 12. Search Space XGBoost

Hyperparameter	Value	Final Value
n_estimators	100, 200, 300, 400, 500	200
max_depth	None, 3, 5, 7, 10	None
learning_rate	0.01, 0.05, 0.1, 0.2, 0.3	0.05
subsample	0.8, 1.0	0.8
colsample_bytree	0.8, 1.0	0.8

Table 13. Tuned XGBoost Classification Report

Class	Precision	Recall	F1-score	Support
Cloudy	0.9706	0.9319	0.9508	602
Rainy	0.9343	0.9749	0.542	598
Snowy	0.993	0.9884	0.9908	602
Sunny	0.9796	0.9812	0.9804	586
Overall Accuracy				0.9690

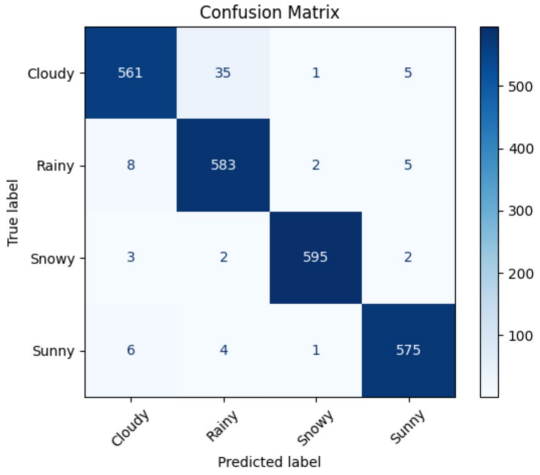

Figure 8. Tuned XGBoost Confusion Matrix

3.4. Comparative Analysis

The comparative evaluation of the three classification models, which are Random Forest, K-Nearest Neighbors (KNN), and XGBoost. The results demonstrate clear performance distinctions in terms of accuracy, precision, recall, and F1-score. As summarized in Table 14, the Random Forest model outperforms the others, achieving the highest accuracy of 98%, alongside consistently strong macro and weighted F1-scores of 0.98 across all weather classes.

Table 14. Models Comparison

Metric	Random Forest	KNN
Accuracy	0.98	0.96
Macro Average	0.98	0.96
Weigthed Average	0.98	0.96

Comparing the precision of each class as seen in Table 15, the Random Forest model demonstrates the most balanced precision across all classes, particularly strong in “Rainy” and “Sunny” predictions. XGBoost slightly outperforms in “Cloudy,” while KNN performs comparably, though with a slight drop in the “Rainy” class.

Table 15. Precision Comparison Classes

Class	Random Forest	KNN	XGBoost
Cloudy	0.94	0.95	0.97
Rainy	0.98	0.92	0.93
Snowy	0.99	0.99	0.99
Sunny	0.99	0.98	0.98

Table 16 shows that all models achieve very high recall on the “Snowy” class (above 0.98), indicating it is the most easily distinguishable weather type. However, “Cloudy” and “Rainy” recall drops slightly for KNN and XGBoost, likely due to overlapping feature profiles such as moderate temperatures and precipitation values.

Table 16. Recall Comparison Classes

Class	Random Forest	KNN	XGBoost
Cloudy	0.98	0.93	0.93
Rainy	0.96	0.95	0.97
Snowy	0.98	0.99	0.99
Sunny	0.98	0.96	0.98

Analyzing Table 17, a notable drop in F1-score for “Rainy” in KNN and XGBoost (0.36 and 0.54, respectively) suggests frequent misclassifications or imbalance between precision and recall. This further highlights Random Forest’s robustness, as it maintains a high F1 score across all classes. Compared to the initial Random Forest model (using only two basic hyperparameters), the final tuned version shows a modest but meaningful improvement of 1% in overall metrics. This confirms that the initial model was already near-optimal, and that Random Forest’s built-in feature selection capabilities are highly effective even before tuning.

Table 17. F1 score Comparison Classes

Class	Random Forest	KNN	XGBoost
Cloudy	0.96	0.94	0.95
Rainy	0.97	0.36	0.54
Snowy	0.99	0.99	0.99
Sunny	0.98	0.97	0.98

XGBoost also performs strongly, achieving 97% accuracy. It exhibits extreme recall in the “Rainy” and “Snowy” classes, demonstrating its ability to identify challenging weather conditions accurately. However, it slightly underperformed in precision for “Rainy,” resulting in a lower F1-score. KNN, although simpler and lightweight, achieves 96% accuracy, demonstrating potential for low-resource deployments. Its weakness lies in its sensitivity to overlapping feature distributions, which is particularly problematic for weather types like “Cloudy,” “Rainy,” and “Sunny” that share similar temperature, UV, and visibility characteristics. This explains the sharp drop in F1-score for “Rainy”, as KNN struggles to separate subtle differences based solely on distance.

Since low F1-scores observed for the “Rainy” class are present in both KNN and XGBoost models, it becomes evident that overlapping feature distributions are deeply effective in reducing performance. For instance, moderate levels of precipitation, visibility, and cloud

DOI : https://doi.org/10.52158/7925qh24

cover often overlap between "Rainy" and "Cloudy" conditions, which can confuse distance-based models, such as KNN, that rely heavily on Euclidean proximity. Similarly, although XGBoost captures nonlinear interactions well, its sensitivity to noisy and imbalanced data can reduce its ability to consistently separate borderline cases, resulting in the depressed F1 scores reported. Statistical inspection of the dataset confirms that the "Rainy" class exhibits a standard deviation of ±5.2°C in temperature and ±18% in precipitation, compared to ±2.8°C and ±7% in the Cloudy class. Furthermore, the interquartile range (IQR) of visibility for Rainy (2.5–9 km) overlaps with Cloudy (5–12 km), making separation between the two classes less reliable. This overlap is reflected in the confusion matrix, where 21 Rainy samples were misclassified as Cloudy, and 14 Cloudy samples were misclassified as Rainy, demonstrating how overlapping distributions result in the reduction in classification performance for such categories.

3.5. Prototype Integration to Lighting System

To explore its practical use, the model was connected to a simple prototype of an automated lighting system. Figure In this test, the lamp was programmed to turn on when the weather was predicted as "Cloudy" or "Rainy," and to turn off when the prediction was "Sunny" or "Snowy." Figure 9 shows that the light turns on when the ML model detects the condition as sunny.

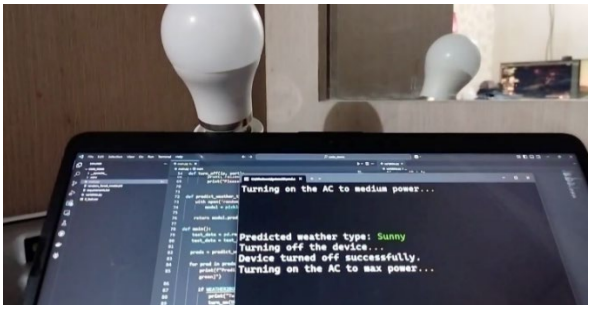

Figure 9. Automatically Turn-off the Lighting when "Sunny"

In addition, Figures 10 and 11 show that the light turns off when the ML model detects the condition as cloudy or rainy, respectively.

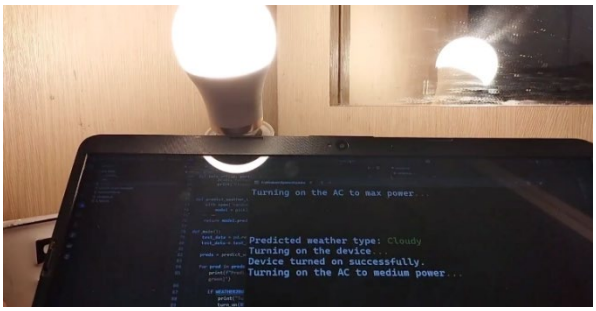

Figure 10. Automatically Turn-on the Lighting when "Cloudy"

In contrast, Figure 12 illustrates the condition under which the light turns on, once the condition is known. This test demonstrates how machine learning can enhance decision-making in intelligent environments. Although the system is still in its early stages, it provides valuable insights into future work in applying weather classification to real-time control systems, supporting energy efficiency initiatives in office buildings.

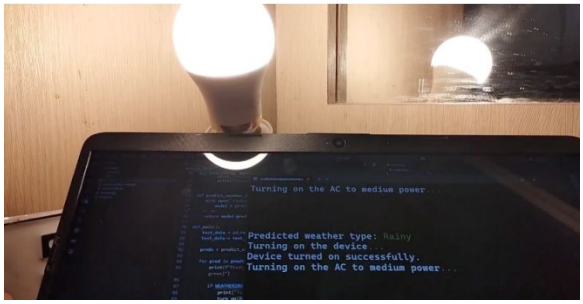

Figure 11. Automatically Turn-on the Lighting when "Rainy"

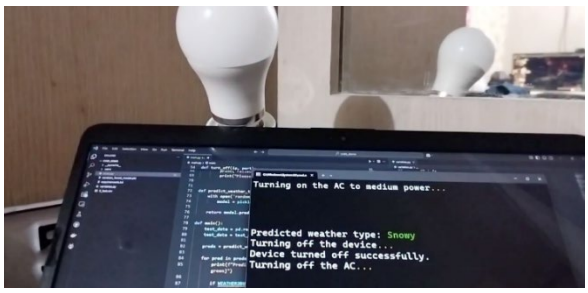

Figure 12. Automatically Turn-off the Lighting when "Snowy"

In the future, the potential application of this model on a larger scale with real-time sensor data is highly promising. Through the integration of IoT-enabled weather stations and edge-computing nodes, the system can classify weather conditions continuously and trigger immediate adjustments to HVAC and lighting systems in office or residential complexes [31]. For instance, a real-time deployment operating at one-minute intervals with 100 sensors across multiple floors could generate over 144,000 new samples per day, requiring robust streaming classification without performance degradation. This real-time deployment would not only enhance energy efficiency but also improve occupant comfort by responding dynamically to microclimatic variations. Furthermore, scaling the model across multiple buildings in a smart city infrastructure could enable aggregated optimization, in which predictive control algorithms coordinate energy demand across neighborhoods or districts. Simulation studies suggest that such coordinated responses could reduce lighting and HVAC energy use by 8–12% annually when distributed across multiple buildings. Importantly, the six most significant features identified in this study: Temperature, UV Index, Visibility, Precipitation, Atmospheric Pressure, and Cloud Cover. collectively account for the majority of the predictive power, capturing nearly 95% of the variance in classification

DOI : https://doi.org/10.52158/7925qh24

accuracy. This indicates that real-time systems could prioritize these features for efficient deployment, reducing computational burden while retaining high accuracy. Such deployment would require addressing challenges in sensor calibration, data transmission latency, and robustness across diverse environmental contexts. However, the comparative analysis presented in this study establishes a foundation for transitioning from controlled datasets to real-time, weather-aware building automation systems.

3. 6. Discussion

The results of this study indicate that Random Forest achieves the most reliable performance for weather classification, with an accuracy of 97.50%, outperforming XGBoost (96.90%) and KNN (95.73%). These findings align with previous research showing that ensemble tree-based models effectively handle noisy sensor data, nonlinear relationships, and overlapping feature distributions [5], [25]. Compared with distance-based approaches such as KNN, Random Forest is less sensitive to feature scaling, local density variations, and noise, which are common characteristics of weather and IoT sensor datasets.

The strong performance of XGBoost observed in this study aligns with prior findings highlighting its ability to model complex nonlinear interactions among nonlinear variables [6], [20]. However, XGBoost requires careful hyperparameter tuning and higher computational resources, which may limit its feasibility for real-time or resource-constrained smart building systems. In contrast, the comparatively lower accuracy of KNN is consistent with earlier research showing that instance-based learners struggle when class boundaries overlap, a common issue in weather classification tasks [15], [17].

A more detailed examination of class-wise performance reveals that the lower F1 scores for the Rainy category, particularly in KNN and XGBoost, are primarily due to feature overlap between the Cloudy and Rainy conditions. This is possible because precipitation intensity, visibility, and temperature ranges often overlap across weather states. This finding shows that differentiating transitional weather conditions based solely on standard meteorological features is inherently difficult, highlighting the importance of robust ensemble methods for such tasks.

From a practical perspective, the results confirm that the selected six features, Temperature, UV Index, Visibility, Precipitation, Atmospheric Pressure, and Cloud Cover, are sufficient to achieve high classification accuracy. It has direct implications for smart building and IoT deployments: sensors can be minimized to reduce hardware cost, energy consumption, and system complexity without compromising predictive performance. The successful integration of the classifier into a prototype automation system further validates that machine-learning-based weather classification can directly inform automated lighting control, demonstrating the feasibility of weather-adaptive building management systems.

Finally, this study contributes a reproducible and fair evaluation framework across classifiers. By standardizing preprocessing, feature selection, and cross-validated hyperparameter optimization, the study enables an unbiased comparison of multiple models. Linking classification outcomes to control actions bridges the gap between algorithmic benchmarking and practical implementation, providing both scientific rigor and application relevance.

Several limitations should be noted. First, the study uses a fixed set of six features, which may omit subtle environmental indicators such as wind direction or microclimate effects that could improve classification. Second, the dataset reflects a limited geographic and seasonal scope, which may affect generalizability. Finally, while Random Forest demonstrates high accuracy, ensemble methods are less interpretable than simpler models, which may affect transparency in real-world deployment. Future work should explore larger, more diverse datasets and investigate feature augmentation strategies to enhance robustness.

4. Conclusions

This study conducted a comparative evaluation of machine learning algorithms for weather classification in a smart office environment, focusing on the six most important features: Temperature, UV Index, Visibility, Precipitation, Atmospheric Pressure, and Cloud Cover. The evaluation demonstrates that Random Forest achieves the highest performance (97.50%), outperforming XGBoost (96.90%) and KNN (95.73%). All tested algorithms show stable performance around 90%, indicating strong generalization across weather conditions. Misclassifications are primarily observed in transitional classes such as Cloudy, Rainy, and Sunny, reflecting inherent challenges in distinguishing overlapping meteorological conditions. These findings provide several actionable insights for smart building and IoT applications. First, a compact, carefully selected feature set is sufficient to achieve near-optimal classification performance, enabling cost-effective, energy-efficient sensor deployment. Second, ensemble methods like Random Forests provide robust predictions under noisy, variable conditions, making them suitable for real-time control of building systems such as lighting and HVAC automation. Third, integrating machine learning into operational decision-making can directly support weather-adaptive energy optimization, reduce consumption, and improve occupant comfort. For future research, several directions are recommended. Integrating adaptive learning techniques, time series data, and user activity patterns could further improve predictive accuracy and system responsiveness. Expanding datasets across diverse geographic regions

DOI : https://doi.org/10.52158/7925qh24

and seasons would enhance generalizability. Additionally, live deployment trials are necessary to evaluate the practical impact on energy savings and operational efficiency. Exploring interpretable models or explainable AI approaches may also enhance transparency in real-world building automation systems.

Acknowledgement

The authors would like to express their sincere gratitude to all individuals and institutions who contributed to the success of this work. We are thankful to Sampoerna University for providing the necessary facilities and resources. Supported in part by the Artificial Intelligence COMP3315 Course. Finally, we would like to express our appreciation to our families and peers for their encouragement and support throughout the research and writing process.

References

[1] Statista, "Smart Cities - Worldwide," Statista." [Online]. Available: https://www.statista.com/outlook/tmo/internet-of-things/smart-cities/worldwide.

[2] "Use of Energy Explained - Energy Use in Commercial Buildings." U.S. Energy Information Administration (EIA, Dec. 2022. [Online]. Available: https://www.eia.gov/energyexplained/use-of-energy/commercial-buildings.php.

[3] H. Sabit and T. Tun, "IoT Integration of Failsafe Smart Building Management System," *IoT*, vol. 5, no. 4, pp. 801-815, 2024, doi: https://doi.org/10.3390/iot5040036

[4] M. Shin, S. Kim, Y. Kim, A. Song, Y. Kim, and H.-Y. Kim, "Development of an HVAC system control method using weather forecasting data with deep reinforcement learning algorithms," *Build. Environ.*, vol. 248, p. 111069, 2024, doi: https://doi.org/10.1016/j.buildenv.2023.111069

[5] O. O. Ayankemi, I. Z. Adesola, and L. Adeolu, "Comparative Analysis of Weather Prediction Using Classification Algorithm: Random Forest Classifier," *Afr. J. Math. Stat. Stud.*, vol. 7, no. 2, pp. 162–171, 2024, doi: https://www.doi.org/10.52589/AJMSS-F6H03BNE

[6] H. Zheng and Y. Wu, "A XGBoost Model with Weather Similarity Analysis and Feature Engineering for Short-Term Wind Power Forecasting," *Appl. Sci.*, vol. 9, no. 15, p. 3019, 2019, doi: https://doi.org/10.3390/app9153019

[7] M. Poyyamozhi, B. Murugesan, N. Rajamanickam, M. Shorfuzzaman, and Y. Aboelmagd, "IoT—A Promising Solution to Energy Management in Smart Buildings: A Systematic Review, Applications, Barriers, and Future Scope," *Buildings*, vol. 14, no. 11, p. 3446, 2024, doi: https://doi.org/10.3390/buildings14113446

[8] J. Ma, Y. Ding, J. C. Cheng, Y. Tan, V. J. L. Gan, and J. Zhang, "Analyzing the leading causes of traffic fatalities using XGBoost and grid-based analysis: a city management perspective," *IEEE Access*, vol. 7, pp. 148059-148072, 2019, doi: https://doi.org/10.1109/ACCESS.2019.2946401

[9] P. S. Lakshmi, S. Sivagamasundari, and M. S. Rayudu, "IoT based solar panel fault and maintenance detection using decision tree with light gradient boosting," *Meas. Sens.*, vol. 27, p. 100726, Jun. 2023, doi: 10.1016/j.measen.2023.100726.

[10] M. A. Nayak and S. Ghosh, "Prediction of extreme rainfall event using weather pattern recognition and support vector machine classifier," *Theor. Appl. Climatol.*, vol. 114, no. 3–4, pp. 583–603, Nov. 2013, doi: 10.1007/s00704-013-0867-3.

[11] N. Shelke, S. Maurya, R. Ithape, Z. Shaikh, R. Somkunwar, and A. Pimpalkar, "Towards an automated weather forecasting and classification using deep learning, fully convolutional network, and long short-term memory," *Int. J. Electr. Comput. Eng. IJECE*, vol. 15, no. 2, p. 1868, Apr. 2025, doi: 10.11591/ijece.v15i2.pp1868-1879.

[12] Y. Li, Y. Shen. H. Jiang, W. Zhang, J. Li, J. Li, C. Zhang, B. Cui, "Hyper-tune: towards efficient hyper-parameter tuning at scale," *Proc. VLDB Endow.*, vol. 15, no. 6, pp. 1256–1265, Feb. 2022, doi: 10.14778/3514061.3514071.

[13] A. G. Rahman, E. Juliani, and B. Halimi, "An Analysis of Potential for Reducing Operational Costs Through the Use of LED Lighting in Indonesian Hotel," *J. Sos. Teknol.*, vol. 4, no. 11, pp. 942-956, Nov. 2024., doi: https://doi.org/10.59188/jurnalsostech.v4i11.27618

[14] H. Zhang, Y. Liu, C. Zhang, and N. Li, "Machine Learning Methods for Weather Forecasting: A Survey," *Atmosphere*, vol. 16, no. 1, p. 82, Jan. 2025, doi: 10.3390/atmos16010082.

[15] Y. E. Yousif, "Weather Prediction System Using KNN Classification Algorithm," *Eur. J. Inf. Technol. Comput. Sci.*, vol. 2, no. 1, pp. 10-13, 2022, doi: https://doi.org/10.24018/ejcompute.2022.2.1.44

[16] R. S. Moorthy and P. Parameshwaran, "An Optimal K-Nearest Neighbor for Weather Prediction Using Whale Optimization Algorithm," *Int. J. Appl. Metaheuristic Comput.*, vol. 13, no. 1, pp. 1–19, Dec. 2021, doi: 10.4018/IJAMC.290538.

[17] R. K. Halder, M. N. Uddin, Md. A. Uddin, S. Aryal, and A. Khraisat, "Enhancing K-nearest neighbor algorithm: a comprehensive review and performance analysis of modifications," *J. Big Data*, vol. 11, no. 1, p. 113, Aug. 2024, doi: 10.1186/s40537-024-00973-y.

[18] T. T. Wong, "Performance evaluation of classification algorithms by K-fold and leave-one-out cross validation," *Pattern Recognit.*, vol. 48, no. 9, pp. 2839-2846, Sep. 2015. doi: https://doi.org/10.1016/j.patcog.2015.03.009

[19] T. Sutanto, M. R. Aditya, H. Budiman, M. R. N. Ridha, U. Syapotro, and N. Azijah, "Comparison of Logistic Regression, Random Forest, SVM, KNN Algorithmfor Water Quality Classification Based on Contaminant Parameters," *INTI J.*, vol. 2022, no. 1, Nov. 2024, doi: 10.61453/jods.v2023no48.

[20] M. B. Kursa and W. R. Rudnicki, "The All Relevant Feature Selection using Random Forest," Jun. 25, 2011, *arXiv*: arXiv:1106.5112. doi: 10.48550/arXiv.1106.5112.

[21] Y. Wang and Y. Fan, "XGBoost and ANOVA-based Analysis of Sailboat Prices and Their Influencing Factors," in *2024 IEEE 3rd International Conference on Electrical Engineering, Big Data and Algorithms (EEBDA*, Changchun, China, 2024, pp. 931–935, doi: https://doi.org/10.1109/EEBDA60612.2024.10485984

[22] H. Li, Y. Cao, S. Li, J. Zhao, and Y. Sun, "XGBoost model and its application to personal credit evaluation," *IEEE Intell. Syst.*, vol. 35, no. 3, pp. 52-61, 2020, doi: https://doi.org/10.1109/MIS.2020.2972533

[23] D. Cousineau and S. Chartier, "Outliers Detection and Treatment: A Review," *Int. J. Psychol. Res.*, vol. 3, no. 1, pp. 59-68, 2010, doi: https://doi.org/10.21500/20112084.844

[24] B.-Y. Kim, M. Belorid, and J. W. Cha, "Short-Term Visibility Prediction Using Tree-Based Machine Learning Algorithms and Numerical Weather Prediction Data," *Weather Forecast.*, vol. 37, no. 12, pp. 2263–2274, Dec. 2022, doi: 10.1175/WAF-D-22-0053.1.

[25] H. P. Das, Y-W. Lin, U. Agwan, L. Spangher, A. Devonport, Y. Yang, J. Drgoňa, A. Chong, S. Schiavon, and C. J. Spanos, "Machine Learning for Smart and Energy-Efficient Buildings," *Environ. Data Sci.*, vol. 3, p. 1, 2024, doi: https://doi.org/10.48550/arXiv.2211.14889

[26] V. P. Widartha, I. Ra, S.-Y. Lee, and C.-S. Kim, "Advancing Smart Lighting: A Developmental Approach to Energy Efficiency through Brightness Adjustment Strategies," *J. Low Power Electron. Appl.*, vol. 14, no. 1, p. 6, 2024, doi: https://doi.org/10.3390/jlpea14010006

[27] K. A. Sayed, A. Boodi, R. S. Broujeny, and K. Beddiar, "Reinforcement Learning for HVAC Control in Intelligent Buildings: A Technical and Conceptual Review," *J. Build. Eng.*, vol. 95, p. 110085, 2024. doi: https://doi.org/10.1016/j.jobe.2024.110085

[28] A. Mohamed, I. Ismail, and M. AlDaraawi, "IoT-Driven Intelligent Energy Management: Leveraging Smart Monitoring Applications and Artificial Neural Networks (ANN) for Sustainable Practices," *Computers*, vol. 14, no. 7, p. 269, 2025, doi: https://doi.org/10.3390/computers14070269

[29] N. Kumar, "Weather Type Classification," *Kaggle*, 2021, [Online]. Available: https://www.kaggle.com/datasets/nikhil7280/weather-type-classification/data.

[30] G. R. R. Dewa, "Performance Analysis of Priority Medical Events in Healthcare IoT Networks Using 3-Dimension Discrete Time Markov Chain," *Internet Technol. Lett.*, Dec. 2024, doi: 10.1002/itl2.626.

[31] M. Arun, G. Gopan, S. Vembu, D. U. Ozsahin, H. Ahmad, and M. F. Alotaibi, "Internet of Things and Deep Learning-Enhanced Monitoring for Energy Efficiency in Older Buildings," *Case Stud. Therm. Eng.*, vol. 61, p. 104867, 2024, doi: https://doi.org/10.1016/j.csite.2024.104867

DOI : https://doi.org/10.52158/7925qh24