

HSEL: An adaptive rotational hybrid stacking ensemble with exponential accuracy weighting for real-time forest fire risk prediction

Yuda Irawan^{*1)}, Sarjon Defit²⁾ and Rini Sovia²⁾

¹⁾Faculty of Computer Science, Universitas Hang Tuah Pekanbaru, Pekanbaru 28288, Indonesia

²⁾Faculty of Computer Science, Universitas Putra Indonesia YPTK Padang, Padang 25147, Indonesia

Received 2 March 2026

Revised 9 July 2026

Accepted 13 July 2026

Abstract

This study proposes the HSEL model to improve the accuracy of fire risk predictions based on meteorological data and real-time IoT integration. The main issues with individual models are their susceptibility to bias and overfitting, while conventional stacking methods have not been able to effectively optimize the contribution of each base model. HSEL addresses these issues through the application of exponential accuracy weighting with parameter $\alpha = 5$ and hyperparameter optimization using Optuna. The results show that accuracy improved from 87.80%–88.26% in the base stacking to 92.55%–94.20% after applying weighting, and reached 96.53% after optimization. Additionally, the model demonstrates stable performance, is capable of handling data imbalance, and possesses good generalization ability. This model also successfully improves classification balance across classes and significantly reduces prediction errors in the minority class. SHAP-based Explainable AI (XAI) analysis shows that the model is capable of providing transparent interpretations of the main factors influencing predictions.

Keywords: HSEL, Stacking ensemble, Exponential weighting, Optuna, Fire prediction

1. Introduction

Forest fires are an environmental disaster that has a significant impact on ecosystems, public health, and economic stability, especially in tropical regions such as Indonesia [1, 2]. The increasing frequency of fires in recent decades demonstrates the urgency of developing accurate, real-time early warning systems. Meteorological data such as temperature, humidity, rainfall, and wind speed have been widely used as primary indicators in fire risk modeling, as they are directly related to the environmental conditions that trigger fires [3, 4]. Therefore, the use of machine learning technology in meteorological data analysis is becoming an increasingly important approach in supporting data-based disaster mitigation systems.

Various previous studies have developed fire risk prediction models using machine learning algorithms such as Random Forest, Support Vector Machine, and Gradient Boosting [3-5]. Although these models are capable of providing quite good performance, the single model approach still has limitations, especially in terms of generalization and robustness to data variations [6-8]. Individual models tend to be susceptible to bias and overfitting, especially when faced with imbalanced or highly complex datasets [9]. This condition causes the model performance to become unstable when tested on data that is different from the training data [10].

To overcome these limitations, ensemble learning approaches such as stacking are starting to be widely used because they are able to combine the strengths of various basic models [11-14]. Stacking allows the creation of meta models that learn from the predicted output of the base model, thus hopefully improving overall performance [15-18]. However, the conventional stacking approach still faces problems in determining the optimal contribution of each base model [19, 20]. The prediction merging process is generally carried out linearly or without considering differences in model quality, so it has the potential to produce combinations that are not optimal and less adaptive to performance variations between models [21].

On the other hand, the development of the Internet of Things (IoT) has opened up opportunities for real-time environmental data collection to support fire prediction systems [22-24]. However, most IoT implementations are still limited to data collection and visualization without strong integration with machine learning models for automated decision making [25, 26]. In addition, many IoT systems use non-industrial sensors such as the DHT22 which have limitations in measurement accuracy and stability, potentially reducing the quality of the data used in modeling [27]. These challenges highlight the need for tighter integration between IoT devices and robust predictive models, as well as the selection of more reliable sensors to support accurate prediction systems.

Based on these problems, this study developed a new model called Hybrid Stacking Ensemble Learning (HSEL) that integrates stacking techniques with exponential accuracy weighting mechanisms and hyperparameter optimization using Optuna. This model combines several base models such as LightGBM, C4.5, HistGradientBoosting, Random Forest, MLP, and Logistic Regression, and uses deep learning models as meta learners, namely GRU, BiLSTM, and Transformer Encoder. In addition, this study also integrates an IoT system based on environmental sensors to support real-time fire risk prediction. The main contribution of this research lies in

*Corresponding author.

Email address: yudairawan89@gmail.com

<https://doi.org/10.64960/easr.2026.266473>

the development of an exponential-based adaptive weighting mechanism in stacking, improving model performance through hyperparameter optimization, and integrating an IoT system that is capable of producing predictions directly.

2. Materials and methods

This study proposes an ensemble-based learning framework called HSEL model. HSEL is an enhanced approach to conventional stacking methods by integrating adaptive weighting mechanisms, meta-level feature engineering, and the use of deep learning models as meta-learners. This approach is designed to enhance the model's ability to capture complex relationships between base model predictions while improving generalization capabilities. Furthermore, HSEL also accommodates a hyperparameter optimization process using Optuna to obtain optimal model configurations. By combining these various components, HSEL is expected to produce more stable and accurate prediction performance. The following is Figure 1 of the HSEL model development proposed in this study.

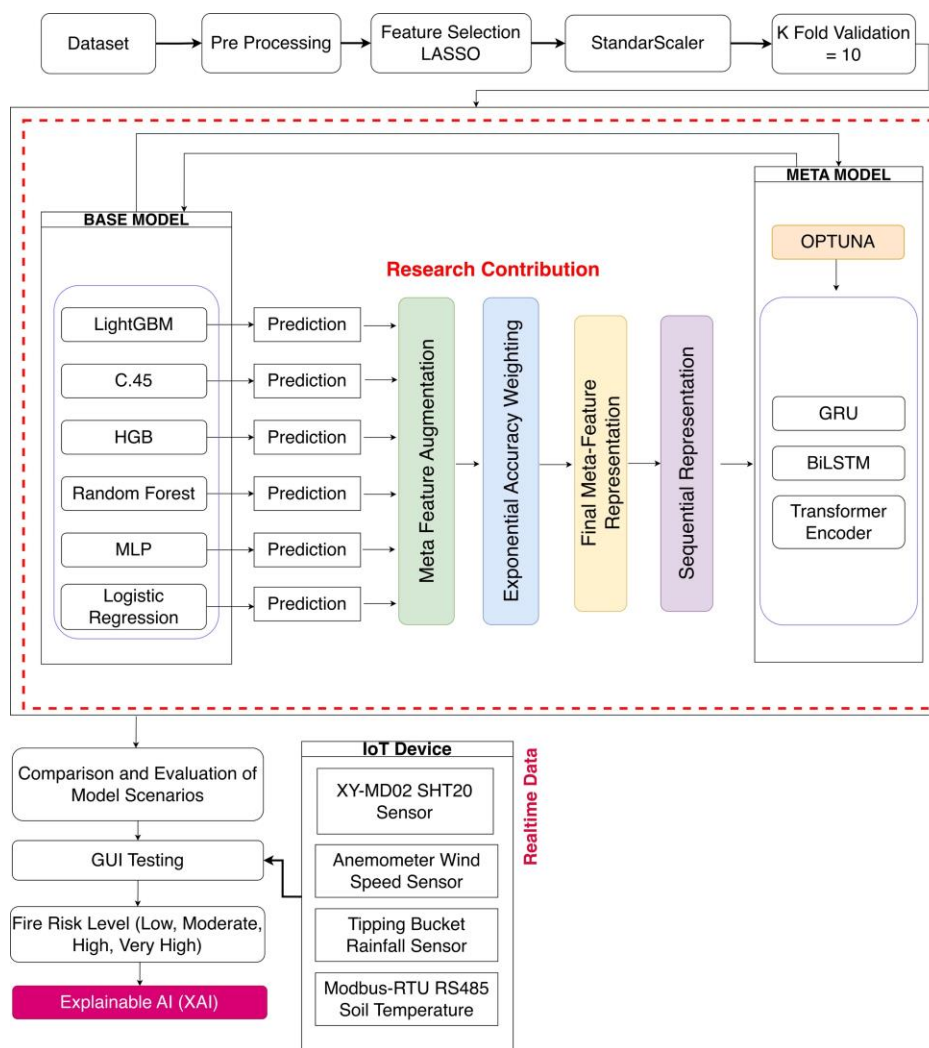


Figure 1 Proposed HSEL Model

Based on the HSEL model development image, the process begins with the data processing stage, which includes preprocessing, feature selection using the LASSO method, normalization with StandardScaler, and validation using K-Fold Cross Validation. Furthermore, several base models are used, including LightGBM, C4.5, HistGradientBoosting, Random Forest, MLP, and Logistic Regression, to generate initial predictions. The output of each base model is then processed through the meta-feature augmentation stage to enrich the feature representation at the meta-level. After that, weighting is carried out using the exponential accuracy weighting method to provide adaptive weights based on the performance of each model. The weighting results form the final meta-feature representation, which is then converted into a structured representation before being entered into the meta-model. In the meta-learning stage, several deep learning models are used, namely GRU, BiLSTM, and Transformer Encoder, which are optimized using Optuna to improve performance.

The selection of base learners was designed to maximize model diversity by combining algorithms with different learning characteristics. LightGBM and HistGradientBoosting were selected as efficient gradient boosting models capable of capturing complex nonlinear relationships. Random Forest was employed because of its robustness against noise and overfitting through bootstrap aggregation. C4.5 was included to provide interpretable rule based decision boundaries, while Logistic Regression represented a linear classifier capable of modeling global linear trends. MLP was incorporated to capture nonlinear feature interactions through neural network learning. This heterogeneous combination allows HSEL to generate complementary prediction patterns that enrich the meta level representation and improve ensemble generalization.

Three deep learning architectures, namely GRU, BiLSTM, and Transformer Encoder, were selected as meta learners because each offers different capabilities for modeling dependencies among meta features. GRU provides a computationally efficient recurrent architecture with fewer parameters, BiLSTM captures bidirectional contextual relationships between base model outputs, while the Transformer Encoder utilizes self attention mechanisms to learn complex global interactions without relying on sequential recurrence. Evaluating these three architectures enables a comprehensive comparison of recurrent and attention based meta learning strategies within the HSEL framework.

In addition, this system is also integrated with IoT devices consisting of temperature, humidity, wind speed, rainfall, and soil temperature sensors to generate real-time data. The final result of the model is a classification of fire risk levels which is then further analyzed using the Explainable AI (XAI) approach to improve the interpretability of the model.

2.1 Dataset and preprocessing

The dataset used in this study comes from the Indonesian Meteorology, Climatology, and Geophysics Agency (BMKG) with an observation period from 2011 to 2024, covering various meteorological parameters to support fire risk prediction analysis. Preprocessing was carried out to ensure optimal data quality through a series of steps, including data cleaning, feature selection using the LASSO method, normalization using StandardScaler, and feature redundancy management to improve model performance.

2.1.1 Feature selection

The feature selection stage in this study was conducted using the Least Absolute Shrinkage and Selection Operator (LASSO) method to identify the most relevant features for the prediction target. The initial dataset consisted of 11 variables, which were then selected based on the regression coefficient values generated by LASSO, where features with low contributions were automatically eliminated through the L1 regularization mechanism. This approach not only helps reduce data dimensionality but also improves computational efficiency and prevents overfitting by retaining features that have a significant influence on the model.

2.1.2 StandardScaler

This process rescales the features so that they have a mean of 0 and a standard deviation of 1. Normalization is necessary because the dataset includes variables with different scales, such as temperature (in degrees Celsius) and wind speed (in meters per second). Feature scaling ensures that each variable contributes equally to the model and helps scale-sensitive machine learning algorithms perform optimally.

2.2 Stacking ensemble learning

The stacking ensemble method is a layered learning approach that utilizes a combination of several basic models to improve the quality of predictions [11]. Suppose there are a number of basic models N denoted as $h_1, h_2, h_3, \dots, h_N$, where each model produces a prediction output based on input data x [12, 13]. The predictions generated by each base model are then collected and arranged into a new feature vector at the meta level as follows:

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

The z vector is then used as input for a second-level model that acts as a meta-learner, denoted by H . This model is responsible for learning the relationships between the base model outputs and producing more optimal final predictions. Mathematically, this process can be expressed as:

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

With this approach, stacking allows the integration of information from multiple base models into a single, more comprehensive learning system, thereby improving overall prediction performance.

2.3 Adaptive meta-feature learning and weighting mechanism

This section describes the main mechanisms within the HSEL framework, including the meta-feature generation process, adaptive weighting based on model performance, and the final representation as input to the meta-learner. This approach enriches the feature representation by not only combining the outputs of the base models but also considering the quality of each model in the learning process.

The initial stage is performed through meta-feature augmentation, where the output from each base model is compiled into a second-level feature representation. This representation is expressed as:

$$z = [h_1(x), h_2(x), \dots, h_N(x)] \quad (3)$$

Next, a weighting process is carried out using exponential accuracy weighting to provide different contributions to each model based on its predictive performance [28]. The weight of each model is calculated using the exponential function as follows:

$$w_x = \frac{e^{(\alpha)acc_x}}{\sum_{j=1}^{|M_r|} e^{\alpha \cdot acc_j}} \quad (4)$$

where w_x is the weight of the x th model, acc_x indicates the accuracy of the x -th model, α is the sensitivity parameter, $|M_r|$ is the number of models used, and e is an exponential constant.

For example, two models with accuracies of 0.90 and 0.85 differ by only 0.05 in absolute terms. However, when multiplied by $\alpha = 5$, their exponential values become $e^{4.5} \approx 90$ and $e^{4.25} \approx 70$. This difference results in a more contrasting weight distribution, so the model with the higher performance receives a more dominant contribution than the model with the lower accuracy.

After the weights are obtained, the process of forming weighted meta-features is carried out by multiplying each model output by its respective weight, which can be expressed as:

$$z' = [w_1 \cdot h_1(x), w_2 \cdot h_2(x), \dots, w_n \cdot h_n(x)] \quad (5)$$

This representation is then further processed to form a final, more informative meta-feature representation ready for use by the next-level model. To support the use of deep learning architectures such as GRU, BiLSTM, and Transformer Encoder, the representation is structured sequentially as follows:

$$Z_{seq} = [z'_1, z'_2, \dots, z'_t] \quad (6)$$

This approach allows the system to not only combine predictions from multiple models, but also to adaptively adjust their contributions based on performance, resulting in a richer and more effective feature representation to improve prediction accuracy.

2.4 Hyperparameter tuning

Hyperparameter optimization in meta-learners is performed using Optuna to automatically obtain the most effective model configuration. Optuna utilizes a Bayesian optimization-based search strategy (Tree-structured Parzen Estimator) that adaptively explores parameter space such as the number of units, number of layers, dropout, learning rate, and batch size in GRU, BiLSTM, and Transformer Encoder models [24, 29]. The optimization process is performed iteratively by evaluating model performance on each trial using the K-Fold Cross Validation scheme, ensuring that the selected parameters not only provide high performance but also have good generalization capabilities. Furthermore, a pruning mechanism is used to terminate unpromising trials early, thereby improving computational efficiency. This approach allows HSEL to achieve optimal performance without relying on manual parameter selection.

2.5 K-Fold validation

Model performance evaluation in this study was conducted using the K-Fold Cross Validation method with a K value of 10 to ensure more robust results and unbiased results based on specific data distributions. The dataset was divided into 10 equal subsets, where at each iteration, one subset was used as test data while the other nine subsets were used as training data. This process was repeated until all subsets served as test data, thus obtaining a more representative average model performance. This approach helps reduce the risk of overfitting and provides more stable estimates of model performance across various data scenarios.

2.6 IoT device design

To support the implementation of the HSEL model in a real environment, an IoT system is designed that is integrated with various environmental sensors and data processing devices, the following is Figure 2 of the proposed system design.



Figure 2 IoT Device Model and Design Elements

Figure 2 illustrates the IoT system architecture consisting of sensors, a processing unit, and a control system. Environmental data including temperature, humidity, rainfall, wind speed, and soil moisture are collected using multiple sensors and transmitted to a Raspberry Pi for processing. The data are then analyzed through a cloud-based IoT system to enable real-time prediction using the HSEL model. The results are visualized through a control interface, allowing users to monitor environmental conditions and fire risk predictions efficiently. This integration enables faster and more accurate decision-making.

2.7 Explainable AI (XAI)

Explainable AI (XAI) approach is applied to improve the interpretability of HSEL models through analyzing the contribution of features to the prediction results [30]. This study used the SHAP (SHapley Additive Explanations) method, which allows interpretation at both global and local levels. Global analysis was performed using SHAP Global Feature Importance to identify the features that most influence overall model performance, while local interpretation was obtained through SHAP Explainer, which explains the contribution of each feature to the prediction for each data instance. This approach provides a more transparent understanding of how

the model makes decisions, thereby increasing user confidence and supporting the validation of prediction results in the context of risk analysis.

3. Results

3.1 Dataset and preprocessing

The dataset used in this study was obtained from the Sultan Syarif Kasim II Meteorological Station, operated by the Indonesian Agency for Meteorology, Climatology, and Geophysics (BMKG), Pekanbaru, Riau Province, Indonesia. The dataset covers daily meteorological observations collected from 2011 to 2024 and includes multiple environmental variables relevant to forest fire risk prediction, including average temperature, relative humidity, rainfall, wind speed, and soil surface moisture. A total of 4,931 observations with 11 attributes were collected and subsequently preprocessed through data cleaning, feature selection using LASSO, normalization with StandardScaler, and feature redundancy management before model development.

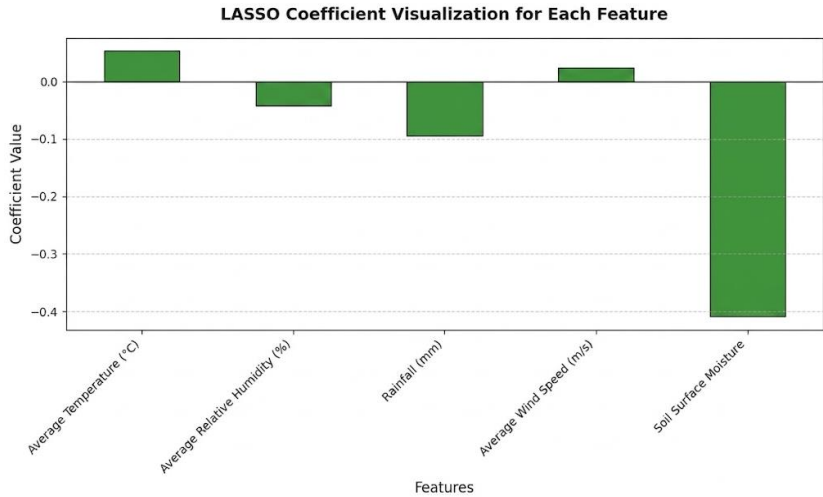


Figure 3 Lasso Coefficient Visualization Results

As shown in Figure 3, the LASSO coefficient visualization indicates that Soil Surface Moisture has the largest absolute coefficient value, approximately -0.41 , suggesting the strongest contribution among the selected features. Rainfall and Average Relative Humidity have negative coefficients of approximately -0.10 and -0.05 , respectively, indicating inverse relationships with the predicted outcome. Conversely, Average Temperature and Average Wind Speed exhibit positive coefficients of approximately 0.06 and 0.03 , respectively. Overall, the coefficient values range from approximately -0.41 to 0.06 , with Soil Surface Moisture exhibiting the largest absolute LASSO coefficient, while the remaining meteorological variables provide comparatively smaller contributions to the model.

To understand the relationship between meteorological variables and fire risk levels, an exploratory analysis using scatter plots was conducted on several key features. This visualization aims to identify data distribution patterns and trends in the relationship between each variable and the risk label. Figure 4 shows the results of the analysis of the relationships between the variables.

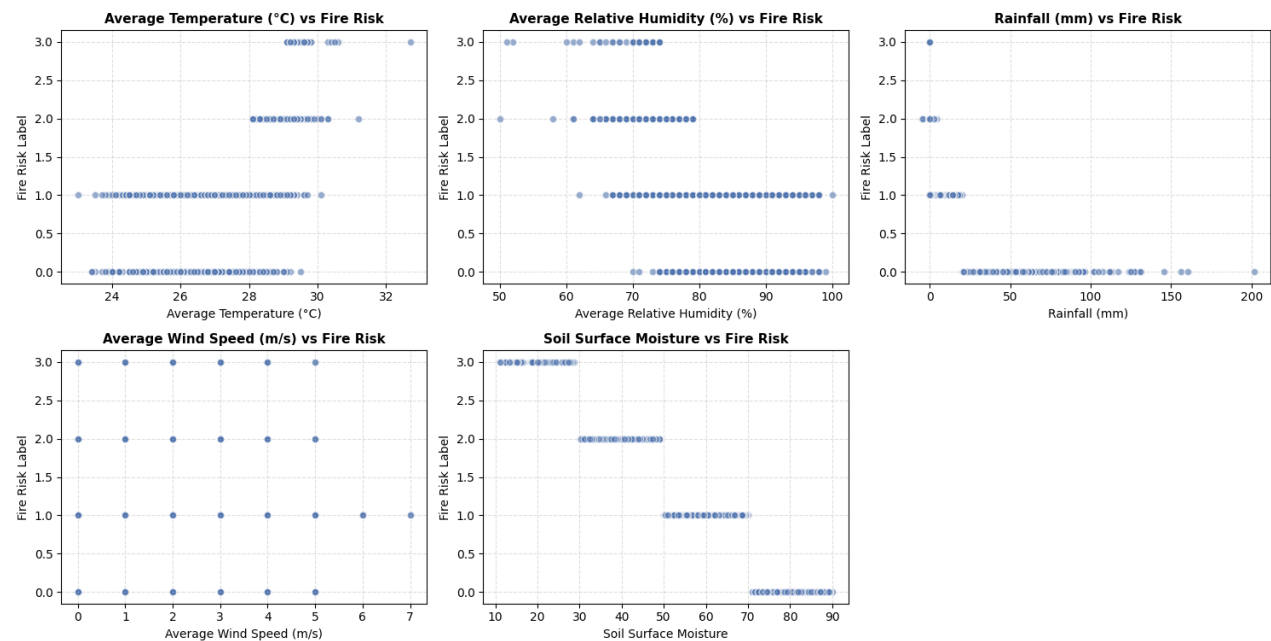


Figure 4 Scatter Plot of the Relationship between Selected Features and Fire Risk Labels

Based on the scatter plot results, Average Temperature ranges from approximately 23°C to 33°C, with higher fire risk observed above 28°C. Average Relative Humidity spans from 50% to 100%, where lower humidity levels (50%–70%) are more frequently associated with higher fire risk. Rainfall ranges widely from 0 to 200 mm, with high-risk events mostly occurring near 0 mm, indicating a strong inverse relationship. Average Wind Speed varies between 0 and 7 m/s, showing a moderate increase in risk above 3 m/s. Soil Surface Moisture exhibits the most significant pattern, ranging from 10 to 90, where lower values (below 40) consistently correspond to higher fire risk. Overall, Soil Surface Moisture and Rainfall show the strongest relationship with fire risk, while other variables contribute more moderately. To further analyze relationships among variables, a Pearson correlation heatmap is used, as shown in Figure 5.

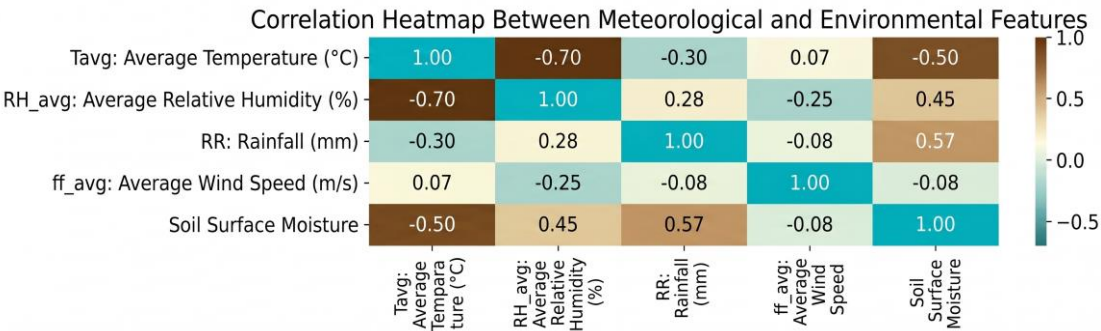


Figure 5 Heatmap of Correlation Between Meteorological Features

Based on Figure 5, the correlation heatmap shows the strongest relationship between Average Temperature and Average Relative Humidity with a value of -0.70, indicating a strong negative correlation. Soil Surface Moisture has a positive correlation with Rainfall (0.57) and Average Relative Humidity (0.45), as well as a negative correlation with Average Temperature (-0.50), indicating the important role of these variables in environmental conditions. Rainfall also has a moderate negative correlation with temperature (-0.30) and a weak positive correlation with humidity (0.28). Meanwhile, Average Wind Speed shows a low correlation with other variables (-0.25 to 0.07), so it has a relatively small influence. In general, temperature, air humidity, rainfall, and soil moisture are dominant factors in shaping fire risk patterns.

3.2 Class distribution

To understand the characteristics of the data distribution for the target variable, an analysis of the proportions of each fire risk class was conducted. This analysis is crucial for identifying potential data imbalances that could impact model performance. Figure 6 shows the results of the class distribution analysis.

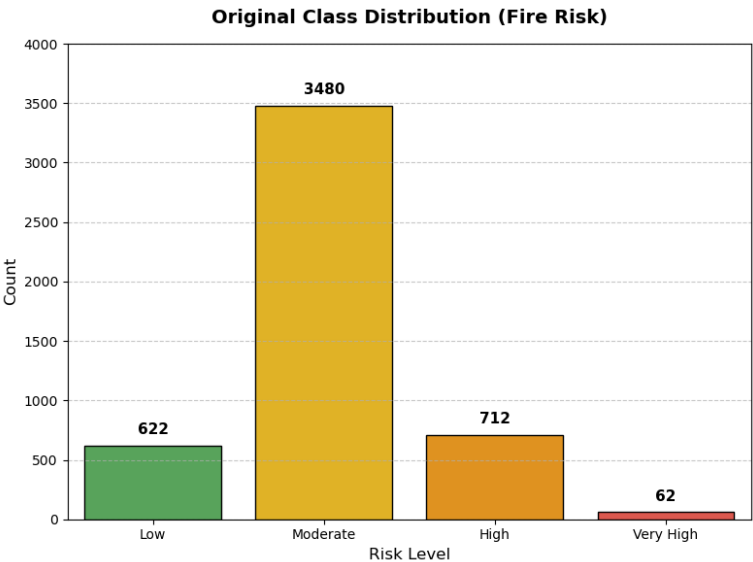


Figure 6 Risk Class Distribution Graph

Based on the class distribution graph, it can be seen that the Moderate class dominates the dataset with 3,480 data points (approximately 70.6%), followed by the High class with 712 data points (14.4%), the Low class with 622 data points (12.6%), and the Very High class with only 62 data points (1.3%). This distribution indicates significant data imbalance, especially in the Very High class, which has a very small proportion compared to other classes. This imbalance poses a challenge in the model learning process, as it has the potential to cause bias towards the majority class and reduce the model's ability to recognize patterns in the minority class. In developing the HSEL model, this condition can affect the weighting and meta-feature learning processes, so strategies are needed that can maintain a balance of contributions between classes and prevent overfitting towards the dominant class. Therefore, approaches such as adaptive weighting and cross-validation are important to ensure the model remains able to generalize well across all risk classes.

3.3 HSEL Performance

To evaluate the initial performance of the stacking method, tests were conducted using several base models, namely LightGBM, C4.5, HistGradientBoosting (HGB), Random Forest, MLP, and Logistic Regression with three different meta model scenarios, namely GRU, BiLSTM, and Transformer Encoder. This test aims to see the baseline performance before applying the HSEL approach, the following is Figure 7, the results of the stacking model performance analysis.

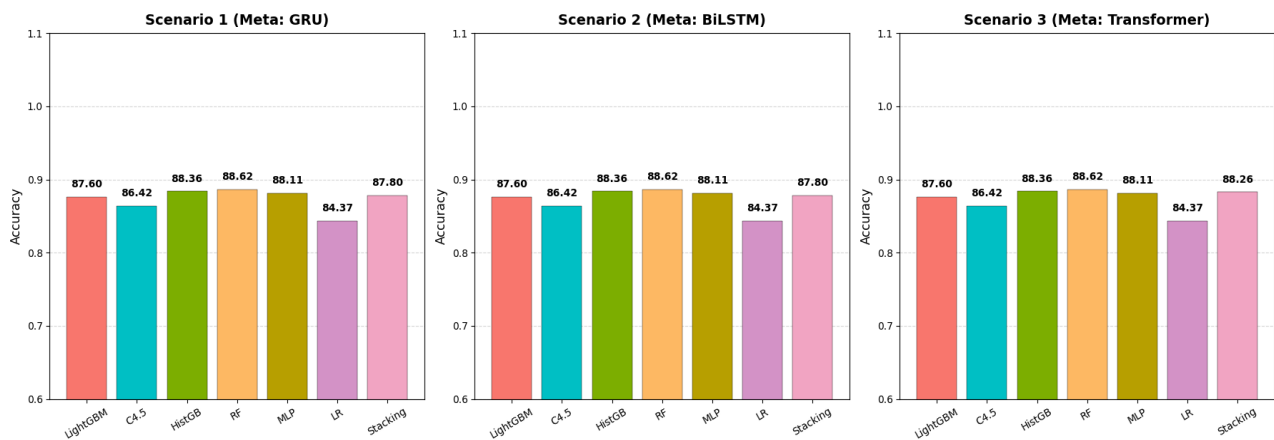


Figure 7 Stacking Accuracy Comparison Chart

Based on the test results in Figure 7, it can be seen that the model performance is relatively consistent across the three meta-model scenarios, but does not show a significant improvement compared to the individual models in the base model. In the GRU and BiLSTM meta-scenarios, the Random Forest and HistGradientBoosting models remain the best with accuracies of 88.62% and 88.36%, respectively, followed by MLP at 88.11%, while LightGBM and C4.5 are in the range of 87.60% and 86.42%. However, the results of stacking as a meta-model only achieve an accuracy of around 87.80%, which is lower than some individual models such as Random Forest and HistGradientBoosting. The same pattern is also seen in the Transformer scenario, where the stacking accuracy is around 88.26%, but still does not consistently exceed the performance of the best model in the base model. This indicates that the process of combining predictions in stacking has not been able to optimize the contribution of each model effectively. This condition indicates that there are limitations in the stacking approach, especially in capturing quality differences between models, so a more adaptive mechanism such as HSEL is needed to improve performance more optimally.

To increase the contribution of the best-performing model, an exponential accuracy weighting method with a sensitivity parameter of $\alpha = 5$ was applied, which aims to increase the weight difference between models based on their accuracy values. This approach is expected to optimize the stacking process by providing a greater influence on the more accurate model. The following is Figure 8, the results of the analysis of the accuracy weighting application.

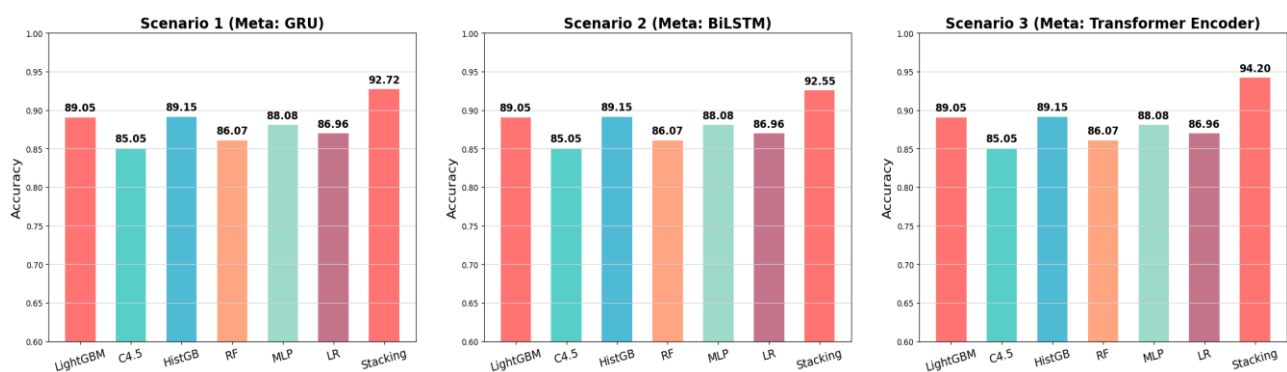


Figure 8 Accuracy Comparison Chart Using Accuracy Weighting Scheme

Based on the test results in Figure 8, the application of exponential accuracy weighting shows a significant performance improvement compared to the previous stacking. In the meta GRU scenario, the stacking accuracy increased to 92.72%, higher than the previous best individual model, HistGradientBoosting, at 89.15%, with an improvement of about 3.57%. In the BiLSTM scenario, the accuracy reached 92.55%, an increase of about 3.40% compared to the best model. Meanwhile, in the Transformer Encoder scenario, the highest accuracy reached 94.20%, which showed an improvement of 5.05% compared to the best baseline. Overall, the application of weighting resulted in an accuracy increase in the range of 92.55% to 94.20%, which consistently outperformed all individual models. This shows that the exponential-based weighting approach with $\alpha = 5$ is able to strengthen the contribution of high-performance models and reduce the influence of low-accuracy models, resulting in a more significant and stable performance improvement.

To evaluate the model performance more comprehensively, a classification report analysis was carried out which included accuracy, precision, recall, and F1-score metrics in each test scenario. The following is Figure 9 showing the results of the model classification performance analysis.

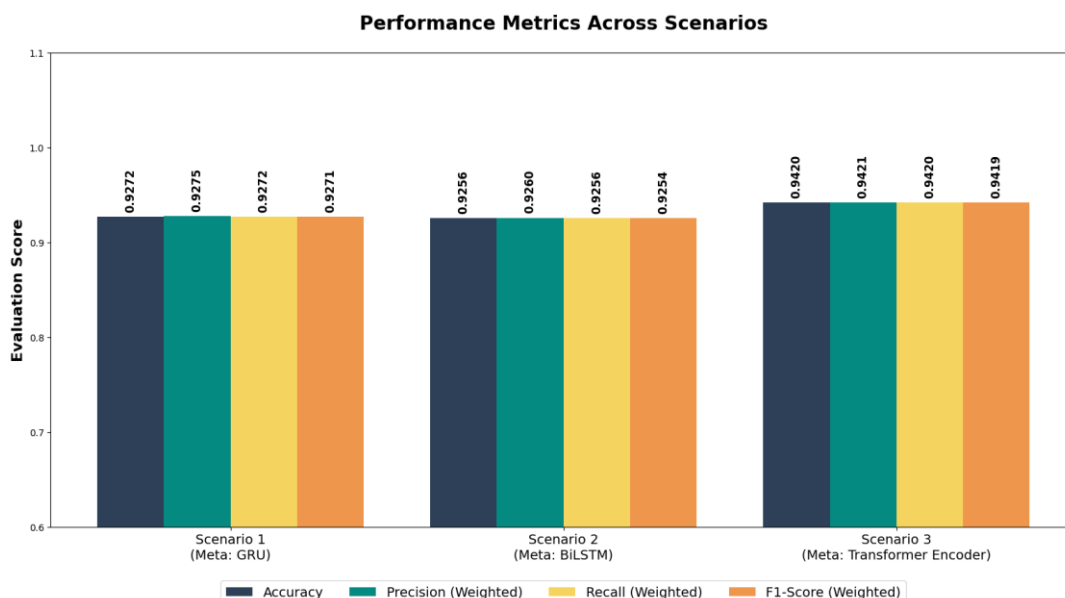


Figure 9 Comparison of Classification Report Model Stacking with Accuracy Weighting

Based on the classification report results, it can be seen that all evaluation metrics show high and consistent performance across all three scenarios. In the GRU scenario, an accuracy value of 0.9272 was obtained with a precision of 0.9275, a recall of 0.9272, and an F1-score of 0.9271. The BiLSTM scenario showed comparable performance with an accuracy of 0.9256, a precision of 0.9260, a recall of 0.9256, and an F1-score of 0.9254. Meanwhile, the Transformer Encoder scenario provided the best results with an accuracy of 0.9420, a precision of 0.9421, a recall of 0.9420, and an F1-score of 0.9419. The very small difference between the metrics (less than 0.002) indicates that the model has a good balance between precision and recall, thus minimizing classification errors in both false positives and false negatives.

To improve the performance of the meta-model, a hyperparameter tuning process using Optuna was performed on each of the GRU, BiLSTM, and Transformer Encoder architectures. This optimization aims to find the best parameter configuration to improve model accuracy and stability. The following Figure 10 shows the results of the analysis of the application of Optuna to the meta-model.

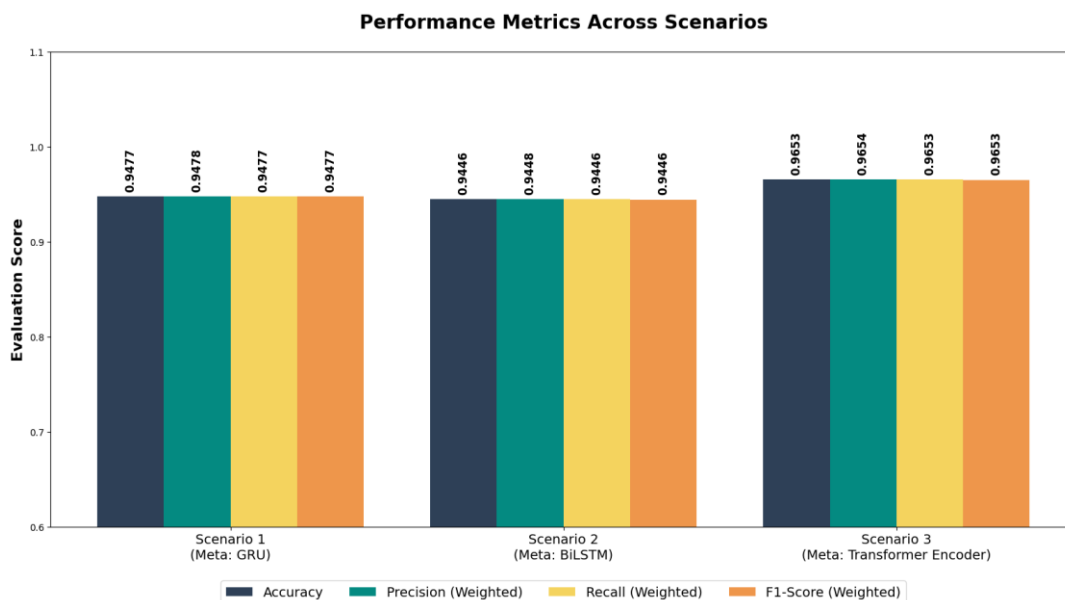


Figure 10 Comparison Chart of Classification Report Implementation of OPTUNA on Meta Model

Based on the test results, the implementation of Optuna showed consistent performance improvements across all meta-model scenarios. In the GRU scenario, an accuracy of 0.9477 was obtained with identical precision, recall, and F1-score at 0.9477. The BiLSTM scenario showed slightly lower results with an accuracy of 0.9446, precision 0.9448, recall 0.9446, and F1-score 0.9446. Meanwhile, the Transformer Encoder produced the best performance with an accuracy of 0.9653, precision 0.9654, recall 0.9653, and F1-score 0.9653. Compared to before tuning, there was an increase in accuracy of around 2.5% to 3% in each scenario. Furthermore, the very small difference between metrics (less than 0.001) indicates that the model has excellent balance in classification. These results indicate that hyperparameter optimization using Optuna is able to improve the model's generalization capability and maximize the meta-model performance within the HSEL framework.

To evaluate classification performance more thoroughly, a confusion matrix and classification report analysis were performed on each meta-model scenario after applying accuracy weighting and Optuna optimization. This evaluation aimed to measure the model's ability to classify each specific class. Figure 11 shows the results of the HSEL model classification performance analysis.

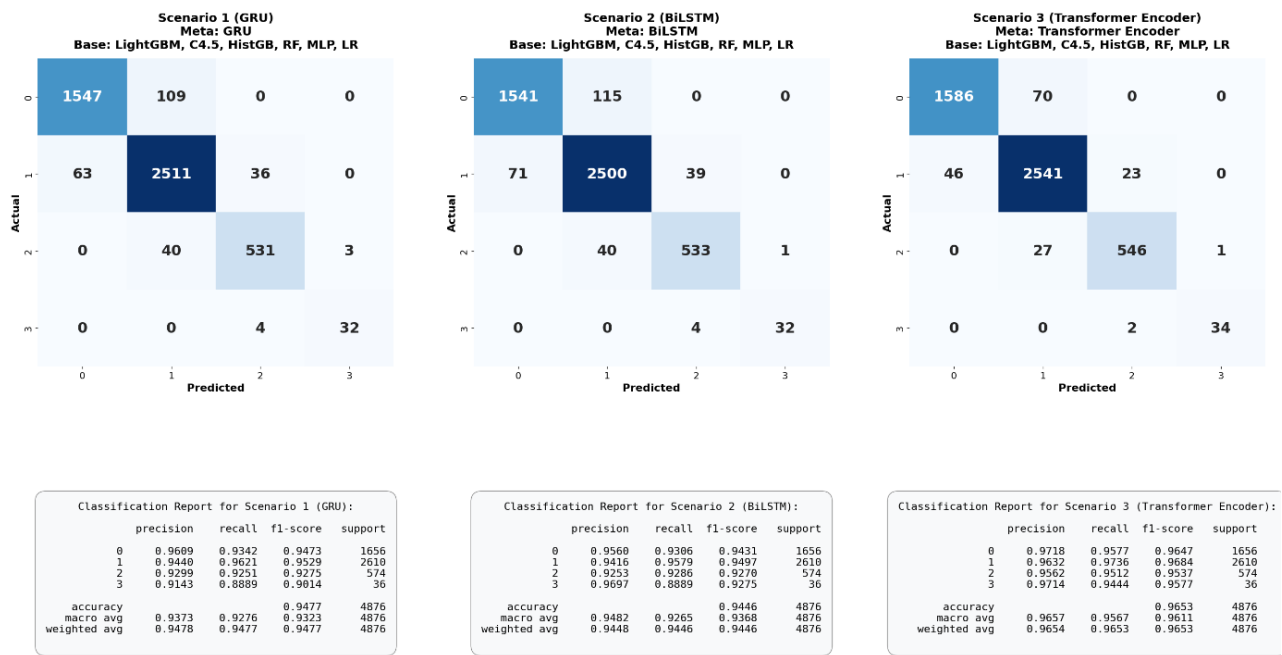


Figure 11 Comparison of Confusion Matrix and Classification Report

Based on the confusion matrix and classification report, the HSEL model demonstrates strong classification performance across all scenarios. The GRU scenario achieves an accuracy of 0.9477 with precision and recall of 0.9478 and 0.9477, respectively, indicating low misclassification as predictions are concentrated along the main diagonal. The BiLSTM scenario shows comparable performance with an accuracy of 0.9446, with slightly higher errors in intermediate classes. The Transformer Encoder scenario delivers the best results, achieving an accuracy of 0.9653 with precision, recall, and F1-score all at approximately 0.9653. Most misclassifications occur in the Moderate and High classes due to their larger data distribution, while the Very High class remains well identified despite its limited samples. High macro and weighted averages (above 0.94) indicate balanced performance across classes. These results confirm that the integration of accuracy weighting and Optuna improves overall accuracy and effectively handles data imbalance. To further evaluate class discrimination capability, ROC curve analysis is conducted for each scenario, as shown in Figure 12.

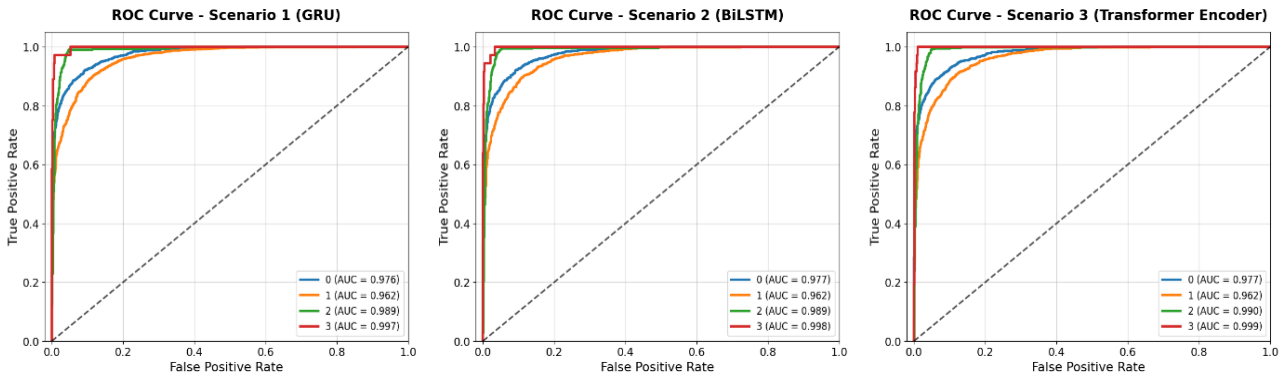


Figure 12 ROC Graph of HSEL Model

Based on the ROC curve in Figure 12, all scenarios performed very well with high AUC values for each class. In the GRU scenario, the AUC values ranged from 0.962 to 0.997, while in BiLSTM they ranged from 0.962 to 0.998, and the Transformer Encoder performed best with an AUC range of 0.962 to 0.999. The curves approaching the upper left corner indicate that the model has excellent ability to differentiate between classes with a low false positive rate and a high true positive rate. Furthermore, the high and consistent AUC values across all classes, including the minority class, indicate that the model is able to effectively address data imbalance without bias towards the majority class. The consistent performance across scenarios also indicates that the model does not experience overfitting, thus having good generalization capabilities on different data sets. To evaluate the model learning process, an analysis of the training and validation curves for each meta-model scenario was conducted. Figure 13 shows the results of the model training analysis.

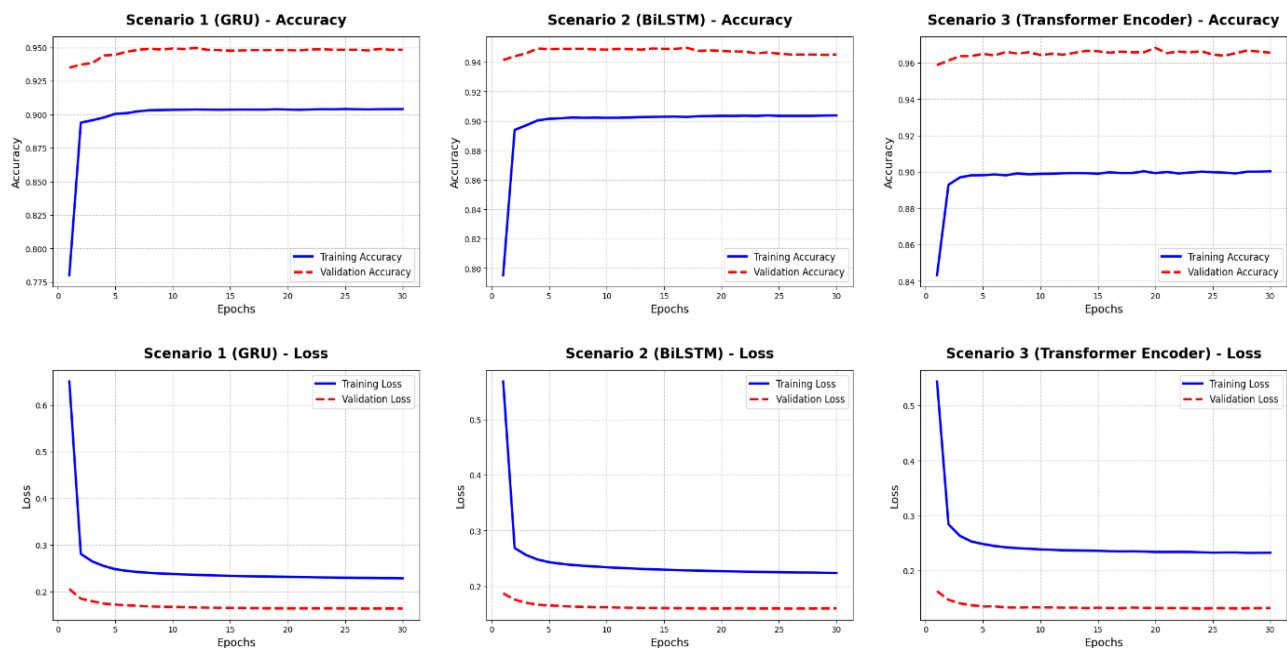


Figure 13 Training and Validation Performance Across HSEL Meta Models

Based on the training graph in Figure 13, all scenarios show a stable convergence process in the first few epochs. In the GRU and BiLSTM scenarios, training accuracy increases significantly from around 0.78–0.80 to around 0.90, while validation accuracy is in the higher range of around 0.94–0.95. The Transformer Encoder scenario shows a similar pattern with training accuracy reaching around 0.90 and validation accuracy approaching 0.96. In terms of loss, there is a significant decrease in the first epochs, where training loss drops from around 0.55–0.65 to around 0.23–0.25, and validation loss is lower in the range of 0.15–0.18. The consistent difference between the training and validation curves without any divergence indicates that the model does not experience overfitting and is able to generalize well to the validation data.

In evaluating the stability of model performance, testing was carried out using K-Fold Cross Validation with $K = 10$ in each meta model scenario. The following is Table 1 which shows the results of the performance analysis for each fold.

Table 1 Accuracy Table on Each Fold for Each Scenario

Fold	Scenario 1 (Meta: GRU)	Scenario 2 (Meta: BiLSTM)	Scenario 3 (Meta: Transformer)
Fold 1	96.52	95.29	96.72
Fold 2	93.03	93.44	94.88
Fold 3	93.65	93.65	95.49
Fold 4	95.90	96.52	98.77
Fold 5	98.57	97.95	99.10
Fold 6	93.65	92.62	95.08
Fold 7	94.87	94.66	97.33
Fold 8	94.66	94.66	96.51
Fold 9	92.61	91.99	94.66
Fold 10	94.25	93.84	95.89
Average (%)	94.77	94.46	96.53

Based on the results of K-Fold Cross Validation, it appears that the model performance is quite stable across all scenarios with relatively small variations in accuracy between folds. In the GRU scenario, accuracy ranges from 92.61% to 98.57% with an average of 94.77%, while BiLSTM is in the range of 91.99% to 97.95% with an average of 94.46%. The Transformer Encoder scenario shows the best performance with an accuracy range of 94.66% to 99.10% and the highest average of 96.53%. The relatively low variation between folds indicates that the model has good consistency and is not sensitive to data partitioning. Furthermore, the high average values across all scenarios confirm that the HSEL model has strong generalization capabilities and is able to maintain stable performance across various data subsets.

3.4 Real-Time IoT device testing results

System testing was conducted to evaluate the integration capability of the HSEL model with IoT devices to process sensor data in real time. This testing aimed to ensure that environmental data obtained from the sensors could be processed directly and produce accurate fire risk predictions. Figure 14 below shows the results of the real-time system implementation.

This dashboard showcases the integration of an HSEL model with an IoT system to process real-time environmental sensor data, including temperature, humidity, rainfall, wind speed, and soil moisture. Prediction results are visualized as fire risk information and location maps, enabling real-time monitoring of conditions and supporting rapid, data-driven decision-making.

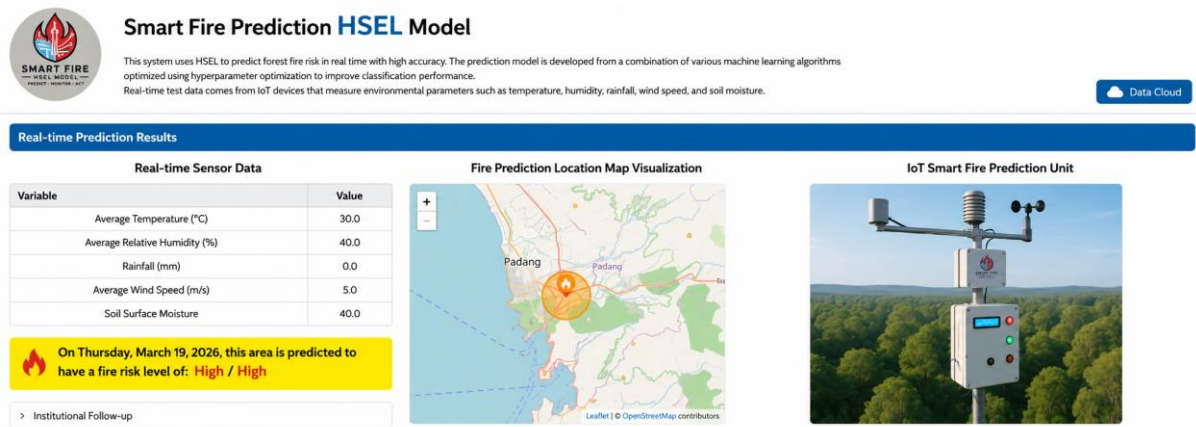


Figure 14 Real-time View of HSEL Model Testing Application with IoT Devices

3.5 Explainable AI

To improve the interpretability of the model, an Explainable AI (XAI) analysis was conducted using the SHAP method to identify the contribution of each feature to fire risk prediction. The following is Figure 15 of the results of the SHAP-based feature importance analysis.

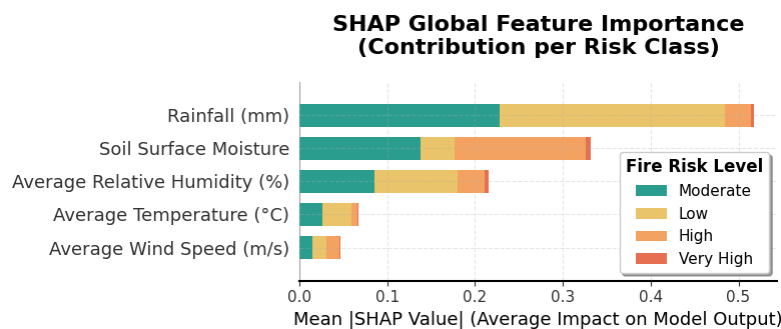


Figure 15 SHAP Global Feature Importance

Based on Figure 15 regarding the SHAP Global Feature Importance visualization, it can be seen that Rainfall has the largest contribution with a mean |SHAP| value reaching around 0.52, followed by Soil Surface Moisture at 0.33 and Average Relative Humidity at 0.21. Meanwhile, Average Temperature and Average Wind Speed show relatively smaller contributions with values below 0.10 each. In terms of class distribution, the contribution of Rainfall and Soil Surface Moisture is dominant in the Moderate and High classes, which indicates the important role of these two variables in determining the level of fire risk. In addition, the consistency of feature contributions across classes indicates that the model is able to capture stable relationship patterns without relying excessively on one particular feature. This indicates that the HSEL model is not only accurate, but also interpretable and able to provide transparent explanations regarding the main factors that influence predictions. To understand the contribution of features at the individual level, SHAP Explainer is used to identify the main factors that influence predictions in the highest risk class (Very High), below is Figure 16, the results of the local feature contribution analysis.

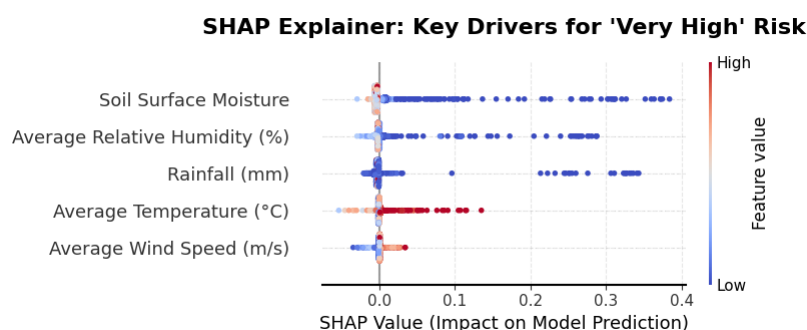


Figure 16 SHAP Explainer

The SHAP Explainer visualization shows that Soil Surface Moisture has the largest contribution with a value of around 0.35–0.40, followed by Rainfall up to 0.30 and Average Relative Humidity up to 0.28. Average Temperature has a moderate influence (≈ 0.15), while Average Wind Speed is relatively small (< 0.10). In general, dry environmental conditions are the main factor in increasing the risk of Very High, which confirms that the HSEL model is able to provide consistent and domain-relevant interpretations.

4. Discussion

Model performance comparisons at each stage of HSEL development were conducted through accuracy analysis between basic stacking, accuracy weighting, and hyperparameter tuning. These stages illustrate the gradual improvement process from basic methods to advanced optimizations in improving model performance. Table 2 shows the results of the model accuracy comparison.

Table 2 HSEL Model Scenario Accuracy Comparison

Scenario	Basic Stackingg	Accuracy Weighting	Hyperparameter Tuning
Scenario 1 (GRU)	87,80	92,72	94,77
Scenario 2 (BiLSTM)	87,80	92,55	94,46
Scenario 3 (Transformer)	88,26	94,20	96,53

Based on Table 2, each stage in HSEL significantly improves accuracy. In the GRU scenario, accuracy increases from 87.80% to 92.72% after applying accuracy weighting and reaches 94.77% after hyperparameter tuning. A similar trend is observed in BiLSTM, improving from 87.80% to 92.55% and then to 94.46%. The Transformer scenario achieves the highest performance, rising from 88.26% to 94.20% and reaching 96.53%. Overall, accuracy improvements range from 6.66% to 8.27%, demonstrating the effectiveness of combining accuracy weighting and hyperparameter optimization. To evaluate the contribution of HSEL, a comparison with previous studies using similar meteorological datasets is conducted. This analysis examines different combinations of base and meta models to assess their effectiveness in improving fire risk prediction accuracy, as presented in Table 3.

Table 3 Comparison of Accuracy Results with Previous Research Stacking Models Related to Meteorological Datasets for Detecting Fire Risk Levels

No	Author	Base model	Meta model	Result
1	[31]	RF, LR, Decision Tree, SVM, KNN, Naive Bayes	RF	94.48%
2	[32]	KNN, Decision tree, Random Forest, XGBoost	Neural Network (NN)	95.02%
3	[33]	AdaBoost, HGB, XGBoost, Gradient Boosting, LGBM	CatBoost	94.88%
4	[34]	SVM, KNN, NB, RF	LR	94.67%
5	[35]	XGBoost, LightGBM, CatBoost, AdaBoost	XGBoost	95.19%
6	[36]	KNN, SVR, DT, RF, Gradient Boosting (GB), Extra Trees (ET)	XGBoost	96.10%
7	[37]	Deep CNN, Bi-LSTM, Bi-GRU, hybrid BiLSTM-LSTM, hybrid CNN-BiLSTM, hybrid CNN-BiGRU, hybrid BiLSTM-BiGRU	LR	96.10%
8	[38]	Decision Tree, Gradient Boosting, RF, LR, SVM, Bayesian Networks, AdaBoost, XGBoost	KNN	95.43%
9	HSEL Model	LightGBM, C4.5, HistGB, MLP, LR, RF	Transformer	96.53%

Based on Table 3, the HSEL model demonstrated competitive performance and ranked highest with an accuracy of 96.53%, compared to previous studies that ranged from 94.48% to 96.10%. Several previous approaches relied on a combination of ensemble models such as XGBoost, Random Forest, and Neural Networks as meta-models, as well as variations of classical base models and boosting, but the resulting accuracy improvements were limited. HSEL surpassed the previous best performance by approximately 0.43%, indicating consistent improvement. This superiority stems from an adaptive weighting mechanism that dynamically adjusts the contribution of each base model, as well as hyperparameter optimization that optimally enhances the meta-model's performance. The HSEL model not only excels in accuracy but also demonstrates effectiveness in integrating various models to produce more accurate and stable predictions on meteorological datasets.

A comparative evaluation between conventional stacking methods and the HSEL approach was conducted to measure the extent of performance improvements achieved across various model configurations used in previous studies. This analysis included comparisons of various base model and meta-model combinations, ranging from classical machine learning algorithms to deep learning-based approaches. This approach aims to provide a comprehensive overview of the effectiveness of HSEL in improving prediction quality compared to traditional stacking methods. Table 4 shows the results of the model performance comparison.

Table 4 Comparison of Stacking Model Results with HSEL

Author	Base Model	Meta Model	Stacking	HSEL
[39]	KNN, LR, SVM, DT, RF, Extra-Tree, Naive Baye, AdaBoost, GBDT, XGBoost, LightGBM, CatBoost	LR	Accuracy: 92.10%	Accuracy: 94.05%
			Precision: 92.00%	Precision: 94.00%
			Recall: 91.00%	Recall: 93.00%
			F1-Score: 91.00%	F1-Score: 93.00%
			Accuracy: 93.20%	Accuracy: 95.05%
[40]	LR, SVM, RF, DT	XGBoost	Precision: 93.00%	Precision: 95.00%
			Recall: 92.00%	Recall: 94.00%
			F1-Score: 92.00%	F1-Score: 94.00%
[41]	RF, XGBoost, Gradient Boosting, FNN	FNN	Accuracy: 93.00%	Accuracy: 95.00%
			Precision: 93.00%	Precision: 95.00%
			Recall: 92.00%	Recall: 94.00%
			F1-Score: 92.00%	F1-Score: 94.00%

Table 4 (continued) Comparison of Stacking Model Results with HSEL

Author	Base Model	Meta Model	Stacking	HSEL
[42]	SVM, Naïve Bayes, LR, MLP, DNN, CNN, LSTM, BiLSTM, GRU, BiGRU	DNN	Accuracy: 91.30%	Accuracy: 93.40%
			Precision: 91.00%	Precision: 93.00%
			Recall: 90.00%	Recall: 92.00%
			F1-Score: 90.00%	F1-Score: 92.00%
			Accuracy: 93.80%	Accuracy: 95.60%
[29]	ANN, DNN, CNN	MLP	Precision: 94.00%	Precision: 96.00%
			Recall: 93.00%	Recall: 95.00%
			F1-Score: 93.00%	F1-Score: 95.00%

Based on Table 4, it can be seen that the application of HSEL consistently provides performance improvements compared to conventional stacking methods in all previous studies. Accuracy improvements range from 1.80% to 2.00%, as in the previous study[39] which increased from 92.10% to 94.05%, and research[29] from 93.80% to 95.60%. The same pattern of improvement also occurs in the precision, recall, and F1-score metrics, which each increased by approximately 2%. This indicates that HSEL is able to improve the overall prediction quality, not only in accuracy but also in classification balance. Additional evaluations were conducted to test the generalization ability of the HSEL model on various types of datasets with different characteristics. This testing included both structured and unstructured data with varying numbers of classes. The following Table 5 shows the results of applying the HSEL model to four different types of datasets.

Table 5 Comparison of HSEL Model Accuracy Results in Testing 4 Different Datasets

No	Dataset	Type	Class	Accuracy (%)
1	Meteorological Data	Structured	4	96.18
2	Flood Historical Data	Structured	4	95.98
3	Social Media Reviews and Disaster-Related News	Unstructured	9	96.47
4	Social Media Reviews and Weather and Fire Disaster News	Unstructured	4	97.55

Table 5 shows that the HSEL model maintains high performance across various datasets, with accuracy ranging from 95.98% to 97.55%. On structured datasets such as meteorology and historical flood data, the model achieves accuracies of 96.18% and 95.98%, respectively, demonstrating stable performance on numerical data. Meanwhile, on unstructured datasets such as social media reviews and news, the model maintains excellent performance, with an accuracy of 96.47% for 9 classes, increasing to 97.55% in the 4-class scenario. These results demonstrate that HSEL has high adaptability to various data types and classification complexities. The model is not only effective in specific domains but also demonstrates strong flexibility and robustness across various prediction scenarios.

Despite the promising performance of the proposed HSEL model, several limitations should be acknowledged. First, the model was developed and evaluated using meteorological data collected from a single geographical region, which may limit its generalization to areas with substantially different climatic characteristics. Second, although the proposed framework incorporates heterogeneous machine learning models and deep learning based meta learners, the computational cost of training increases due to hyperparameter optimization and multiple meta learner evaluations. Third, the current study relies primarily on meteorological variables and does not incorporate additional spatial information such as satellite imagery, vegetation indices, or topographical features that may further improve prediction performance. Finally, although SHAP improves model interpretability, further investigation is required to evaluate the robustness of explanations across different environmental conditions and larger real world deployments. Future work will focus on multi regional validation, multimodal data integration, and lightweight HSEL architectures for resource constrained IoT environments.

5. Conclusions

This study proposes the HSEL model as a new approach to improve the performance of real-time fire risk prediction based on meteorological data and IoT integration. Experimental results show that the basic stacking method produces accuracy in the range of 87.80% to 88.26%, which then increases significantly after the application of exponential accuracy weighting to 92.55% to 94.20%. Furthermore, through hyperparameter optimization using Optuna, the model performance is further improved to reach the highest accuracy of 96.53%. This improvement indicates that each stage in the HSEL makes a significant contribution to improving the quality of predictions. The application of the parameter $\alpha = 5$ in the exponential weighting mechanism is proven effective in increasing the difference in contributions between base models, so that models with higher performance receive more dominant weights and produce more optimal prediction combinations. In addition, the HSEL model shows stable and consistent performance across various meta-model scenarios and has good generalization capabilities across various types of datasets, both structured and unstructured. SHAP-based Explainable AI (XAI) analysis also demonstrated that the model is capable of providing clear interpretations of key factors influencing predictions, thereby enhancing transparency and trust in the model's results. Integration with an IoT system further strengthens this research's contribution in supporting real-time fire risk prediction based on actual environmental data. Overall, HSEL not only provides improved performance in terms of accuracy but also offers a more adaptive, robust, and interpretable approach compared to previous methods. Further research can be directed at developing multilayer stacking models with the application of accuracy weighting to further optimize model performance.

6. References

- [1] Sengdara K, Sukendro A, Heridadi. The role of the Government of Riau provincial in dealing with forest and land fires. IOP Conf Ser Earth Environ Sci. 2023;1173:012063.

- [2] Yuliani F, Zulkarnaini. Land and forest fire control strategy through inter-organizational network in efforts to implement disaster management in Riau province, Indonesia. *IOP Conf Ser: Earth Environ Sci.* 2024;1419:012072.
- [3] Zaidi A. Predicting wildfires in Algerian forests using machine learning models. *Heliyon.* 2023;9(7):e18064.
- [4] Manalu DR, Sitompul OS, Mawengkang H, Zarlis M. Model classification of fire weather index using the SVM-FF method on forest fire in North Sumatra, Indonesia. *Int J Adv Comput Sci Appl.* 2023;14(8):329-37.
- [5] Singh KR, Neethu KP, Madhurekaa K, Harita A, Mohan P. Parallel SVM model for forest fire prediction. *Soft Comput Lett.* 2021;3:100014.
- [6] Willem T, Shitov VA, Luecken MD, Kilbertus N, Bauer S, Piraud M, et al. Biases in machine-learning models of human single-cell data. *Nat Cell Biol.* 2025;27(3):384-92.
- [7] Mavrogiorgos K, Kiourtis A, Mavrogiorgou A, Menychtas A, Kyriazis D. Bias in machine learning: a literature review. *Appl Sci.* 2024;14(19):8860.
- [8] Ramampandra EC, Scheidegger A, Wydler J, Schuwirth N. A comparison of machine learning and statistical species distribution models: quantifying overfitting supports model interpretation. *Ecol Modell.* 2023;481:110353.
- [9] Lubis A, Irawan Y, Junadhi, Defit S. Leveraging K-Nearest neighbors with SMOTE and boosting techniques for data imbalance and accuracy improvement. *J Appl Data Sci.* 2024;5(4):1625-38.
- [10] Mujahid M, Kina E, Rustam F, Villar MG, Alvarado ES, De La Torre Diez I, et al. Data oversampling and imbalanced datasets: an investigation of performance for machine learning and feature engineering. *J Big Data.* 2024;11(1):87.
- [11] Song L, Zhao J, Li Y, Liu L, Duan J, He L, et al. Early detection of wheat powdery mildew: a multi-source in situ remote sensing approach enabled by stacked ensemble learning. *Artif Intell Agric.* 2026;16(1):124-38.
- [12] Li Y, Guo S, Chen P, Chen L, Mou J. A stacking-based ensemble learning model for intelligent ship trajectory interpolation. *Reliab Eng Syst Saf.* 2026;265:111615.
- [13] Putri AK, Suparwito H. Uji algoritma stacking ensemble classifier pada kemampuan adaptasi mahasiswa baru dalam pembelajaran online. *KONSTELASI: Konvergensi Teknologi dan Sistem Informasi.* 2023;3(1):1-12. (In Indonesian)
- [14] Irawan Y, Defit S, Sovia R. Rotational stacking ensemble with accuracy-based weighting for real-time forest fire risk prediction. *Int J Robot Control Syst.* 2026;6(3):1713-36.
- [15] Devis Y, Muhamadiyah, Yulanda, Irawan Y, Wahyuni R. Optimization of machine learning models for risk prediction of DHF spread to support management strategies in urban areas. *J Appl Data Sci.* 2025;6(4):2407-20.
- [16] Herianto, Kurniawan B, Hartomi ZH, Irawan Y, Anam MK. Machine learning algorithm optimization using stacking technique for graduation prediction. *J Appl Data Sci.* 2024;5(3):1272-85.
- [17] Daza A, Ponce Sánchez CF, Apaza-Perez G, Pinto J, Zavaleta Ramos K. Stacking ensemble approach to diagnosing the disease of diabetes. *Inform Med. Unlocked.* 2024;44:101427.
- [18] Irawan Y, Defit S, Sovia R. Optimizing stacking ensemble learning to enhance meta-model performance in forest fire risk level detection. 1st International Conference on Emerging Trends in Information Systems and Informatics (ICETISI); 2025 Dec 1-2; Jakarta, Indonesia. USA: IEEE; 2025. p. 1-6.
- [19] Reza MS, Amin R, Yasmin R, Kulsum W, Ruhi S. Improving diabetes disease patients classification using stacking ensemble method with PIMA and local healthcare data. *Heliyon.* 2024;10(2):e24536.
- [20] Li C, Xu B, Chen Z, Huang X, He J, Xie X. A stacking model-based classification algorithm is used to predict social phobia. *Appl Sci.* 2024;14(1):433.
- [21] Lu M, Hou Q, Qin S, Zhou L, Hua D, Wang X, et al. A stacking ensemble model of various machine learning models for daily runoff forecasting. *Water.* 2023;15(7):1265.
- [22] Suresh J. Fire-Fighting Robot. 2017 International Conference on Computational Intelligence in Data Science (ICCIDS); 2017 Jun 2-3; Chennai, India. USA: IEEE; 2017. p. 1-4.
- [23] Kurniawan B, Wahyuni R, Yulanda, Irawan Y, Yuhandri MH. Multimodal deep learning and iot sensor fusion for real-time beef freshness detection. *J Appl Data Sci.* 2025;6(4):2921-37.
- [24] Muhaimin A, Edriyansyah, Wahyat, Irawan Y, Wahyuni R. Optimized IoT-based multimodal fusion for early forest fire detection and prediction. *ECTI Trans Comput Inf Technol.* 2025;19(4):569-82.
- [25] Kharisma RS, Setiyansah A. Fire early warning system using fire sensors, microcontroller, and SMS gateway. *J Robot Control.* 2021;2(3):165-9.
- [26] Irawan Y, Muzawi R, Alamsyah A, Renaldi R, Elisawati, Nurhadi, 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. *Ilk J Ilm.* 2023;15(3):445-54.
- [27] Irawan Y, Wahyuni R, Muhandi, Fonda H, Hamzah ML, Muzawi R. Real time system monitoring and analysis-based internet of things (IoT) technology in measuring outdoor air quality. *Int J Interact Mob Technol.* 2021;15(10):224-40.
- [28] Stracqualursi L. A lightweight exponential-weighted ensemble for crop recommendation. *J Agric Biol Environ Stat.* 2025:1-17.
- [29] Boateng VA, Yang B. A global modeling pruning ensemble stacking with deep learning and neural network meta-learner for passenger train delay prediction. *IEEE Access.* 2023;11:62605-15.
- [30] Almohimeed A, Saad RMA, Mostafa S, El-Rashidy NM, Farrag S, Gaballah A, 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. *IEEE Access.* 2023;11:123173-93.
- [31] Alabdulhafith M, Saleh H, Elmannai H, Ali ZH, El-Sappagh S, Hu JW, et al. A clinical decision support system for edge/cloud ICU readmission model based on particle swarm optimization, ensemble machine learning, and explainable artificial intelligence. *IEEE Access.* 2023;11:100604-21.
- [32] Rezaei Melal S, Aminian M, Shekarian SM. A machine learning method based on stacking heterogeneous ensemble learning for prediction of indoor humidity of greenhouse. *J Agric Food Res.* 2024;16:101107.
- [33] Mondal S, Ghosh S, Nag A. Brain stroke prediction model based on boosting and stacking ensemble approach. *Int J Inf Technol.* 2024;16(1):437-46.
- [34] Rahim MA, Hossain MA, Hossain MN, Shin J, Yun KS. Stacked ensemble-based type-2 diabetes prediction using machine learning techniques. *Ann Emerg Technol Comput.* 2023;7(1):30-9.
- [35] Kalabarige LR, Rao RS, Pais AR, Gabralla LA. A boosting-based hybrid feature selection and multi-layer stacked ensemble learning model to detect phishing websites. *IEEE Access.* 2023;11:71180-93.

- [36] Kaleem W, Tewari S, Fogat M, Martyushev DA. A hybrid machine learning approach based study of production forecasting and factors influencing the multiphase flow through surface chokes. *Petroleum*. 2023;10(2):354-71.
- [37] Al Shamsi AA, Abdallah S. Ensemble stacking model for sentiment analysis of Emirati and Arabic dialects. *J King Saud Univ - Comput Inf Sci*. 2023;35(8):101691.
- [38] Daza A, Arroyo-Paz, Bobadilla J, Apaza O, Pinto J. Stacking ensemble learning model for predict anxiety level in university students using balancing methods. *Inform Med Unlocked*. 2023;42:101340.
- [39] Zhao N, Li X, Ma Y, Wang H, Lee SJ, Wang J. Improved stacked ensemble with genetic algorithm for automatic ECG diagnosis of children living in high-altitude areas. *Biomed Signal Process Control*. 2024;87:105506.
- [40] Om Kumar CU, Singh I, Suguna M. Optimizing patient recruitment for clinical trials : a hybrid classification model and game-theoretic approach for strategic interaction. *IEEE Access*. 2024;12:10254-80.
- [41] Niyogisubizo J, Liao L, Nziyumva E, Murwanashyaka E, Claver P. Predicting student's dropout in university classes using two-layer ensemble machine learning approach: a novel stacked generalization. *Comput Educ Artif Intell*. 2021;3:100066.
- [42] Serrano-Guerrero J, Alshouha B, Bani-Doumi M, Chiclana F, Romero FP, Olivas JA. Combining machine learning algorithms for personality trait prediction. *Egypt Inform J*. 2024;25:100439.