Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 10, Issue 1, Januari 2026
DOI : https://doi.org/10.33395/sinkron.v10i1.15546
e-ISSN : 2541-2019
p-ISSN : 2541-044X

Classification of Smoke Detection Based on Sensor Data Using Machine Learning

Marcus Liecero[1], Robet[2], Jackri Hendrik[3]
[1,2,3)]Teknik Informatika, STMIK TIME, Medan, Indonesia
[1)]da.liecero@gmail.com, [2)]robertdetime@gmail.com, [3)]jackri.hendrik@gmail.com

Submitted :Nov 5, 2025 | **Accepted** : Dec 3, 2025 | **Published** : Jan 02, 2026

Abstract: Smoke detection plays a critical role in preventing fire-related hazards, particularly in intelligent monitoring and early warning systems. Conventional smoke sensors often exhibit limited responsiveness in dynamic environmental conditions, prompting the adoption of IoT-based sensor data combined with machine learning techniques. This study presents a comparative evaluation of four supervised classification algorithms, K-Nearest Neighbors (KNN), Decision Tree, Random Forest, and Gradient Boosting, using the Smoke Detection Dataset from Kaggle. The methodology integrates SMOTE to address class imbalance and Z-score normalization for feature standardization. Hyperparameter tuning was performed using GridSearchCV with 5-fold cross-validation, and model performance was assessed based on accuracy and execution time. Experimental results show that KNN achieved the highest accuracy (98.33%) with the lowest execution time (0.0327 s), whereas Decision Tree recorded the lowest accuracy (84.17%) but remained computationally fast (0.0406 s). Random Forest and Gradient Boosting demonstrated strong predictive capability (97.22% and 96.94%, respectively), but at higher computational costs (1.4338 s and 8.3819 s, respectively). Almost all models achieved perfect scores (1.00) for precision, recall, and F1-score following SMOTE-based balancing, except KNN which obtained slightly lower values (0.99). The findings indicate a trade-off between predictive performance and computational efficiency, suggesting that lightweight models such as KNN are better suited for real-time IoT-based smoke detection. In contrast, ensemble models may be more appropriate for backend analysis. This research contributes an integrated evaluation framework that combines data rebalancing, multi-model benchmarking, and time-based performance analysis, providing practical insights for the development of responsive and scalable early smoke detection systems.

Keywords: Decision Tree, Gradient Boosting, IoT sensors, K-Nearest Neighbors, Machine learning, Random Forest, Smoke Detection, SMOTE

INTRODUCTION

Smoke is a mixture of gases and particles resulting from incomplete combustion. Its presence is often an early indicator of potential hazards, such as household, forest, or industrial system fires (He et al. 2025). In addition to threatening physical safety, smoke exposure negatively affects human health, both in the short term, such as respiratory and eye irritation, and in the long term, potentially triggering chronic lung diseases (Jamal et al. 2025). Therefore, early smoke detection is crucial for enabling timely warning and response systems (Carletti et al. 2024).

Conventional smoke detection technologies that utilize ionization or photoelectric principles are often less responsive to dynamically changing environmental conditions, making them less reliable for proactive detection (Vasconcelos et al. 2024). To address these limitations, recent studies have shifted towards the utilization of environmental sensor data, including temperature, humidity, air pressure, carbon monoxide (CO), and particulate concentration, as a more adaptive and flexible approach (Deepa et al. 2022; Erkmen and Ayrancı 2024). Along with technological advancements, such sensor data can now be combined with machine learning algorithms to develop intelligent, adaptive, and real-time smoke detection systems (Liu et al. 2024; Wang et al. 2023). Classification methods such as K-Nearest Neighbors (KNN), Decision Trees, Random Forests, and Gradient Boosting have demonstrated the ability to learn and classify complex patterns in sensor-based environmental data effectively.

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 10, Issue 1, Januari 2026
DOI : https://doi.org/10.33395/sinkron.v10i1.15546
e-ISSN : 2541-2019
p-ISSN : 2541-044X

However, previous studies still exhibit several methodological limitations. First, most existing research only focuses on one or two algorithms (Vasconcelos et al. 2024; Wang et al. 2023), limiting the ability to assess model performance comprehensively. Second, performance evaluation predominantly emphasizes accuracy, while the efficiency of prediction time, crucial for real-time detection scenarios, is rarely considered (Handoko and Aditya 2025; Jamal et al. 2025). Third, only a limited number of studies investigate the effectiveness of data balancing techniques such as SMOTE on model performance (Fulazzaky, Saefuddin, and Soleh 2024; Hairani, Saputro, and Fadli 2020; Handoko and Aditya 2025).

To the best of our knowledge, no study has yet provided a comprehensive comparison of four classical machine learning algorithms (KNN, Decision Tree, Random Forest, and Gradient Boosting) while simultaneously evaluating the trade-off between prediction accuracy and computational time on imbalanced environmental sensor data for smoke detection. This constitutes the main research gap addressed in this work.

This research focuses on the simultaneous evaluation of accuracy and prediction efficiency across four machine learning classifiers using IoT-based environmental sensor data rebalanced with the SMOTE technique. In addition, the study provides a comparative analysis of the generalization capabilities of each model under data imbalance mitigation, offering empirical evidence on the effectiveness of SMOTE for sensor-based smoke detection. This integrated methodological setup, combining multi-algorithm benchmarking, time-efficiency analysis, and data rebalancing, has not been explicitly explored in prior work.

Based on these gaps and objectives, this research aims to evaluate the performance of four machine learning classification algorithms, KNN, Decision Tree, Random Forest, and Gradient Boosting, in detecting smoke using environmental sensor data. The evaluation considers accuracy, precision, generalization capability, and computational efficiency to identify the most effective model for the development of smart, adaptive, and real-time smoke detection systems (Julian, Dewantara, and Wahyuni 2024; Rajoli et al. 2024). The results of this study are expected to provide practical guidance for implementing IoT-based early warning systems that are not only accurate but also time-efficient at predicting fire-related hazards.

LITERATURE REVIEW

The advancement of smoke detection technology has progressed through two dominant research directions: vision-based deep learning models and sensor-based machine learning systems. (Liu et al. 2024) demonstrate that deep learning architectures such as Transformer-Boosted U-Net achieve high accuracy in image-based segmentation in complex environments, highlighting their suitability for remote sensing applications. In contrast, (Deepa et al. 2022) emphasize that IoT sensor-based systems, built on features such as temperature, humidity, gas concentration, and particulate matter, are more favorable for real-time field implementation due to lower computational latency. Taken together, these findings suggest a fundamental trade-off: vision-based deep learning excels in detection precision and spatial representation, whereas sensor-based approaches are more practical for early warning applications where rapid prediction and low-cost deployment on embedded hardware are crucial.

A similar contrast emerges when comparing modeling strategies across studies. (Vasconcelos et al. 2024) confirm that CNN and LSTM architectures are robust for smoke detection in dynamic visual environments, while (Wang et al. 2023) stress the importance of high-volume annotated datasets to exploit the capacity of deep models fully. However, these approaches depend heavily on large-scale image data and centralized computing resources, which may be less suitable in environments where dense camera networks are unavailable but distributed environmental sensors can be easily deployed. Complementing this line of work, (Li et al. 2023) report improved spatial detection using Random Forest combined with Sub-pixel Mapping, reinforcing the potential of lightweight traditional machine learning methods for environmental data. This contrast indicates that, while deep learning dominates image-based smoke detection, classical machine learning remains a compelling alternative for sensor-based settings in which the input space is tabular, the feature set is interpretable, and computational budgets are constrained.

In evaluating model performance under data imbalance, both (Fulazzaky et al. 2024; Handoko and Aditya 2025) demonstrate the superiority of SMOTE-based enhancement methods, with the latter quantitatively proving accuracy improvements up to 15%. These results suggest that synthetic oversampling can effectively mitigate underrepresentation of the minority class in hazard detection tasks. However, neither study investigates how such accuracy gains affect real-time prediction efficiency, nor do they systematically compare multiple model families under the same rebalanced dataset. At the same time, ensemble-based methods such as Random Forest and Gradient Boosting have shown generalization capabilities and robustness to noisy sensor readings (Julian et al. 2024; Zhang, Tan, and Robert 2024), but they generally involve higher computational costs, which may be disadvantageous for edge-based IoT implementations with limited CPU and memory. Consequently, the interaction between imbalance handling (e.g., SMOTE), model complexity, and prediction speed remains insufficiently explored in the context of smoke detection.

Synthesizing current research trends from 2023 to 2025, recent studies clearly prioritize accuracy optimization through deep learning and ensemble models, often on visual data, while only a limited number explicitly analyze

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 10, Issue 1, Januari 2026
DOI : https://doi.org/10.33395/sinkron.v10i1.15546
e-ISSN : 2541-2019
p-ISSN : 2541-044X

the trade-off between predictive performance and computational efficiency. Moreover, most existing works either focus on visual inputs or evaluate a single algorithm (or a narrow subset of models) rather than performing systematic multi-model comparison using environmental sensor features. This situation motivates the present study to focus on four representative classical machine learning algorithms, K-Nearest Neighbors, Decision Tree, Random Forest, and Gradient Boosting, for sensor-based smoke detection. K-Nearest Neighbors and Decision Trees provide relatively low computational complexity and high interpretability, making them suitable baselines for resource-constrained IoT devices. Random Forest and Gradient Boosting, on the other hand, represent stronger ensemble learners with proven generalization but higher computational demands. By applying SMOTE to rebalance environmental sensor data and jointly evaluating these four algorithms on both accuracy and prediction speed, this research directly addresses the identified gaps. It provides a conceptually grounded basis for selecting models that are not only accurate but also computationally efficient for real-time smoke detection applications.

METHOD

Research Stages

This research was conducted through a structured workflow that began with data collection and preprocessing, followed by data splitting and SMOTE-based balancing. The models (KNN, Decision Tree, Random Forest, and Gradient Boosting) were trained and optimized using hyperparameter tuning, then evaluated using multiple performance metrics, learning curve analysis, and prediction time assessment. The process continued with threats-to-validity analysis, discussion, and conclusion, as illustrated in Figure 1.

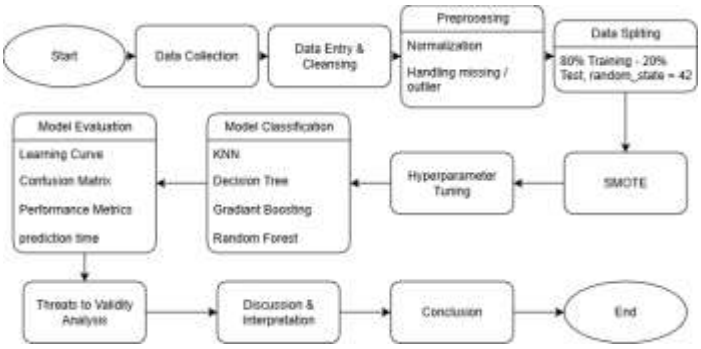

Fig 1. Research Stages

The data used in this study were obtained from Kaggle, titled "Smoke Detection Dataset," developed by Deepcontractor. The dataset link is available at https://www.kaggle.com/datasets/deepcontractor/smoke-detection-dataset. This dataset is a simulated air quality monitoring system based on Internet of Things (IoT) sensors, designed to detect the presence of smoke in an environment (Bhamra et al. 2023; Wang et al. 2023). The next stage is data preprocessing, which includes data cleaning (handling missing values), normalization of numerical features, and data balancing using the SMOTE technique to address class imbalance (Fulazzaky et al. 2024; Hairani et al. 2020; Handoko and Aditya 2025). SMOTE was selected over other resampling techniques, such as ADASYN and SMOTE-Tomek, because it generates synthetic samples by interpolating minority class neighbors while preserving the feature distribution, reducing noise injection risk (Talukder et al. 2024). Previous studies demonstrated that SMOTE achieves higher classification stability in environmental sensor-based detection models compared to ADASYN, which tends to generate noisy instances (Fulazzaky et al. 2024; Handoko and Aditya 2025). Moreover, SMOTE-Tomek focuses on both over and undersampling, but may remove borderline samples, which are essential in early smoke detection scenarios.

Z-score normalization was applied to scale all numerical sensor features to a standardized range before model training. This technique transforms each feature so that its mean is zero and its standard deviation is 1, reducing the influence of extreme values and improving algorithm convergence. The standardization formula is shown below:

$$\sigma z = (x - \mu) / \sigma \quad (1)$$

Where x represents the feature value, μ is the mean, and σ is the standard deviation of the feature in the training set. Standardization was applied after the data split to prevent data leakage, ensuring that the parameters μ and σ were calculated only from the training data.

To ensure reliable experimental evaluation, the dataset was split into 80% for training and 20% for testing, using random_state = 42 to guarantee reproducibility. SMOTE balancing was applied only to the training data, avoiding information leakage into the test set that could artificially inflate performance scores.

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 10, Issue 1, Januari 2026
DOI : https://doi.org/10.33395/sinkron.v10i1.15546
e-ISSN : 2541-2019
p-ISSN : 2541-044X

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

Fig 2. Split Data

Once the data were ready, the following stage involved model training, which analyzed four machine learning algorithms.

Hyperparameter optimization was conducted using GridSearchCV with 5-fold cross-validation. The optimal configurations obtained were as follows: KNN (n_neighbors=3, metric='minkowski', weights='distance'), Decision Tree (criterion='gini', max_depth=10), Random Forest (n_estimators=150, max_depth=15), and Gradient Boosting (learning_rate=0.1, n_estimators=200, subsample=0.8). The highest cross-validation accuracy was achieved by the Gradient Boosting model (0.99994).

Hyperparameter optimization for the Random Forest model was conducted using a focused grid search strategy. Although the initial range of n_estimators (100–300) was based on prior literature (Fulazzaky et al. 2024; Li et al. 2023), preliminary testing revealed that configurations above 200 substantially increased computation time (1–3 hours per setup) and posed resource constraints in Google Colab. Consequently, the search space was refined to only 150 and 200, enabling more efficient evaluation without compromising accuracy. Similarly, the selection of other parameters, such as max_depth, was restricted to the most empirically relevant values to balance model performance and computational feasibility.

Following the hyperparameter optimization procedure, the theoretical background of each classification algorithm is briefly outlined to justify its methodological relevance and to provide conceptual context for its application in sensor-based smoke detection.

K-Nearest Neighbors (KNN)

KNN is a distance-based classification method that assigns a new data point to the class with the most votes among its *k* nearest neighbors. The distance between data points is calculated using the Euclidean distance formula (Erkmen and Ayrancı 2024; Jamal et al. 2025).

$$\boldsymbol{d(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}} \quad (2)$$

Random Forest

Random Forest is an ensemble learning method that constructs multiple decision trees to improve predictive performance. Each tree is trained using a random subset of data and features, and the final result is determined through majority voting (Fulazzaky et al. 2024; Li et al. 2023)

$$\boldsymbol{\hat{y} = mode(h_1(x), h_2(x), \ldots, h_T(x))} \quad (3)$$

Decision Tree

A Decision Tree is a tree-structured model that splits data based on the most informative features. One of the most commonly used splitting methods is the Gini impurity (Erkmen and Ayrancı 2024; Syukron, Santoso, and Widiharih 2020)

$$\boldsymbol{Gini(t) = 1 - \sum_{i=1}^{c}(P_i)^2} \quad (4)$$

Gradient Boosting

Gradient Boosting is an ensemble technique that builds models sequentially, where each new model attempts to correct the errors of the previous ones by minimizing a loss function through Gradient optimization (Fulazzaky et al., 2024; Talukder et al., 2024; Tao et al., 2021).

$$\boldsymbol{\hat{y}(x) = \sum_{m=1}^{M}\gamma_m h_m(x)} \quad (5)$$

Each model was trained on the preprocessed training data and adjusted using the optimal parameters. The process then continued with model evaluation and classification, where the performance of each model was assessed on the test data using evaluation metrics such as accuracy, precision, recall, F1-score, and a confusion matrix (Jamal et al. 2025; Rajoli et al. 2024).

The evaluation results were then used in the visualization stage, where model performance was presented as graphs or comparison tables to facilitate analysis and interpretation. The entire process culminated in the final

*name of corresponding author

e-ISSN : 2541-2019
p-ISSN : 2541-044X

stage, in which conclusions and recommendations were drawn from findings obtained throughout the classification and evaluation (Julian et al. 2024; Zhang et al. 2024).

Model Evaluation Framework

The models were evaluated using four main metrics: Accuracy, Precision, Recall, and F1-Score. Each of these metrics complements the others and is especially important for imbalanced datasets, such as smoke detection (Alharbi, Ouarbya, and Ward 2022; Fulazzaky et al. 2024; Hairani et al. 2020)

$$\boldsymbol{Accuracy} = \frac{\boldsymbol{TP+TN}}{\boldsymbol{TP+TN+FP+FN}} \quad (6)$$

$$\boldsymbol{Precision} = \frac{\boldsymbol{TP}}{\boldsymbol{TP+FP}} \quad (7)$$

$$\boldsymbol{Recall} = \frac{\boldsymbol{TP}}{\boldsymbol{TP+FN}} \quad (8)$$

$$\boldsymbol{F1} = \frac{\boldsymbol{2 \cdot Precision \cdot Recall}}{\boldsymbol{Precision+Recall}} \quad (9)$$

To ensure the robustness and scientific validity of the experimental results, several potential threats to validity were identified and addressed during the research process. These include algorithmic bias, sensor variability, hardware dependency, and data drift, each of which may affect the performance and generalization of the models in real-world scenarios. The evaluation process not only focused on performance metrics but also assessed methodological rigor by identifying potential validity threats and their mitigation strategies, as outlined in Table 1.

Table 1. Risk Analysis on Model Validity and Mitigation Strategies

Threat to Validity	Potential Impact	Mitigation Strategy
Algorithmic bias	Overfitting to the dominant class, leading to misleading accuracy	Applied SMOTE to balance classes and used 5-fold cross-validation to ensure generalization
Sensor noise / environmental fluctuations	Unstable model predictions due to anomalies in sensor readings	Implemented data cleaning and anomaly detection using z-score filtering during preprocessing
Hardware dependency on computational performance	Variability in prediction speed due to dynamic cloud resource allocation	Experiments were executed in the same Google Colab runtime session (free tier) to maintain consistency. Time-based comparisons were analyzed relatively rather than absolutely
Data drift in real-world IoT deployment	Reduced long-term performance due to environmental changes	Recommended periodic retraining and monitoring of model performance in operational environments

These mitigation strategies were incorporated to enhance the reliability of the findings and increase the applicability of the model in real-time IoT-based early warning systems.

Although ensemble-based algorithms such as Random Forest and Gradient Boosting generally offer higher robustness and generalization, they require greater computational resources, which may limit their applicability in real-time edge-based IoT environments. Conversely, lightweight models such as K-Nearest Neighbors and Decision Trees provide lower prediction latency but may be more sensitive to noisy or imbalanced sensor data. Therefore, including both classification accuracy and prediction time in the evaluation framework is essential for determining the most effective algorithm for real-time smoke detection. The following section presents the experimental results and comparative analysis of the four classification models based on the outlined evaluation framework, highlighting both predictive accuracy and computational efficiency.

RESULT

This section presents the experimental results from the evaluation of the four machine learning models, conducted according to the predefined methodology. The findings are structured into data inspection, preprocessing impact, model learning behavior, and comparative performance analysis in terms of predictive accuracy and execution time.

Data Collection

The dataset consists of 62,630 observations with 16 input features and one target label (Fire Alarm). The input features include temperature, relative humidity, air pressure, concentrations of carbon monoxide (CO), ethanol,

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 10, Issue 1, Januari 2026
DOI : https://doi.org/10.33395/sinkron.v10i1.15546
e-ISSN : 2541-2019
p-ISSN : 2541-044X

butane, and sensor measurements from the MQ-135 gas sensor. Additionally, the dataset includes timestamp information and device location metadata.

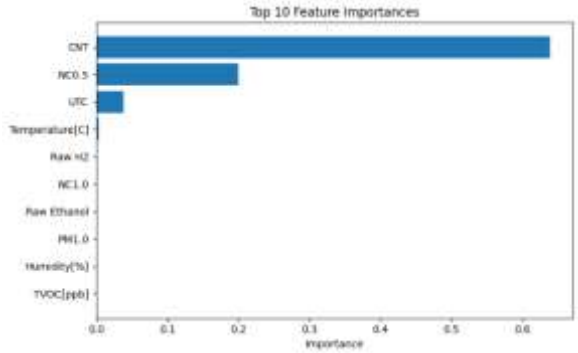

Fig 3. Dataset Label

```
Cek nilai kosong:
UTC                0
Temperature[C]     0
Humidity[%]        0
TVOC[ppb]          0
eCO2[ppm]          0
Raw H2             0
Raw Ethanol        0
Pressure[hPa]      0
PM1.0              0
PM2.5              0
NC0.5              0
NC1.0              0
NC2.5              0
CNT                0
Fire Alarm         0
dtype: int64
```

Fig 4. Missing Value Check

During dataset inspection, no missing values or empty columns were detected, as shown in Figure 4. The target label is a binary classification: 0 (False) or 1 (True). A total of 44,757 observations belong to class 1 (True) and 17,873 observations belong to class 0 (False), as presented in Figure 5 and Table 2.

Table 2. Total Values for 1 and 0

Fire alarm	Total	Percent
1	44757	71.46
0	17873	28.54

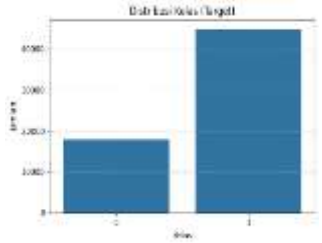

Fig 5. Comparison Diagram of Class 1 and Class 0

Preprosessing

At the preprocessing stage, the dataset was imbalanced, with 44,757 observations labeled as class 1 (True) and 17,873 as class 0 (False). To address the imbalance, the Synthetic Minority Over-sampling Technique (SMOTE) was applied. After resampling, both classes contained 35,825 instances each. The distribution before and after balancing is shown in Figure 5.

```
Sebelum SMOTE: Counter({1: 44757, 0: 17873})
Setelah SMOTE: Counter({1: 35825, 0: 35825})
```

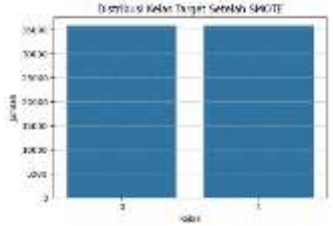

Fig 6. Data Distribution Before and After SMOTE

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 10, Issue 1, Januari 2026
DOI : https://doi.org/10.33395/sinkron.v10i1.15546
e-ISSN : 2541-2019
p-ISSN : 2541-044X

Model Training and Learning Curves
Model training was monitored using learning curves for the four algorithms. Figures 7–10 show the training and validation accuracy as a function of the training set size for:

Random Forest
The training accuracy gradually increased and approached 100% as the training set size expanded. The validation accuracy also increased consistently.

Fig 7. Random Forest Accuracy Results

KNN
Training accuracy decreased as the training data increased, while the validation accuracy increased.

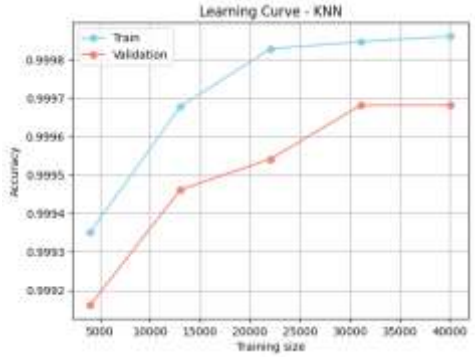

Fig 8. KNN Accuracy Results

Decision Tree
Training accuracy rapidly increased and reached nearly 100% at early training stages. Validation accuracy also increased as the training set size grew.

Fig 9. Decision Tree Accuracy Results

Gradient Boosting
Training and validation accuracy both increased and remained consistently high across all training sizes.

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 10, Issue 1, Januari 2026
DOI : https://doi.org/10.33395/sinkron.v10i1.15546
e-ISSN : 2541-2019
p-ISSN : 2541-044X

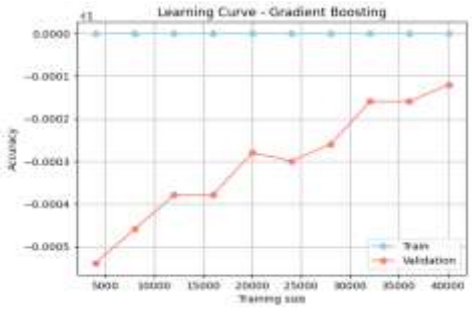

Fig 10. GradientBoosting Accuracy Results

After the training process, model performance was summarized in terms of accuracy and execution time. Results are shown in Table 3.

Table 3. Comparison of Model Accuracy and Execution Time

MODEL	ACCURACY	ESTIMATION (s)
RandomForest	97.22%	1.4338
KNN	98.33%	0.0327
DecisionTree	84.17%	0.0406
GradientBoost	96.94%	8.3819

The differences in accuracy among the models are minimal; however, execution time results show significant variation.

Model Evaluation

After model training, the performance of each model was evaluated using a confusion matrix. The confusion matrices for the four models are shown in Figure 11.

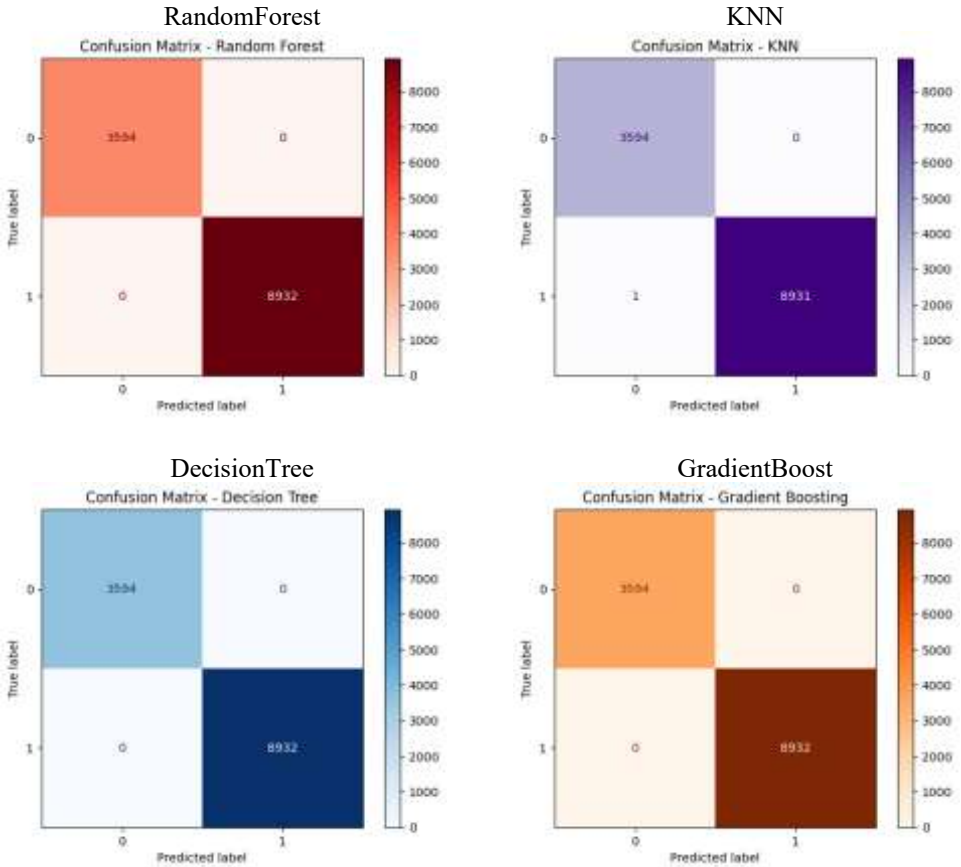

Fig 11. Evaluation Confusion Matrix for Four Models

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 10, Issue 1, Januari 2026
DOI : https://doi.org/10.33395/sinkron.v10i1.15546
e-ISSN : 2541-2019
p-ISSN : 2541-044X

Random Forest achieved an accuracy of 97.22% with an execution time of 1.4338 seconds. Based on the confusion matrix, the model correctly classified 3,584 samples as class 0 and 8,932 as class 1.

KNN achieved an accuracy of 98.33%with an execution time of 0.0327 seconds. The confusion matrix shows 3,594 correctly classified samples for class 0 and 8,931 for class 1.

Decision Tree achieved an accuracy of 84.17% and an execution time of 0.0406 seconds. The model correctly classified 3,594 samples as class 0 and 8,932 as class 1.

Gradient Boosting achieved an accuracy of 96.94% with an execution time of 8.3819 seconds. The model correctly classified 3,594 samples as class 0 and 8,932 as class 1.

The summary of Precision, Recall, and F1-Score is shown in Table 4

Table 4. Evaluation Results Based on Precision, Recall, and F1-Score

MODEL	PRECISION	RECALL	F1-SCORE
RandomForest	1.00	1.00	1.00
KNN	0.99	0.99	0.99
DecisionTree	1.00	1.00	1.00
GradientBoost	1.00	1.00	1.00

Although the performance scores are nearly identical across all models, the execution time reveals substantial differences, indicating trade-offs between predictive accuracy and computational efficiency. These findings are further interpreted in the Discussion section.

DISCUSSIONS

The classification results indicate that all four algorithms achieve relatively high predictive accuracy on the sensor-based smoke detection task, with accuracy values ranging from 84.17% to 98.33%. KNN attains the highest accuracy (98.33%), followed by Random Forest (97.22%) and Gradient Boosting (96.94%), while the Decision Tree shows the lowest accuracy (84.17%). When these results are interpreted in light of model complexity and computational cost, apparent differences emerge in terms of efficiency and scalability, which are critical for real-time IoT deployments where latency and resource constraints are as crucial as raw accuracy. In terms of execution time, KNN achieves the lowest runtime (0.0327 s), followed by Decision Tree (0.0406 s), Random Forest (1.4338 s), and Gradient Boosting (8.3819 s), reflecting the increasing computational burden of more complex ensemble methods.

The relatively low execution time of the Decision Tree is consistent with its hierarchical, node-based structure, in which prediction traverses only a single path from root to leaf. In theory, prediction cost grows approximately with tree depth rather than with the size of the training set, making tree-based models attractive for fast decision-making in embedded systems (Deepa et al. 2022; Erkmen and Ayrancı 2024). However, the observed accuracy of 84.17% indicates weaker generalization than KNN and ensemble models, reflecting the well-known limitation of unpruned trees in handling complex feature interactions and nonlinear decision boundaries. Gradient Boosting, by contrast, offers high predictive performance (96.94%) at the cost of the longest execution time (8.3819 s). This is expected because boosting sequentially constructs an ensemble of weak learners, where each new tree is trained to correct residual errors from previous trees, increasing both training and inference costs as the number of boosting stages grows. Similar accuracy–efficiency trade-offs have been reported in previous studies (Talukder et al. 2024), which generally recommend boosting-based models for offline analysis or batch prediction rather than strict real-time settings. Random Forest also shows strong performance (97.22%), benefiting from reduced variance through bagging and the aggregation of multiple decorrelated trees, consistent with robustness findings for environmental sensor data (Fulazzaky et al. 2024). Nevertheless, the execution time of 1.4338 s suggests that this robustness comes with a higher computational cost, which may limit its applicability on ultra-constrained IoT devices.

KNN presents a distinct behavior profile. In this study, it not only achieves the highest accuracy but also the lowest execution time, making it highly attractive for the current dataset and problem scale. However, as an instance-based method, its prediction cost scales with the size and dimensionality of the training set, since each new instance must be compared to many stored samples. Prior work (Vasconcelos et al. 2024) has shown that KNN becomes less practical for continuous, high-frequency monitoring on large-scale data. Thus, while KNN is highly effective under the conditions of this experiment, its suitability for extensive deployments or devices with strict memory constraints may be limited.

The use of SMOTE during preprocessing successfully mitigated class imbalance, yielding balanced classes and consistently high precision, recall, and F1-scores across all models, in line with the findings of (Handoko and Aditya 2025) for hazard-detection tasks. However, because the dataset is simulated rather than derived from long-term real-world deployments, it may not fully capture environmental variability, sensor drift, or hardware heterogeneity. Under non-stationary conditions, SMOTE-generated synthetic samples may not represent future

*name of corresponding author

distributions accurately, potentially leading to performance degradation over time. From an IoT implementation perspective, the observed trade-offs suggest that KNN is a strong candidate for latency-critical early warning systems at the current problem scale. At the same time, Decision Trees remain attractive for ultra-low-resource settings due to their simplicity and interpretability. Ensemble models such as Random Forest and Gradient Boosting are better suited for edge nodes with sufficient resources or cloud-based backends. These findings support the broader literature (Talukder et al. 2024; Vasconcelos et al. 2024), which indicates that lightweight models are generally preferable for real-time deployments, whereas heavier ensembles are more appropriate for backend analytics and periodic model refinement, potentially within hybrid edge–cloud architectures.

CONCLUSION

This study presents a comparative evaluation of four supervised machine learning algorithms, K-Nearest Neighbors (KNN), Decision Tree, Random Forest, and Gradient Boosting, for smoke detection using environmental sensor data. Scientifically, the main contribution lies in an integrated framework that combines SMOTE-based class imbalance handling, evaluation of classification accuracy, and explicit analysis of computational performance, allowing a balanced assessment between predictive quality and efficiency. All models achieved relatively high accuracy, with KNN obtaining the highest value (98.33%) and Decision Tree the lowest (84.17%). From a practical perspective, KNN demonstrated both the highest accuracy and the lowest execution time (0.0327 s), indicating its strong potential for real-time IoT and edge-based smoke detection systems where rapid response is essential. The Decision Tree, while less accurate, still offered low execution time (0.0406 s) and a simple, interpretable structure, which may be advantageous for ultra-low-resource devices. In contrast, Gradient Boosting, despite its competitive accuracy (96.94%), required a substantially longer execution time (8.3819 s), which may limit its use on resource-constrained devices. Future research should investigate lightweight deep learning or hybrid models that combine the speed of shallow classifiers with the adaptability of more complex architectures. Further work is also recommended to improve the real-time processing of streaming sensor data and to enable tighter integration with edge computing platforms, thereby enhancing scalability, robustness, and responsiveness in intelligent environmental monitoring and early warning applications.

REFERENCES

Alharbi, Fayez, Lahcen Ouarbya, and Jamie A. Ward. 2022. “Comparing Sampling Strategies for Tackling Imbalanced Data in Human Activity Recognition.” *Sensors* 22(4):1–20. doi: 10.3390/s22041373.

Bhamra, Jaspreet Kaur, Shreyas Anantha Ramaprasad, Siddhant Baldota, Shane Luna, Eugene Zen, Ravi Ramachandra, Harrison Kim, Chris Schmidt, Chris Arends, Jessica Block, Ismael Perez, Daniel Crawl, Ilkay Altintas, Garrison W. Cottrell, and Mai H. Nguyen. 2023. “Multimodal Wildland Fire Smoke Detection.” *Remote Sensing* 15(11). doi: 10.3390/rs15112790.

Carletti, Vincenzo, Antonio Greco, Alessia Saggese, and Bruno Vento. 2024. “A Smart Visual Sensor for Smoke Detection Based on Deep Neural Networks.” *Sensors* 24(14):1–17. doi: 10.3390/s24144519.

Deepa, K. R., A. S. Chaitra, K. Jhansi, R. D. Anitha Kumari, P. Ashwini Kumari, and Mallikarjun M. Kodabagi. 2022. “Development of Fire Detection Surveillance Using Machine Learning & IoT.” *MysuruCon 2022 - 2022 IEEE 2nd Mysore Sub Section International Conference* (February). doi: 10.1109/MysuruCon55714.2022.9972725.

Erkmen, Burcu, and Ahmet Aytuğ Ayrancı. 2024. “IoT-Based Fire Detection: A Comparative Study of Machine Learning Techniques.” *Ömer Halisdemir Üniversitesi Mühendislik Bilimleri Dergisi* 13(4):1298–1307. doi: 10.28948/ngumuh.1444349.

Fulazzaky, Tahira, Asep Saefuddin, and Agus Mohamad Soleh. 2024. “Evaluating Ensemble Learning Techniques for Class Imbalance in Machine Learning: A Comparative Analysis of Balanced Random Forest, SMOTE-RF, SMOTEBoost, and RUSBoost.” *Scientific Journal of Informatics* 11(4):969–80. doi: 10.15294/sji.v11i4.15937.

Hairani, Hairani, Khurniawan Eko Saputro, and Sofiansyah Fadli. 2020. “K-Means-SMOTE for Handling Class Imbalance in the Classification of Diabetes with C4.5, SVM, and Naive Bayes.” *Jurnal Teknologi Dan Sistem Komputer* 8(2):89–93. doi: 10.14710/jtsiskom.8.2.2020.89-93.

Handoko, Chanavaro Bayu, and Christian Sri Kusuma Aditya. 2025. “Penerapan Teknik SMOTE Dalam Mengatasi Imbalance Data Penyakit Diabetes Menggunakan Algoritma ANN.” *Smart Comp: Jurnalnya Orang Pintar Komputer* 14(1):13–20. doi: 10.30591/smartcomp.v14i1.7045.

He, Luhao, Yongzhang Zhou, Lei Liu, Yuqing Zhang, and Jianhua Ma. 2025. “Research and Application of Deep Learning Object Detection Methods for Forest Fire Smoke Recognition.” *Scientific Reports* 15(1):1–20. doi: 10.1038/s41598-025-98086-w.

Jamal, Muhammad Hassan, Abdulwahab Alazeb, Shahid Allah Bakhsh, Wadii Boulila, Syed Aziz Shah, Aizaz Ahmad Khattak, and Muhammad Shahbaz Khan. 2025. “Optimizing Fire Safety: Reducing False Alarms

*name of corresponding author

Using Advanced Machine Learning Techniques." *ArXiv Preprint ArXiv:2503.09960* 1–12.

Julian, James, Annastya Bagas Dewantara, and Fitri Wahyuni. 2024. "Design of Smoke Detection System Using Deep Learning and Sensor Fusion with Recursive Feature Elimination Cross-Validation." *IAES International Journal of Artificial Intelligence* 13(2):1658–67. doi: 10.11591/ijai.v13.i2.pp1658-1667.

Li, Xihao, Gui Zhang, Sanqing Tan, Zhigao Yang, and Xin Wu. 2023. "Forest Fire Smoke Detection Research Based on the Random Forest Algorithm and Sub-Pixel Mapping Method." *Forests* 14(3). doi: 10.3390/f14030485.

Liu, Jixue, Jiuyong Li, Stefan Peters, and Liang Zhao. 2024. "A Transformer Boosted UNet for Smoke Segmentation in Complex Backgrounds in Multispectral LandSat Imagery." *Remote Sensing Applications: Society and Environment* 36:1–17. doi: 10.1016/j.rsase.2024.101283.

Rajoli, Hossein, Sahand Khoshdel, Fatemeh Afghah, and Xiaolong Ma. 2024. "FlameFinder: Illuminating Obscured Fire Through Smoke With Attentive Deep Metric Learning." *IEEE Transactions on Geoscience and Remote Sensing* 62(Dml):1–11. doi: 10.1109/TGRS.2024.3440880.

Syukron, Muhamad, Rukun Santoso, and Tatik Widiharih. 2020. "Perbandingan Metode Smote Random Forest Dan Smote Xgboost Untuk Klasifikasi Tingkat Penyakit Hepatitis C Pada Imbalance Class Data." *Jurnal Gaussian* 9(3):227–36. doi: 10.14710/j.gauss.v9i3.28915.

Talukder, Md Alamin, Selina Sharmin, Md Ashraf Uddin, Md Manowarul Islam, and Sunil Aryal. 2024. "MLSTL-WSN: Machine Learning-Based Intrusion Detection Using SMOTETomek in WSNs." *International Journal of Information Security* 23(3):2139–58. doi: 10.1007/s10207-024-00833-z.

Tao, Wu, Fan Honghui, Zhu HongJin, You CongZhe, Zhou HongYan, and Huang XianZhen. 2021. "Intrusion Detection System Combined Enhanced Random Forest With Smote Algorithm." *EURASIP Journal on Advances in Signal Processing* 1–30.

Vasconcelos, Rodrigo N., Washington J. S. Franca Rocha, Diego P. Costa, Soltan G. Duverger, Mariana M. M. d. Santana, Elaine C. B. Cambui, Jefferson Ferreira-Ferreira, Mariana Oliveira, Leonardo da Silva Barbosa, and Carlos Leandro Cordeiro. 2024. "Fire Detection with Deep Learning: A Comprehensive Review." *Land* 13(10). doi: 10.3390/land13101696.

Wang, Ming, Liangcun Jiang, Peng Yue, Dayu Yu, and Tianyu Tuo. 2023. "FASDD: An Open-Access 100,000-Level Flame and Smoke Detection Dataset for Deep Learning in Fire Detection." *Earth System Science Data Discussions* 00103(November):1–26.

Zhang, Ziyang, Lingye Tan, and Tiong Lee Kong Robert. 2024. "An Improved Fire and Smoke Detection Method Based on YOLOv8n for Smart Factories." *Sensors* 24(15). doi: 10.3390/s24154786.

*name of corresponding author