



# Adaptive Hybrid Stacking Ensemble Learning for Real-Time Forest Fire Risk Prediction

Yuda Irawan<sup>1</sup>, Sarjon Defit<sup>2</sup> and Rini Sovia<sup>3</sup>

## ABSTRACT

Forest and land fires remain a recurring environmental issue, particularly in regions with high climate variability such as Indonesia. This study proposes an Adaptive Hybrid Stacking Ensemble approach for real-time fire risk prediction by integrating meteorological data and IoT-based sensing systems. The model combines multiple base learners with a rotation strategy and an exponential accuracy weighting mechanism to optimize the contribution of each model. A dataset consisting of 4,931 meteorological records was used, including variables such as Average Temperature, Relative Humidity, Rainfall, Wind Speed, and Soil Surface Moisture. Experimental results show that the proposed model significantly outperforms conventional stacking, achieving an accuracy of up to 92.47%, compared to 86%–87% in baseline stacking. The optimal performance was obtained at  $\alpha = 5$  in the weighting scheme, which effectively balances model contribution. Evaluation using confusion matrix, ROC curve, and K-fold cross-validation confirms the model's robustness, with AUC values reaching 0.98–1.00 and stable accuracy across folds. In addition, Explainable AI using SHAP reveals that Rainfall and Soil Surface Moisture are the most influential features in fire risk prediction. The integration with IoT enables real-time data processing and prediction, making the system applicable for early warning and disaster mitigation.

## Article information:

**Keywords:** Stacking Ensemble, Accuracy Weighting, Forest Fire Prediction, Internet of Things, Explainable AI

## Article history:

Received: March 13, 2026

Revised: July 23, 2026

Accepted: August 20, 2026

Published: September 26, 2026

(Online)

DOI: 10.37936/ecti-cit.2026204.266684

## 1. INTRODUCTION

Forest and land fires are still a recurring environmental problem in Indonesia, especially in areas such as Riau Province which experiences similar incidents almost every year[1], [2], [3]. This event not only causes ecosystem damage, but also causes major economic losses, public health problems due to haze, and extensive social impacts. As environmental monitoring technology develops, various systems have been developed to monitor meteorological conditions in real time that are directly related to the potential for fires[4].

The development of Internet of Things technology enables the use of environmental sensors for continuous monitoring[5]. Various studies have utilized IoT devices with temperature, humidity, rainfall, and gas sensors to detect early signs of fire[6], [7], [8].

Furthermore, integration with communications and cloud computing technologies has enhanced the system's ability to transmit and store data in real time. However, IoT-based systems are generally limited to data acquisition and threshold-based detection, thus failing to capture complex patterns in fire risk prediction.

To address these limitations, machine learning approaches have begun to be widely used to improve prediction accuracy. Various algorithms such as Decision Trees, Random Forests, Support Vector Machines, and Neural Networks have been applied to fire prediction[9], [10], [11]. Although they have shown quite good results, the performance of these models is highly dependent on data quality and often struggles to handle nonlinear relationships between meteorological variables. Furthermore, the issue of data

<sup>1</sup>The author is with the Faculty of Computer Science, Universitas Hang Tuah Pekanbaru, Pekanbaru, 28288, Indonesia, Email: yudairawan89@gmail.com

<sup>2,3</sup>The authors are with the Faculty of Computer Science, Universitas Putra Indonesia YPTK Padang, Padang, 25147, Indonesia, Email: sarjon\_defit@upiyptk.ac.id and rini\_sovia@upiyptk.ac.id

<sup>1</sup>Corresponding author: yudairawan89@gmail.com

imbalance is often overlooked, where high-risk classes have significantly less data than other classes, leading to biased predictions.

The Stacking Ensemble Learning approach developed as a method for combining multiple models to improve predictive performance[12], [13], [14]. By utilizing base learners and meta learners, stacking can integrate the advantages of various algorithms. However, several studies have shown that conventional stacking approaches do not always produce better performance than individual models. In some cases, meta models actually have lower accuracy than the best base model[15], [16], [17], [18]. In addition, stacking models are generally black-box in nature and therefore difficult to interpret, which is an obstacle in their application to decision making[19].

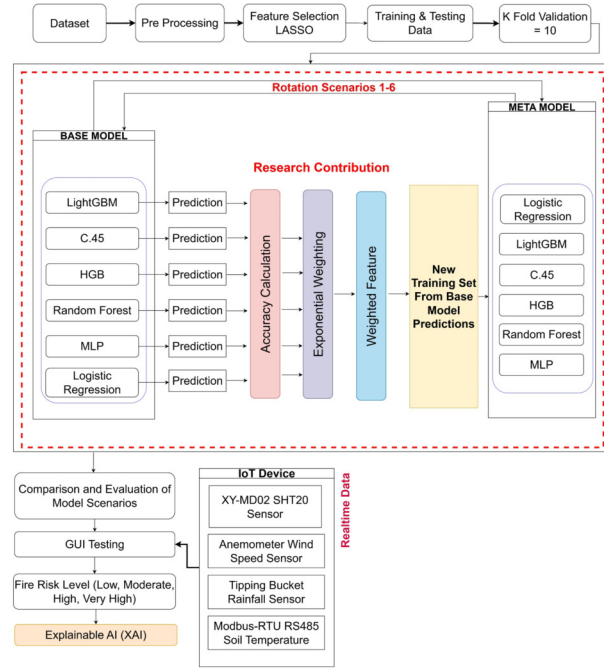
Another quite important problem is the absence of an adaptive weighting mechanism in determining the contribution of each base model[20]. Without such mechanisms, model stacking tends to fail to optimize the role of high-performing models and remains influenced by low-performing models. Furthermore, the limited application of explainable artificial intelligence also limits understanding of the factors that most influence wildfire risk prediction[21].

The novelty of the proposed Adaptive Hybrid Stacking Ensemble lies in the integration of three complementary mechanisms within a single stacking framework: (1) a rotational strategy that systematically evaluates each learning algorithm as both base learner and meta learner, (2) an adaptive exponential accuracy weighting mechanism that dynamically adjusts the contribution of each base model according to its predictive performance, and (3) the incorporation of SHAP based explainable artificial intelligence to provide transparent interpretation of prediction results. Unlike conventional stacking methods that rely on fixed model roles and equal model contributions, the proposed approach improves both predictive performance and model interpretability.

## 2. MATERIALS AND METHODS

This study develops a new approach in Stacking Ensemble Learning called Adaptive Hybrid Stacking Ensemble. This model is designed to improve predictive performance in detecting fire risk levels by implementing a rotation mechanism between models in the learning stage, as well as using an accuracy-based weighting scheme in the base model. This adaptive approach allows each model to contribute according to its performance level, resulting in more stable and accurate predictions. The flowchart of the model development proposed in this study is presented in Figure 1.

Figure 1 presents the development flow of the Adaptive Hybrid Stacking Ensemble model, designed to improve the accuracy and stability of fire risk prediction. The process begins with dataset collection



**Fig.1: Proposed Model.**

and then goes through a preprocessing stage, including data cleaning and normalization using Standard-Scaler to ensure each feature is at a uniform scale. Next, feature selection is performed using the LASSO method to identify the most relevant variables.

In the modeling stage, several base models are used, including LightGBM, C4.5, HistGradientBoosting, Random Forest, MLP, and Logistic Regression. The six learning algorithms were intentionally selected because they represent diverse machine learning paradigms with complementary learning characteristics. LightGBM and HistGradientBoosting were included for their efficiency in capturing complex nonlinear relationships through gradient boosting. Random Forest provides robust ensemble learning with strong generalization capability and resistance to overfitting. C4.5 contributes an interpretable decision tree structure that effectively handles structured data. MLP was selected to model nonlinear feature interactions through neural network learning, while Logistic Regression serves as a simple linear classifier that provides a strong baseline and complementary decision boundaries. The diversity of these algorithms increases prediction variability among base learners, which is beneficial for stacking ensemble performance. Each model produces predictions that are then evaluated through accuracy calculations. These accuracy values are used in an exponential weighting mechanism, so that models with higher performance receive greater weight. The resulting weighting forms new features (weighted features) as input to the meta-model.

To enhance the exploration of model combinations, scenario rotation (1–6) is applied, where each algo-

rithm alternately acts as the meta-model, while the other model serves as the base model. This approach allows for systematic identification of the best configuration. The model is then validated using K-Fold Cross Validation. The prediction results are integrated with the IoT system in real-time, and analyzed using SHAP to provide an interpretation of the feature contribution to the prediction results.

## 2.1 Dataset and preprocessing

This research began with the collection of meteorological data from the BMKG (Indonesia's Meteorology, Climatology, and Geophysics Agency) for the period 2011 to 2024, which was then used as the main dataset in this study with a total of 4,931 data. The preprocessing stage was carried out to ensure optimal data quality before being used in the model training process. This process includes several important stages in structured data, namely data cleaning to address missing or inconsistent values, class balancing to address imbalanced data distribution, feature standardization using normalization techniques, and feature selection to select the most relevant variables for the prediction target.

### 1) Feature selection

Feature selection is a crucial step in data preprocessing, aiming to determine the subset of features that are most relevant and influential to the prediction target[22]. By eliminating insignificant or redundant features, the model training process can be more efficient, computational complexity can be reduced, and the model's generalizability to new data can be improved. Based on this process, five key features were identified that have the greatest impact on fire risk: Tavg (Daily average temperature), RH\_avg (Daily average air humidity), RR (Daily rainfall), ff\_avg (Daily average wind speed), and Soil Surface Moisture.

Feature selection was performed using L1-regularized Logistic Regression (LASSO), which is suitable for multiclass classification problems by enforcing sparsity in the model coefficients. Features with non-zero coefficients were retained for subsequent model training. This approach reduces redundant variables while preserving the most informative features for the classification task.

### 2) StandardScaler

The StandardScaler step is used to adjust the scale of all features so that they have a distribution with a mean value close to 0 and a standard deviation of 1[23][24]. This step is necessary because the data used contains variables with different value ranges, for example, temperature in degrees Celsius and wind speed in meters per second. These differences in ranges have the potential to cause imbalances in the model's learning process if not adjusted. Therefore, the application of feature scaling aims to equalize the contribution of each feature in the training process,

while simultaneously improving the performance of machine learning algorithms that are sensitive to differences in data scale.

## 2.2 Stacking ensemble learning

Stacking Ensemble Learning is an ensemble learning method that combines multiple predictive models to improve classification performance[12]. This approach consists of two main levels: a base learner at the first level and a meta learner at the second level. At the first level, several base models are trained using the same dataset to generate initial predictions. The output from each model is then used as new features that become input for the next-level model[25].

Mathematically, if there is  $N$  basic models  $h_1, h_2, h_3, \dots, h_N$ , then for each input data  $x$ , each model produces a prediction:

$$z = [h_1(x), h_2(x), \dots, h_N(x)] \quad (1)$$

The prediction vector  $z$  is then used as input for the meta learner  $H$  to produce the final prediction output.

$$\hat{y} = H(z) = H(h_1(x), h_2(x), \dots, h_N(x)) \quad (2)$$

Through this mechanism, stacking can leverage the strengths of each base model while mitigating their individual weaknesses. In this study, the stacking approach was combined with a model rotation mechanism and accuracy-based weighting to improve stability and predictive performance. The base models used included LightGBM, C4.5, HistGB, Random Forest, MLP, and Logistic Regression. The resulting models were not only more accurate but also more adaptive to data variations and able to capture complex patterns from the various characteristics of the models used.

## 2.3 Rotational model strategy

The rotational model strategy is an approach used to enhance the exploration of model combinations within a stacking ensemble framework. In this method, each algorithm alternately acts as a meta-learner, while the other model serves as a base learner. This creates multiple rotation scenarios, allowing for a more comprehensive and objective evaluation of model performance.

Formally, if there is a set of models  $M = \{m_1, m_2, \dots, m_k\}$ , then in every scenario to  $-r$ , one model  $m_r$  selected as meta learners, while the rest become base learners:

$$B_r = M/\{m_r\}, H_r = m_r \quad (3)$$

Each base learner produces predictions on the input data  $x$ , which are then combined into a prediction vector:

$$z_r = [h_1(x), h_2(x), \dots, h_{k-1}(x)] \quad (4)$$

The vector  $z_r$  is then used as input for the meta learner  $H_r$  to produce the final prediction:

$$\hat{y}_r = H_r(z_r) \quad (5)$$

To systematically evaluate the contribution of each learning algorithm, the proposed framework employs a complete rotation strategy. Given six learning algorithms, six rotation scenarios are generated. In each scenario, one algorithm is assigned as the meta-learner, while the remaining five algorithms are used as base learners. This process is repeated until every algorithm has served exactly once as the meta-learner. No meta-model is selected based on predictive performance or random selection; instead, all candidate algorithms are evaluated under identical experimental conditions. The final model performance is then compared across all scenarios to identify the most effective meta-learner configuration.

## 2.4 Accuracy weighting

The accuracy weighting method is used to adjust the contribution of each base model in the stacking process based on its performance level. This approach aims to give greater weight to models with high accuracy, thus optimizing the prediction aggregation results[26]. In this study, the weighting was performed using an exponential function to amplify the differences in contributions between models.

Mathematically, the weights for each model are  $-x$  calculated as follows:

$$w_x = \frac{e^{\alpha \cdot acc_x}}{\sum_{j=1}^{|M_r|} e^{\alpha \cdot acc_j}} \quad (6)$$

where  $w_x$  is the weight of the model to  $-x$ ,  $acc_x$  is the prediction accuracy value of the model, and  $\alpha$  is a constant that controls the sensitivity to differences in accuracy, which in this study is set at 5. The notation  $|M_r|$  represents the number of active base models in the  $r$ -th rotation scenario, while  $e$  is an exponential number with a value close to 2.718. The obtained weights are then used to form a weighted composite feature as follows:

$$z = \sum_{x=1}^{|M_r|} w_x \cdot h_x(x) \quad (7)$$

This exponential approach can magnify the difference in contributions between models. For example, two models with accuracies of 0.90 and 0.85 differ by only 0.05. However, with  $\alpha = 5$ , the exponential values become  $e^{4.5} \approx 90$  and  $e^{4.25} \approx 70$ , respectively making the weight difference more significant. This allows the higher-performing model to contribute more significantly to the final prediction.

## 2.5 K-fold cross-validation

K-fold cross-validation is a model evaluation technique used to measure model performance more accurately and reduce the risk of overfitting[27]. This method works by dividing the dataset into K equal parts (folds). At each iteration, one fold is used as test data, while the other K-1 folds are used as training data. This process is repeated K times so that each data section has an equal chance of being used as test data.

In this study, a value of K=10 was used, known as 10-Fold Cross Validation, because it balances bias and variance in model evaluation. Each iteration produces a different accuracy value, which is then averaged to obtain an estimate of the overall model performance.

Mathematically, the average accuracy from the K-fold cross-validation process can be formulated as follows:

$$Acc_{cv} = \frac{1}{K} \sum_{i=1}^K Acc_i \quad z = \sum_{x=1}^{|M_r|} w_x \cdot h_x(x) \quad (8)$$

where  $Acc_{cv}$  is the average overall accuracy,  $Acc_i$  is the accuracy value at the  $i$ -th fold, and  $K$  is the number of folds used. With this approach, all data is optimally used for both training and testing data. This allows the model to have better generalization capabilities to new data and provides a more representative evaluation of the model's overall performance.

## 2.6 Explainable ai (XAI)

Explainable Artificial Intelligence (XAI) is used to increase model transparency in explaining the contribution of each feature to the prediction results[28], [29], [30]. In this study, the XAI approach is implemented using the SHapley Additive Explanations (SHAP) method, which is capable of providing both global and local interpretations of model decisions[31], [32]. Mathematically, SHAP bases its calculations on the Shapley value from cooperative game theory, which is formulated as follows:

$$f(x) = \emptyset_0 + \sum_{i=1}^n \emptyset_i \quad (9)$$

where  $f(x)$  is the model prediction output,  $\emptyset_0$  is the baseline value, and  $\emptyset_i$  indicates the contribution of each  $i$ -th feature to the prediction. The value of  $\emptyset_i$  is calculated by considering all possible feature combinations, thus providing a fair estimate of the contribution.

In its implementation, TreeExplainer is used in the Random Forest model to calculate SHAP values based on data that has undergone feature selection and standardization. The results of the SHAP analysis are visualized in the form of a summary plot to show the global level of feature importance, as well

as a beeswarm plot to identify the distribution of feature contributions to specific risk classes, particularly the high-risk category.

Through this approach, the model not only produces accurate predictions but also provides a clear interpretation of the key factors influencing fire risk levels, thereby increasing user trust and understanding of the system.

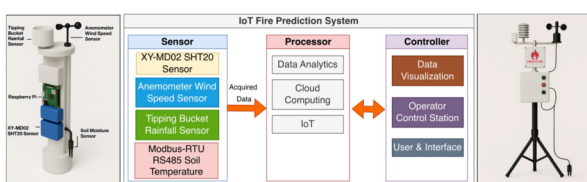
## 2.7 Evaluation

A comprehensive evaluation of model performance was conducted using several metrics and visualization techniques to ensure model accuracy, stability, and generalization. A classification report was used to measure model performance based on precision, recall, and F1-score values for each risk class[33]. Furthermore, a confusion matrix was used to observe the distribution of correct and incorrect predictions within each class, thus determining the model's ability to differentiate fire risk levels[34].

Further analysis was conducted using a Receiver Operating Characteristic (ROC) curve to evaluate the model's ability to separate classes by examining the relationship between the true positive rate and false positive rate[35]. Scatter plot visualization was also used as an initial exploration to understand the relationship between features and risk labels. To identify the contribution of each feature, the permutation feature importance method was used, which measures the decline in model performance when a feature is shuffled. Furthermore, learning curves were used to analyze the relationship between the amount of training data and model performance, thereby detecting potential overfitting or underfitting. The combination of these various evaluation methods provides a more comprehensive picture of the developed model's performance.

## 2.8 IoT device design

IoT devices are used to acquire real-time meteorological data, which is then processed using the Adaptive Hybrid Stacking Ensemble model. This system is designed to collect various environmental parameters directly from sensors, enabling continuous and accurate condition monitoring. The obtained data is then sent and processed as input in the fire risk prediction process. The design and architecture of the IoT devices used in this study are shown in Figure 2.



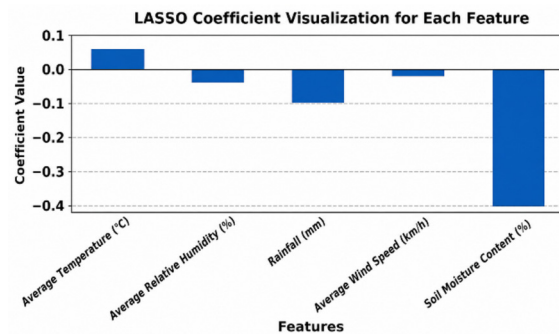
**Fig.2:** IoT Device Model and Design Elements.

In the sensor section, there are four main types of sensors used to collect meteorological data. The XY-MD02 SHT20 sensor is used to measure air temperature and humidity with a high level of accuracy and is capable of operating in extreme environmental conditions. An anemometer sensor is used to monitor wind speed, which is a crucial factor in accelerating the spread of fire. Furthermore, a tipping bucket rainfall sensor functions to measure rainfall intensity, which plays a role in determining the level of fire risk, especially during the dry season. In addition, a soil sensor is used as an additional indicator to estimate soil surface moisture. This variable is important because it is often overlooked in previous studies, even though it has a significant influence on the potential for fires.

## 3. RESULT AND DISCUSSION

### 3.1 Dataset and preprocessing results

The dataset used in this study consists of 4,931 meteorological data obtained from the BMKG, covering various environmental parameters that influence fire risk. The dataset then underwent preprocessing, including data cleaning, feature standardization using StandardScaler, and feature selection to improve data quality before the modeling process. A crucial step in preprocessing is feature selection using the LASSO method to identify the most relevant variables. Figure 3 presents a visualization of the LASSO coefficients for each feature used in this study.

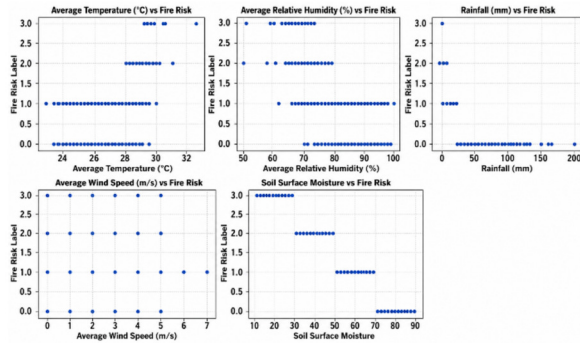


**Fig.3:** Lasso Coefficient Visualization Results.

Based on Figure 3, the Soil Surface Moisture feature has the largest negative coefficient, at around -0.40, indicating that lower soil surface moisture tends to significantly increase fire risk. The Rainfall and Average Relative Humidity features also show negative coefficients of around -0.10 and -0.04, respectively, indicating that increased rainfall and air humidity contribute to reducing fire risk. Conversely, Average Temperature has a positive coefficient of around 0.05, indicating that rising temperatures have the potential to increase fire risk. Meanwhile, Average Wind Speed shows a relatively small effect, with a coefficient value approaching 0.01, but it is retained

due to its role in accelerating fire spread. These findings indicate that variables related to humidity and rainfall have a more dominant influence on determining fire risk levels than other variables.

To analyze the relationship between input variables and fire risk labels, a scatter plot was used as an initial exploratory step before the modeling process. This visualization aims to observe data distribution patterns and identify possible linear and non-linear relationships between selected features and fire risk levels. The results of the scatter plot analysis are shown in Figure 4.

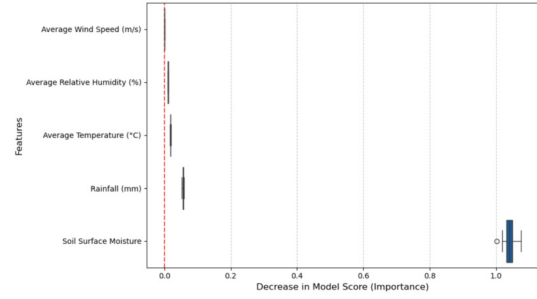


**Fig.4:** Scatter Plot of the Relationship between Selected Features and Fire Risk Labels.

Based on Figure 4, the scatter plot shows a fairly clear relationship between input variables and fire risk. Average Temperature tends to increase with increasing risk levels, especially at values above  $\pm 28^{\circ}\text{C}$ . Conversely, Average Relative Humidity and Rainfall show a negative relationship, with high humidity values ( $> 80\%$ ) and high rainfall correlated with low fire risk, while low values were more prevalent at high risk. Average Wind Speed does not show a strong linear pattern, but still plays a supporting role in fire spread. Meanwhile, Soil Surface Moisture has the most significant negative relationship, with low values ( $< 40$ ) appearing more frequently at high risk and high values ( $> 70$ ) predominating at low risk. These findings indicate that humidity and rainfall variables have the strongest influence on fire risk.

To identify the contribution of each feature to model performance, the Permutation Feature Importance method was used as an additional evaluation approach. This method measures the decrease in model performance when a feature is shuffled, thus identifying its level of importance. Figure 5 shows the results of the Permutation Feature Importance analysis.

Based on Figure 5, it can be seen that Soil Surface Moisture has the highest model score reduction value, approaching 1.0, indicating that this feature is the most dominant factor in determining fire risk. Furthermore, Rainfall shows a quite significant contribution with an importance value of around 0.10–0.12, followed by Average Temperature and Average Rela-

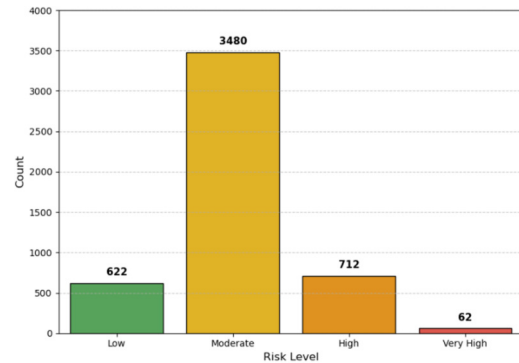


**Fig.5:** Permutation Feature Importance (Box Plot).

tive Humidity which have a relatively small influence. Meanwhile, Average Wind Speed shows the lowest importance value and is close to zero, indicating its contribution to the model is very minimal. These results are consistent with previous analyses showing that moisture-based variables have the strongest influence on increasing fire risk.

### 3.2 Class distribution

The class distribution in the dataset was analyzed to understand the data balance before the modeling process. This analysis is important because class imbalance can affect the model's performance in classifying accurately. Figure 6 shows the class distribution in the dataset used in this study.



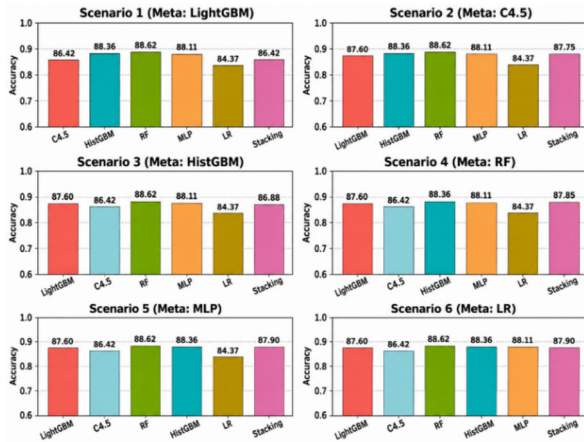
**Fig.6:** Class Distribution Graph.

Figure 6 shows a highly imbalanced class distribution. The Moderate class dominates the dataset with 3,480 observations, followed by the High class with 712 observations and the Low class with 622 observations, whereas the Very High class contains only 62 observations. This imbalance indicates that the model has the potential to be biased towards the majority class and has difficulty in recognizing patterns in minority classes, especially the Very High class which has a very limited amount of data. This condition is a major challenge in the classification process because it can reduce the model's sensitivity to important classes with high risk. Therefore, this study addresses this problem by implementing the proposed modeling approach, which is designed

to improve the model's ability to handle imbalanced data and produce more accurate and stable predictions across classes.

### 3.3 Performance of the proposed model

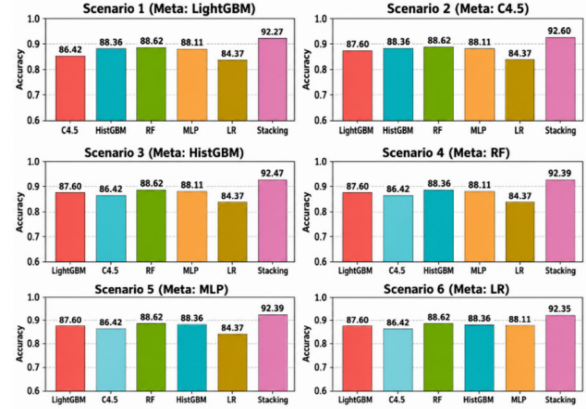
An initial evaluation was conducted on the baseline stacking model before applying development mechanisms such as rotation and weighting. This stage aimed to determine the baseline performance of the model combination used and to compare it with the proposed model. The following graph shows the results of the baseline stacking performance evaluation across six meta-model rotation scenarios.



**Fig.7:** Basic Stacking Accuracy Comparison Chart.

Figure 7 shows that the baseline stacking model achieved relatively low performance, with accuracy ranging from 86.42% to 87.90% across the six scenarios. In Scenario 1 (Meta: LightGBM), the stacking accuracy was recorded at 86.42%, lower than several base models such as Random Forest (88.62%) and MLP (88.11%). A similar pattern is also seen in other scenarios, where stacking is unable to surpass the performance of the best model individually. For example, in Scenario 3 (Meta: HistGB) and Scenario 4 (Meta: RF), the Random Forest and HistGB models still show the highest accuracy of around 88.62%, while the stacking results are below. Even in Scenarios 5 and 6, although stacking reaches around 87.90%, this value is still not significant compared to the best model. This indicates that without an optimization mechanism, the aggregation process in stacking is not able to maximize the combination of predictions from the base models.

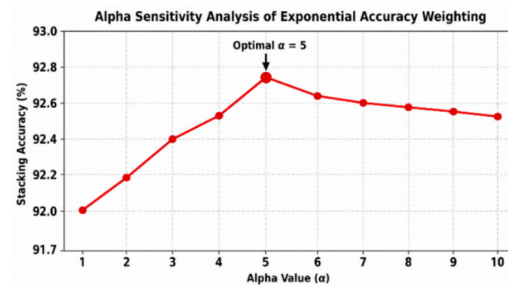
To address the limitations of base stacking, an accuracy weighting mechanism was implemented, which aims to optimize the contribution of each base model based on its performance. This approach is expected to improve model accuracy and stability through adaptive weighting. Figure 8 shows a comparison of model accuracy after applying accuracy weighting across six rotation scenarios.



**Fig.8:** Accuracy Comparison Chart Using Accuracy Weighting Scheme.

Based on Figure 8, it can be seen that the application of accuracy weighting significantly improves stacking performance across all scenarios, with accuracy ranging from 92.00% to 92.47%, an increase of approximately 5%–6% compared to the baseline stacking which was previously in the range of 86%–87%. In Scenario 3 (Meta: HistGB), the highest accuracy was obtained at 92.47%, followed by Scenario 6 (Meta: LR) at 92.35% and Scenario 4 (Meta: RF) at 92.39%. Meanwhile, other scenarios also show consistent improvements, such as Scenario 1 (92.27%) and Scenario 2 (92.00%). Interestingly, the stacking results across all scenarios are now able to surpass the performance of the best individual model which was previously in the range of 88.62%, indicating that the weighting mechanism successfully optimizes the contribution of models with high accuracy. This improvement also demonstrates that the exponential weighting approach can strengthen the dominance of more accurate models while reducing the influence of less optimal models, resulting in more stable and consistent predictions.

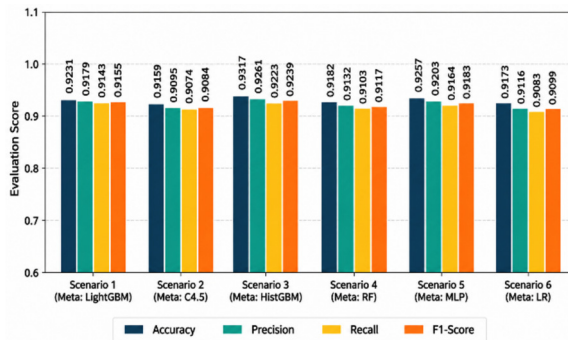
To evaluate the influence of the  $\alpha$  parameter in the exponential accuracy weighting scheme, a sensitivity analysis was performed to variations in the  $\alpha$  value. Figure 9 shows the results of the sensitivity analysis of the  $\alpha$  parameter on model performance.



**Fig.9:** Alpha Sensitivity Analysis in Exponential Accuracy Weighting.

Figure 9 shows that increasing  $\alpha$  from 1 to 5 significantly improves stacking accuracy, from approximately 92.00% to 92.47% at  $\alpha = 5$ . After reaching the optimal point, further increases in  $\alpha$  do not significantly improve performance, and instead tend to gradually decrease to approximately 92.38% at  $\alpha = 10$ . This pattern indicates that too small an  $\alpha$  value is unable to significantly differentiate the contributions between models, while too large an  $\alpha$  value leads to excessive dominance of certain models, thus reducing the balance of contributions. Therefore,  $\alpha = 5$  is considered the optimal parameter, providing the best balance between exploiting high-accuracy models and maintaining predictive diversity in the stacking process.

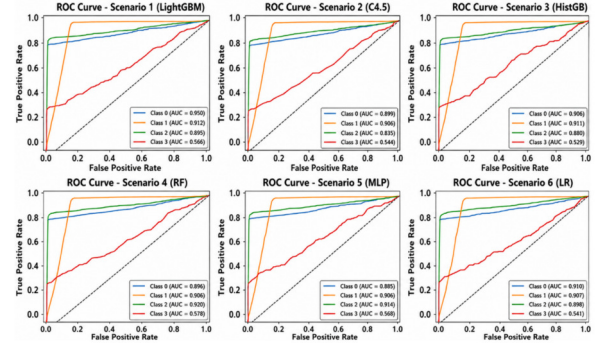
To evaluate the model classification performance in more detail, a confusion matrix and classification report were used for each rotation scenario after applying accuracy weighting. Overall, these results demonstrate that the model is capable of maintaining stable and accurate performance under imbalanced data conditions. The following is a comparison of the meta model performance based on accuracy, precision, recall, and F1-score:



**Fig.10:** Meta Model Accuracy Comparison Chart After Accuracy Weighting Application.

Figure 10 shows that all scenarios exhibit relatively stable and consistent performance, with evaluation values ranging from 0.92 to 0.94. In Scenario 1 (Meta: LightGBM), accuracy, precision, recall, and F1-score values hovered around 0.927, indicating a good balance between the metrics. Scenario 3 (Meta: HistGB) and Scenario 4 (Meta: RF) demonstrated the highest performance, with values approaching 0.939–0.940, indicating the model's ability to classify more accurately and consistently. Meanwhile, other scenarios, such as Scenarios 2, 5, and 6, also exhibited stable performance in the range of 0.928–0.935, with no significant differences between the metrics. The consistent accuracy, precision, recall, and F1-score values indicate that the model is not biased toward any particular class and is well balanced in handling all fire risk categories. This confirms that the application of accuracy weighting not only improves accuracy but also maintains the stability of the model's

overall performance. To evaluate the model's ability to distinguish between risk classes, Receiver Operating Characteristic (ROC) curve analysis was used for each scenario. The following figure 11 shows ROC curves for the six model scenarios tested.



**Fig.11:** ROC Graph of Accuracy Weighting Application.

Figure 11 shows that classification performance varied across scenarios and risk classes. The Low class (Class 0) achieved AUC values ranging from 0.885 to 0.950, indicating good to excellent discrimination. The Moderate class (Class 1) consistently achieved high AUC values of 0.906 to 0.912, demonstrating strong discrimination. The High class (Class 2) showed AUC values between 0.835 and 0.920, indicating generally good performance. In contrast, the Very High class (Class 3) obtained the lowest AUC values, ranging from 0.529 to 0.578. This lower performance is likely related to the limited number of observations, with only 62 datasets available for this class, restricting the model's ability to learn representative patterns. Overall, the proposed approach maintained strong discrimination for the Low, Moderate, and High classes despite substantial class imbalance. The rotation and accuracy based weighting strategy also helped maintain relatively consistent performance across scenarios.

To ensure the stability and generalizability of the model, an evaluation was conducted using the K-Fold Cross Validation method with K=10 for each scenario. Table 1 shows the accuracy results for each fold for each scenario.

The evaluation results in Table 1 show that all scenarios have consistent performance with an average accuracy ranging from 92.00% to 92.47%. Scenario 3 (HistGB) obtained the highest average value of 92.47%, followed by Scenario 4 and Scenario 5, which each reached 92.39%, and Scenario 6 at 92.35%. Accuracy variation between folds is relatively small, with the lowest value around 90.55% (Scenario 2, Fold 7) and the highest reaching 94.25%, which appears consistently across several scenarios in Fold 9. This stability indicates that the model has good generalization capabilities across various data subsets. Furthermore, the absence of extreme fluctuations be-

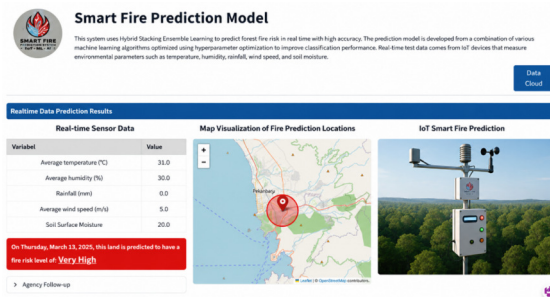
**Table 1:** Accuracy Table on Each Fold for Each Scenario.

| Scenario              | Fold 1 | Fold 2 | Fold 3 | Fold 4 | Fold 5 | Fold 6 | Fold 7 | Fold 8 | Fold 9 | Fold 10 | Average (%) |
|-----------------------|--------|--------|--------|--------|--------|--------|--------|--------|--------|---------|-------------|
| Scenario 1 (LightGBM) | 91.6   | 91.39  | 92.21  | 92.42  | 91.60  | 93.44  | 91.58  | 93.02  | 93.84  | 91.58   | 92.27       |
| Scenario 2 (C4.5)     | 91.39  | 90.98  | 92.21  | 91.80  | 91.19  | 93.24  | 90.55  | 92.61  | 94.05  | 91.99   | 92.00       |
| Scenario 3 (HistGB)   | 91.80  | 92.01  | 92.62  | 92.21  | 91.39  | 93.44  | 91.79  | 93.63  | 94.25  | 91.58   | 92.47       |
| Scenario 4 (RF)       | 91.60  | 91.80  | 92.42  | 92.42  | 91.39  | 93.24  | 91.38  | 93.22  | 94.25  | 92.2    | 92.39       |
| Scenario 5 (MLP)      | 91.19  | 91.80  | 92.21  | 92.42  | 91.60  | 93.24  | 91.79  | 93.43  | 94.25  | 91.99   | 92.39       |
| Scenario 6 (LR)       | 91.39  | 91.80  | 92.21  | 92.21  | 91.60  | 92.83  | 91.79  | 93.43  | 94.25  | 91.99   | 92.35       |

tween folds indicates that the model does not experience significant overfitting. The consistent performance across scenarios also indicates that the rotation and accuracy weighting approaches are capable of producing robust models, even when tested on different data subsets.

### 3.4 Real-time iot device testing results

System testing was conducted to evaluate model performance in real-time using data obtained directly from IoT devices. The system implementation is presented in a Streamlit-based interactive interface that allows for live visualization of sensor data, location, and prediction results. Figure 12 shows the results of real-time testing of the IoT-based prediction system using Streamlit.

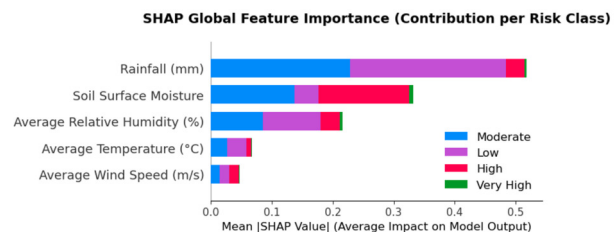
**Fig.12:** Real-time View of Model Testing Application with IoT Devices.

Test results show figure 12 that IoT devices are capable of transmitting meteorological data such as Average Temperature, Average Relative Humidity, Rainfall, Average Wind Speed, and Soil Surface Moisture in real-time to the system. The data is then directly processed by the Adaptive Hybrid Stacking Ensemble model to generate automatic fire risk level predictions. On the system display, sensor parameter values are displayed dynamically, accompanied by location map visualization using map visualization to show prediction points. In addition, the system also

provides direct classification results, for example the Very High category, based on detected environmental conditions. The integration between IoT devices and this prediction model shows that the system is capable of working responsively with low latency, so it can be used as a decision-support tool in real-time fire risk monitoring and mitigation.

### 3.5 Explainable ai

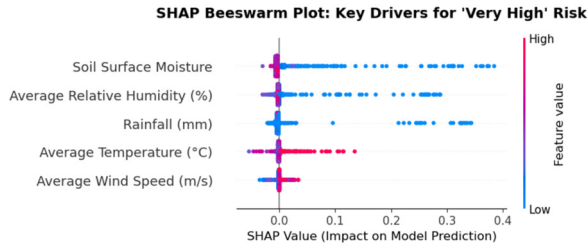
To improve model transparency and interpretability, an analysis was conducted using SHAP-based Explainable Artificial Intelligence. This approach aims to identify the contribution of each feature to the prediction results for each risk class. The following figure 13 shows results of the SHAP Global Feature Importance analysis for the developed model.

**Fig.13:** SHAP Global Feature Importance.

The visualization results in Figure 13 show that Rainfall is the feature with the highest contribution to the model, with a mean |SHAP value| of approximately 0.52, followed by Soil Surface Moisture with a value of approximately 0.30–0.32. This indicates that these two variables have a dominant influence in determining the level of fire risk. Furthermore, Average Relative Humidity provides a moderate contribution with a value of approximately 0.20–0.22, while Average Temperature and Average Wind Speed show relatively small influences, each below 0.10. The distribution of contributions within each class also shows that the High and Very High classes are more sensitive to changes in Rainfall and Soil Surface Moisture

values, as evidenced by the dominant colors in these two features.

To analyze the contribution of features more specifically to the highest risk class, a SHAP visualization in the form of a beeswarm plot was used. This approach allows for more detailed identification of the distribution of each feature's influence on the prediction of a particular class. Figure 14 shows the results of the SHAP Beeswarm Plot analysis for the Very High Risk class sample.



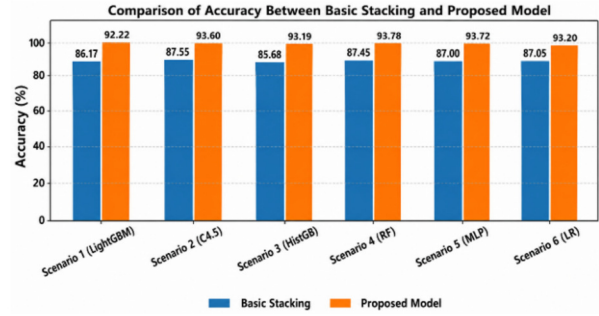
**Fig.14:** SHAP Beeswarm Plot.

The visualization results in Figure 14 show that Soil Surface Moisture has the most dominant influence on the prediction of the Very High class, with SHAP values spreading to around 0.35–0.40, especially at low feature values (marked in blue). This indicates that the lower the soil moisture, the greater its contribution in increasing the risk of fire. Average Relative Humidity and Rainfall also show significant contributions with SHAP values ranging up to around 0.25–0.30, confirming that dry environmental conditions play a major role in increasing the risk. Conversely, Average Temperature shows a positive contribution at high values (in red) with SHAP around 0.15–0.20, while Average Wind Speed has the smallest influence with values below 0.10. This distribution shows that the model consistently identifies moisture as the main determinant of extreme risk conditions, thus strengthening the model's validity in the context of the forest fire domain.

### 3.6 Discussion

The effectiveness of the proposed model is evaluated through a direct comparison between the basic stacking method and a model developed with rotation and accuracy weighting mechanisms. This analysis aims to demonstrate the performance improvements achieved by the proposed approach. Figure 15 shows an accuracy comparison between the basic stacking and the proposed model.

The comparison results in Figure 15 show that the proposed model consistently outperforms the baseline stacking model across all scenarios with significant accuracy improvements. In Scenario 1, accuracy increases from 86.42% to 92.27%, a 5.85% increase, while in Scenario 2, it increases from 87.75% to 92.00% (+4.25%). The highest improvement occurs in Scenario 3, from 86.88% to 92.47% (+5.59%),



**Fig.15:** Comparison Chart of Baseline Stacking Accuracy with the Proposed Model.

which is also the best overall accuracy. Other scenarios show a similar pattern, such as Scenario 4 (87.85% → 92.39%), Scenario 5 (87.90% → 92.39%), and Scenario 6 (87.90% → 92.35%), with an average improvement of around 4.5% to 5.8%. These results demonstrate that the integration of rotation mechanisms and accuracy-based exponential weighting can adaptively optimize the contribution of each model, resulting in more stable and superior performance compared to conventional stacking approaches.

To strengthen the research results, a comparison of the performance of the previous stacking model with the proposed model, which uses an accuracy weighting algorithm to optimize the contribution of each base model, was conducted. The following table 2 compares the results of the stacking model with the proposed model.

The comparison results in Table 2 show that the proposed model consistently provides significant performance improvements across all the compared studies. In [53], accuracy increased from 88.76% to 90.21% (+1.45%), while in [54] it increased from 90.12% to 91.34% (+1.22%). The most significant improvement was seen in [33], where accuracy increased from 85.43% to 89.67% (+4.24%), accompanied by an increase in F1-score from 84.75% to 89.10%. In [32], accuracy also increased from 88.22% to 90.88% (+2.66%). In addition, all evaluation metrics such as precision, recall, and F1-score showed consistent improvements with an average increase in the range of 1.2% to 4.2%. This demonstrates that applying accuracy weighting to the proposed model can improve the effectiveness of the base model combination, resulting in more optimal and stable performance compared to conventional stacking methods.

To test the model's generalizability, performance evaluations were conducted on several datasets with varying characteristics and number of classes. Table 3 compares the accuracy of the proposed model across four different datasets.

The test results in Table 3 show that the proposed model is able to maintain high and stable performance on various types of datasets, both structured and unstructured. On the meteorological dataset, the

**Table 2:** Comparison of Stacking Model Results with Proposed Model.

| Researchers | Base Model       | Meta Model | Basic Stacking                                                                  | Proposed Model                                                              |
|-------------|------------------|------------|---------------------------------------------------------------------------------|-----------------------------------------------------------------------------|
| [36]        | SVM, KNN, NB, RF | LR         | Accuracy: 88.76 %<br>Precision: 88.10 %<br>Recall: 87.95 %<br>F1-Score: 87.82 % | Accuracy: 90.21%<br>Precision: 90.05%<br>Recall: 89.88%<br>F1-Score: 89.76% |
| [18]        | SVM, LR, MLP, RF | XGBoost    | Accuracy: 90.12%<br>Precision: 90.45 %<br>Recall: 89.88 %<br>F1-Score: 89.76 %  | Accuracy: 91.34%<br>Precision: 91.10%<br>Recall: 90.95%<br>F1-Score: 90.82% |
| [17]        | LR, NB, KNN      | NB         | Accuracy: 85.43%<br>Precision: 85.10%<br>Recall: 84.92%<br>F1-Score: 84.75%     | Accuracy: 89.67%<br>Precision: 89.45%<br>Recall: 89.21%<br>F1-Score: 89.10% |
| [16]        | DT, SVM, RF      | LR         | Accuracy: 88.22%<br>Precision: 87.95%<br>Recall: 87.60%<br>F1-Score: 87.41%     | Accuracy: 90.88%<br>Precision: 90.72%<br>Recall: 90.55%<br>F1-Score: 90.41% |

model achieved an accuracy of 92.47%, while on the Flood History Data dataset, it increased to 93.72%, which is the highest value. On unstructured datasets such as social media reviews and disaster-related news with a larger number of classes (9 classes), the model still achieved a high accuracy of 92.55%, demonstrating its adaptability to the complexity of multiclass classification. Meanwhile, on datasets with four other classes, the accuracy remained at around 91.24%, which is still considered high. This consistency indicates that the model is not only effective in one domain but also able to perform optimally on different variations in the number of classes, thus having strong and flexible generalization capabilities in various prediction scenarios. Future research should investigate the application of time-based validation strategies, such as chronological train-test splitting or rolling-window validation, to better reflect real-world forecasting conditions. This approach may provide a more realistic assessment of the proposed Adaptive Hybrid Stacking Ensemble framework when applied to operational fire risk prediction using temporal meteorological data.

#### 4. CONCLUSION

This study successfully developed an Adaptive Hybrid Stacking Ensemble model for fire risk prediction based on meteorological data and real-time IoT integration. Experimental results show that the proposed approach can significantly improve performance compared to conventional stacking methods. Model accuracy increased from 86%–87% in the basic stacking range to 92.00%–92.47%, with an average increase of around 5%–6% across all scenarios. The best scenario was obtained in Scenario 3 (Meta: HistGB) with an accuracy of 92.47%, supported by a precision value of 92.48%, a recall of 92.47%, and an F1-score of 91.90%. The application of an exponential function-based accuracy weighting scheme proved effective in optimizing the contribution of each base model, where the optimal parameter was obtained at a value of  $\alpha = 5$ , which was able to provide the best balance between exploiting the best model and maintaining prediction diversity. Furthermore, the scenario rotation mechanism allows for systematic exploration of model combinations, thereby improving model stability and generalization. This is evidenced by the K-fold cross-validation results, which showed a stable average accuracy above 92% and an AUC value of 0.98–1.00. In terms of scientific contribution, this study presents

**Table 3:** Comparison of Proposed Model Accuracy Results in Testing 4 Different Datasets.

| No | Dataset                                                             | Type         | Number of Class | Best Scenario         | Accuracy(%) |
|----|---------------------------------------------------------------------|--------------|-----------------|-----------------------|-------------|
| 1  | Meteorology                                                         | Structured   | 4               | Scenario 3 (HistGB)   | 92.47       |
| 2  | Flood History Data                                                  | Structured   | 4               | Scenario 5 (RF)       | 93.72       |
| 3  | Social media reviews and disaster-related news                      | Unstructured | 9               | Scenario 1 (LightGBM) | 92.55       |
| 4  | Social media reviews and news related to weather and fire disasters | Unstructured | 4               | Scenario 2 (C4.5)     | 91.24       |

novelty through the integration of a model rotation strategy, an exponential accuracy weighting scheme, and SHAP-based explainable AI to improve model accuracy and interpretability. The results indicate that rainfall and soil surface moisture are dominant factors in fire risk prediction. However, data imbalance remains a challenge. Future research is recommended to utilize a multi-layer stacking architectural approach with accuracy weighting to improve performance. Integration with real-time IoT systems also demonstrates potential for broad implementation in fire early warning systems.

## AUTHOR CONTRIBUTIONS

Conceptualization, Y.I. and S.D.; methodology, Y.I. and S.D.; software, Y.I.; validation, Y.I., S.D. and R.S.; formal analysis, Y.I. and S.D.; investigation, Y.I.; data curation, Y.I.; writing—original draft preparation, Y.I.; writing—review and editing, S.D. and R.S.; visualization, Y.I.; supervision, S.D. and R.S.; funding acquisition, Y.I. All authors have read and agreed to the published version of the manuscript.

## References

- [1] S. Y. Ardiyanto and S. Andrikasmi, "Prevention, control and law enforcement forest and land burning in Riau Province," in *Proc. 2nd Riau Annu. Meeting Law Social Sci. (RAMLAS)*, pp. 90–94, 2022.
- [2] D. Sekarjati, W. Putklang and N. Wattanakij, "Identifying Peatland Fire Detection and Spatiotemporal Distribution: Implications for Land Use and land Cover Changes in Riau Province, Indonesia," *2024 28th International Computer Science and Engineering Conference (ICSEC)*, Khon Kaen, Thailand, pp. 1–6, 2024.
- [3] A. Muhaimin, Edriyansyah, Wahyat, Y. Irawan, and R. Wahyuni, "Optimized IoT-Based Multimodal Fusion for Early Forest Fire Detection and Prediction," *ECTI-CIT Transactions*, vol. 19, no. 4, pp. 569–582, Sep. 2025.
- [4] A. P. Selvam and S. N. S. Al-Humairi, "Environmental impact evaluation using smart real-time weather monitoring systems: A systematic review," *Innovative Infrastructure Solutions*, vol. 10, no. 1, Art. no. 13, 2024.
- [5] D. R. Ramani, B. Ben Sujitha and S. Tangade, "Smart environmental monitoring systems," in *Environmental Monitoring Using Artificial Intelligence*, pp. 45–60, 2025.
- [6] A. Morchid, Z. Oughannou, R. El Alami, H. Qjidaa, M. O. Jamil and H. M. Khalid, "Integrated Internet of Things (IoT) solutions for early fire detection in smart agriculture," *Results in Engineering*, vol. 24, Art. no. 103392, 2024.
- [7] Y. Irawan *et al.*, "Realtime monitoring and analysis based on cloud computing Internet of Things (CC-IoT) technology in detecting forest and land fires in Riau Province," *ILKOM Jurnal Ilmiah*, vol. 15, no. 3, pp. 445–454, 2023.
- [8] T. L. Narayana *et al.*, "Advances in real time smart monitoring of environmental parameters using IoT and sensors," *Heliyon*, vol. 10, no. 7, Apr. 2024.
- [9] A. N. Mabdeh, A. Al-Fugara, L. Abualigah, K. Saleem and V. Snasel, "Enhanced forest fire susceptibility mapping by integrating feature selection genetic algorithm and bagging-based support vector machine with artificial neural networks," *Stochastic Environmental Research and Risk Assessment*, vol. 38, no. 12, pp. 5039–5058, 2024.
- [10] K. Yazici and A. Taskin, "A comparative Bayesian optimization-based machine learning and artificial neural networks approach for burned area prediction in forest fires: An application in Turkey," *Natural Hazards*, vol. 119, no. 3, pp. 1883–1912, 2023.
- [11] A. R. Uthappa, B. Das, A. Raizada, P. Kumar, P. Jha and P. V. V. Prasad, "Forest fire susceptibility mapping using multi-criteria decision making and machine learning models in the Western Ghats of India," *Journal of Environmental Management*, vol. 379, Art. no. 124777, 2025.
- [12] L. Song *et al.*, "Early detection of wheat powdery mildew: A multi-source in situ remote sensing approach enabled by stacked ensemble learning," *Artificial Intelligence in Agriculture*, vol. 16, no. 1, pp. 124–138, 2026.
- [13] Y. Li, S. Guo, P. Chen, L. Chen and J. Mou, "A stacking-based ensemble learning model for intelligent ship trajectory interpolation," *Reliability Engineering & System Safety*, vol. 265, Art. no. 111615, 2026.
- [14] Y. Irawan, S. Defit and R. Sovia, "Rotational stacking ensemble with accuracy-based weighting for real-time forest fire risk prediction," *International Journal of Robotics and Control Systems*, vol. 6, no. 3, May 2026.
- [15] Y. Irawan, S. Defit and R. Sovia, "Optimizing stacking ensemble learning to enhance meta-model performance in forest fire risk level detection," in *Proc. 2025 1st International Conference on Emerging Trends in Information Systems and Informatics (ICETISI)*, pp. 1–6, 2025.
- [16] S. Reza, R. Amin, R. Yasmin, W. Kulsum and S. Ruhi, "Improving diabetes disease patients classification using stacking ensemble method with PIMA and local healthcare data," *Heliyon*, vol. 10, no. 2, Art. no. e24536, 2024.
- [17] C. Li, B. Xu, Z. Chen, X. Huang, J. He and X. Xie, "A stacking model-based classification algorithm is used to predict social phobia," *Applied Sciences*, vol. 14, no. 1, Art. no. 433, 2024.
- [18] H. Fonda, Y. Irawan, R. Melyanti, R. Wahyuni

- and A. Muhaimin, “A comprehensive stacking ensemble approach for stress level classification in higher education,” *Journal of Applied Data Sciences*, vol. 5, no. 4, pp. 1701–1714, 2024.
- [19] M. Garouani, A. Barhrhouj and O. Teste, “XS-tacking: An effective and inherently explainable framework for stacked ensemble learning,” *Information Fusion*, vol. 124, Art. no. 103358, 2025.
- [20] M. Lu *et al.*, “A stacking ensemble model of various machine learning models for daily runoff forecasting,” *Water*, vol. 15, no. 7, Art. no. 1265, 2023.
- [21] A. Abdollahi and B. Pradhan, “Explainable artificial intelligence (XAI) for interpreting the contributing factors feed into the wildfire susceptibility prediction model,” *Science of The Total Environment*, vol. 879, Art. no. 163004, 2023.
- [22] A. A. Awan, A. Majid, R. Riaz, S. S. Rizvi and S. J. Kwon, “A Novel Deep Stacking-Based Ensemble Approach for Short-Term Traffic Speed Prediction,” in *IEEE Access*, vol. 12, pp. 15222–15235, 2024.
- [23] N. Pramanick, S. Srivastava, J. Mathew and M. Agarwal, “Enhanced IDS using BBA and SMOTE-ENN for imbalanced data for cybersecurity,” *SN Computer Science*, vol. 5, no. 7, Art. no. 875, 2024.
- [24] Y. Devis, Muhamadiyah, Yulanda, Y. Irawan and R. Wahyuni, “Optimization of machine learning models for risk prediction of DHF spread to support management strategies in urban areas,” *Journal of Applied Data Sciences*, vol. 6, no. 4, pp. 2407–2420, 2025.
- [25] A. K. Putri and H. Suparwito, “Uji algoritma stacking ensemble classifier pada kemampuan adaptasi mahasiswa baru dalam pembelajaran online,” *Konstelasi: Konvergensi Teknologi dan Sistem Informasi*, vol. 3, no. 1, pp. 1–12, 2023.
- [26] L. Stracqualursi, “A lightweight exponential-weighted ensemble for crop recommendation,” *Journal of Agricultural, Biological and Environmental Statistics*, pp. 1–17, 2025.
- [27] A. Juna *et al.*, “Water quality prediction using KNN imputer and multilayer perceptron,” *Water*, vol. 14, no. 17, Art. no. 2592, 2022.
- [28] I. Aruleba and Y. Sun, “Enhanced credit risk prediction using deep learning and SMOTE-ENN resampling,” *Machine Learning with Applications*, vol. 21, Art. no. 100692, 2025.
- [29] A. M. Salih *et al.*, “A perspective on explainable artificial intelligence methods: SHAP and LIME,” *Advanced Intelligent Systems*, vol. 7, no. 1, Art. no. 2400304, Jan. 2025.
- [30] A. AlMohimeed *et al.*, “Explainable Artificial Intelligence of Multi-Level Stacking Ensemble for Detection of Alzheimer’s Disease Based on Particle Swarm Optimization and the Sub-Scores of Cognitive Biomarkers,” in *IEEE Access*, vol. 11, pp. 123173–123193, 2023.
- [31] H. El-Sofany, S. A. El-Seoud, O. H. Karam, Y. M. Abd El-Latif and I. A. T. F. Taj-Eddin, “A proposed technique using machine learning for the prediction of diabetes disease through a mobile app,” *International Journal of Intelligent Systems*, vol. 2024, 2024.
- [32] T. G. P. Megouo and S. Pierre, “A Stacking Ensemble Machine Learning Model for Emergency Call Forecasting,” in *IEEE Access*, vol. 12, pp. 115820–115837, 2024.
- [33] A. Lubis, Y. Irawan, Junadhi and S. Defit, “Leveraging K-nearest neighbors with SMOTE and boosting techniques for data imbalance and accuracy improvement,” *Journal of Applied Data Sciences*, vol. 5, no. 4, pp. 1625–1638, 2024.
- [34] K. Hernandez and A. B. Hoskins, “Machine learning algorithms applied to wildfire data in California’s Central Valley,” *Trees, Forests and People*, vol. 15, Art. no. 100516, 2024.
- [35] A. B. Adeyemi and A. A. Komolafe, “Flood hazard zones prediction using machine-learning-based geospatial approach in lower Niger River basin, Nigeria,” *Natural Hazards Research*, vol. 5, no. 2, pp. 399–412, 2025.
- [36] M. A. Rahim, M. A. Hossain, M. N. Hossain, J. Shin and K. S. Yun, “Stacked ensemble-based type-2 diabetes prediction using machine learning techniques,” *Annals of Emerging Technologies in Computing*, vol. 7, no. 1, pp. 30–39, 2023.



**Yuda Irawan** is a lecturer in the Informatics Engineering undergraduate program (S1) at Universitas Hang Tuah Pekanbaru, Indonesia. He completed his Bachelor’s degree in Computer Science at STMIK Amik Riau and his Master’s degree in Informatics Engineering at Bina Nusantara University. He completed his doctoral degree (Ph.D.) in Information Technology at Universitas Putra Indonesia YPTK Padang. His research interests include the Internet of Things (IoT), Machine Learning, and Deep Learning.



**Sarjon Defit** is a Professor at Universitas Putra Indonesia YPTK Padang. He completed his B.Sc. at Universitas Putra Indonesia YPTK Padang and obtained his M.Sc. and Ph.D. from Universiti Teknologi Malaysia (UTM). His research centers on data science, including machine learning, predictive analytics, and data-driven decision support.



**Rini Sovia** is an Associate Professor at Universitas Putra Indonesia YPTK Padang. She completed her B.Sc., M.Sc., and Ph.D. at Universitas Putra Indonesia YPTK Padang, where her research centers on data science, including machine learning, predictive analytics, and data-driven decision support.