






# Multi-sensor Data Fusion Algorithm for Indoor Fire Detection Based on Ensemble Learning

Lei Wang  and Jia Zhang  

Beijing Institute of Technology, Beijing, China  
zhangjia@bit.edu.cn

**Abstract.** Among all the disasters, fire is one of the most catastrophic events that frequently and universally threaten public safety and social development. In recent years, indoor fires have resulted in an increasing number of casualties and extensive damages. Therefore, in order to mitigate the impact of indoor fire disasters, it is crucial to propose an indoor fire detection method that can quickly and accurately detect the fire. In the research, a multi-sensor data fusion algorithm based on LogitBoost ensemble learning was designed for indoor fire detection. To improve detection accuracy and robustness, the proposed model utilizes an S-G filter and Min-Max normalization method, then uses Logitboost to synergistically integrate four classifiers, including Naïve Bayes, backpropagation neural network (BPNN), support vector machine (SVM) and k-nearest neighbor (KNN) classifier. A dataset containing various fire scenarios from the National Institute of Standards and Technology (NIST) was adopted to evaluate the effectiveness of the proposed algorithm. The experimental results demonstrated that the proposed method outperforms single models in terms of accuracy, stability, and efficiency.

**Keywords:** Fire Detection · Ensemble Learning · Multi-sensor Fusion

## 1 Introduction

In recent years, the continuous advancement of science and technology has led to increasingly complex modern building constructions, which concurrently has resulted in greater fire safety hazards. According to a report from the Department of Emergency Management Fire and Rescue of China on April 3, 2022, a total of 219,000 fires were reported across the country in the first quarter of the year, with residential areas being the most prominent location for fires, accounting for 38% of the total number of fires and 80.5% of the total number of fatalities [1]. It is evident that indoor fire accidents pose a significant threat to people's lives and property safety. Furthermore, minimizing the response time of fire detection systems is a crucial factor in reducing fire losses, as it increases the chances of extinguishing a fire [2]. Therefore, exploring a rapid response and high accuracy fire detection system is of paramount importance.

Currently, there are mainly three types of fire detection methods: traditional single sensor algorithms, multi-sensor fusion algorithms, and vision-based algorithms.

The traditional single sensor algorithm detects temperature, smoke, sound, light, and other parameters monolithically. This method employs a single source of information from the fire scene for threshold judgment, which is highly susceptible to external environmental interference, resulting in missing or false alarms. Furthermore, the alarm signal is generated late, making fire rescue more difficult and increasing the risk of fire damage.

The multi-sensor fusion algorithm incorporates data fusion theory into the fire detection system. This method comprehensively analyzes and intelligently processes the multi-source information collected by different sensors, which effectively compensates for the uncertainty and one-sidedness caused by the traditional single sensor detection method. As a result, more reliable and faster detection results can be obtained. It has become one of research hotspots in the fire detection field. Multi-sensor fusion algorithm fuses temperature, smoke, combustible gases, and other parameters. The main algorithms include Bayesian network, neural network algorithm, support vector machine (SVM), fuzzy algorithm, etc. Nakip proposed a trend prediction neural network (TPNN) model [3], and on this basis, proposed a recursive trend prediction neural network (rTPNN) model [4] which captures the trend of time series data from multi-sensors through trend prediction and horizontal prediction. Anand proposed a multi-sensor information fusion algorithm based on a feedforward neural network (FNN) for the delay-free prediction of forest fire storms [5]. Wen proposed a self-organizing T-S fuzzy neural network infrared flame detection algorithm, which overcomes local minimums by adjusting parameters through gradient descent. And consequentially improves the robustness of the model effectively [6]. Zheng combined BP neural network and D-S evidence theory in an underground fire scenario [7]. Ren proposed a fault identification method based on multi-information fusion for the fault arc problem of the electrical fire in low-voltage distribution system[8]. Although the algorithms in the above literatures have respective advantages in fire detection, most of the models are trained and tested using laboratory data and do not fully consider the temporal dimension information of sensor data. In the early stage of the fire, the sensor data shows a steady upward or downward trend in the long term but shows a random opposite trends or even irregular fluctuations in the short term.

The vision-based algorithm mainly detects smoke and flame images and processes them using neural networks, SVM, Markov model, expert systems, etc. Smoke image detection is based on features such as irregular motion and grayscale of smoke, while flame image detection relies on flame color, flame shape, and dynamic characteristics. Premal proposed a forest fire detection algorithm using a rules-based color model to classify fire pixels [9]. Mahmoud used the SVM method to divide the fire area, effectively improving the accuracy of the algorithm [10]. Pritam proposed a fire detection system based on LUV colour space and hybrid transforms [11]. Saima used the Grad-CAM method to visualize and locate flames in images, and built a CNN network based on an attention mechanism to detect fires[12]. Li combined multi-scale feature extraction, implicit depth supervision, and channel attention mechanisms to construct a fire detection system that achieved a better balance between accuracy, model size and speed[13]. Mukhriddin used a deep CNN model and an improved YOLOv4 object detector to identify smoke and flames[14]. However, in actual indoor fire applications, surveillance

cameras may have blind areas due to furniture occlusion, resulting in missed fire reports. Moreover, vision-based fire detection algorithms generally do not detect combustible gases and are slow to detect smoldering fires.

In summary, this paper proposed a multi-sensor data fusion algorithm based on a LogitBoost ensemble learning model that uses time-dimensional multi-sensor data from real fire scenes to solve the problem of false alarms caused by irregular fluctuations of fire data in a short period. This method effectively improves the accuracy of fire detection.

## 2 Data Selection and Analysis

### 2.1 Data Selection

The open-source fire data from NIST Test FR 4016 [15] was adopted in this work to test the effectiveness of the proposed algorithm. The dataset includes 27 experiments (SDC01-SDC15, SDC30-SDC41) conducted in a manufactured home to investigate the performance of different sensors in fire detection.

**Table 1.** Details of training set.

| Scenario | Fire Type  | Fire Location | Fire Material | Ignition Time(s) | No-Fire Data Count | Fire Data Count |
|----------|------------|---------------|---------------|------------------|--------------------|-----------------|
| SDC02    | Flaming    | Living area   | Chair         | 84               | 366                | 142             |
| SDC04    | Smoldering | Bedroom       | Mattress      | 8                | 149                | 797             |
| SDC05    | Flaming    | Bedroom       | Mattress      | 87               | 166                | 56              |
| SDC08    | Smoldering | Bedroom       | Mattress      | 131              | 253                | 1886            |
| SDC11    | Smoldering | Living area   | Chair         | 53               | 361                | 2264            |
| SDC12    | Flaming    | Kitchen       | Cooking oil   | 233              | 544                | 710             |
| SDC13    | Flaming    | Kitchen       | Cooking oil   | 100              | 390                | 880             |
| SDC31    | Smoldering | Living area   | Chair         | 4931             | 3338               | 1660            |
| SDC34    | Smoldering | Living area   | Chair         | 218              | 534                | 1852            |
| SDC35    | Flaming    | Living area   | Chair         | 94               | 546                | 75              |
| SDC36    | Flaming    | Bedroom       | Mattress      | 32               | 1794               | 974             |
| SDC37    | Smoldering | Bedroom       | Mattress      | 130              | 485                | 926             |
| SDC38    | Flaming    | Bedroom       | Mattress      | 35               | 787                | 568             |
| SDC39    | Flaming    | Bedroom       | Mattress      | 22               | 357                | 55              |
| SDC40    | Smoldering | Bedroom       | Mattress      | 522              | 665                | 1425            |
| SDC41    | Flaming    | Kitchen       | Cooking oil   | 110              | 310                | 972             |

Since SDC03, SDC30, and SDC32 were aborted due to ignition failure, the remaining 24 experiments were selected as dataset to investigate the performance of different

**Table 2.** Details of test set.

| Scenario | Fire Type  | Fire Location | Fire Material | Ignition Time(s) | No-Fire Data Count | Fire Data Count |
|----------|------------|---------------|---------------|------------------|--------------------|-----------------|
| SDC01    | Smoldering | Living area   | Chair         | 377              | 100                | 1355            |
| SDC06    | Smoldering | Bedroom       | Mattress      | 83               | 151                | 1179            |
| SDC07    | Flaming    | Bedroom       | Mattress      | 59               | 392                | 169             |
| SDC09    | Flaming    | Bedroom       | Mattress      | 31               | 424                | 536             |
| SDC10    | Flaming    | Living area   | Chair         | 100              | 566                | 167             |
| SDC14    | Flaming    | Bedroom       | Mattress      | 3398             | 2407               | 407             |
| SDC15    | Flaming    | Living area   | Chair         | 271              | 441                | 106             |
| SDC33    | Flaming    | Living area   | Chair         | 88               | 1352               | 72              |

sensors in fire detection. The experiment recorded in detail the entire process data of various kinds of sensor data including temperature, smoke, CO, CO<sub>2</sub>, and O<sub>2</sub> concentration. In this work, temperature, CO concentration, and smoke concentration is used as input feature parameters.

The training set contains 16 groups of experimental data and test set contains 8 groups of experimental data. Details of training set and test set are shown in Tables 1 and 2. As is shown in the table, the training set includes 11405 no-fire data count and 15242 fire data count, and the test set includes 5833 no-fire data count and 3391 fire data count. Among all the tests, SDC09, SDC14, and SDC36 were conducted with the bedroom door closed, while the rest were open.

## 2.2 Data Processing and Analysis

In various combustion states, different fire detection parameters exhibit distinct characteristics. Therefore, selecting appropriate fire characteristic parameters is essential for accurately and efficiently identifying the fire state. Moreover, since real experiments are subject to external interference, the obtained data exhibit random and irregular fluctuations, so it is necessary to preprocess the data before the subsequent model training and testing.

**Select Input Feature Parameters.** In this work, the following three fire characteristic parameters are used as the identification basis.

*Temperature.* One of the easiest indicators to measure and can be used as an important characteristic parameter of fire. Expressed as  $x = (x_1, x_2, \dots, x_{24})^T$ , where  $x_i = (x_{i1}, x_{i2}, \dots, x_{ip})^T$ ,  $i = 1, 2, \dots, 24$ , the unit is Celsius and  $p$  is the data length, as shown in the sixth column of Tables 1 and 2.

*CO Concentration.* Increases rapidly when a fire occurs, especially when smoldering, easy to float to the roof due to low density, and easy to be detected. Expressed as  $y = (y_1, y_2, \dots, y_{24})^T$ , where  $y_i = (y_{i1}, y_{i2}, \dots, y_{ip})^T$ ,  $i = 1, 2, \dots, 24$ , the unit is volume fraction  $\times 10^6$ .

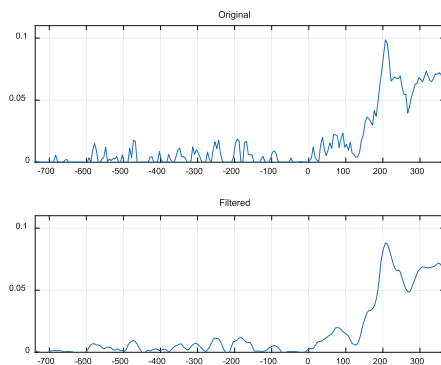
*Smoke Concentration.* One of the most obvious phenomena in early fire, is of great significance to describe the state of fire. Expressed as  $z = (z_1, z_2, \dots, z_{24})^T$ , where  $z_i = (z_{i1}, z_{i2}, \dots, z_{ip})^T$ ,  $i = 1, 2, \dots, 24$ , the unit is %/ft.

### Data Processing

*Filtering.* The S-G (Savitzky-Golay) filtering method is adopted to filter the temperature, CO concentration and smoke concentration. It is based on polynomial least square fitting in the time domain which can preserve the shape and width of the signal while filtering noise. The smoothing formula of S-G filtering method is shown in  $\bar{y}_i = \sum_{n=-M}^{+M} y_{i+n} h_n$ ,  $i = 1, 2, \dots, 24$  (1). Taking CO concentration as an example, for a set of data within the window  $y_i[n]$ ,  $n = -M, \dots, 0, \dots, M$ , The value of n is  $2M + 1$  consecutive integer values, M is the half width of the approximate interval,  $M = 5$ , frame length is 11, and a third-order fitting polynomial  $f(n) = \sum_{k=0}^3 b_{nk} n^k = b_{n0} + b_{n1}n + b_{n2}n^2 + b_{n3}n^3 = h_n y_i[n]$  is constructed to fit the data, the smoothing coefficient is  $h_n = \underset{h_n}{\operatorname{argmin}} \sum_{n=-M}^M (f(n) - y_i[n])^2$  and the filtered data is  $\bar{y}_i$ . Similarly for temperature and smoke concentration filtering, the S-G-filtered dataset is expressed as  $[\bar{x}_i \ \bar{y}_i \ \bar{z}_i]$ .

$$\bar{y}_i = \sum_{n=-M}^{+M} y_{i+n} h_n, i = 1, 2, \dots, 24 \quad (1)$$

The comparison of CO concentration data before and after filtering is shown in Fig. 1. It can be seen from the figure that the zigzag fluctuations of the filtered curve is reduced and smoothed obviously, indicating the S-G filter can effectively reduce data noise.



**Fig. 1.** Comparison of CO concentration data before and after filtering

*Normalization.* To mitigate the influence of larger attribute values (such as temperature) on smaller values (such as CO concentration) and eliminate the adverse effects caused by singular sample data, all characteristic parameter values are normalized using  $[\tilde{x}_i \ \tilde{y}_i \ \tilde{z}_i] = \left[ \frac{\bar{x}_i - \bar{x}_{i\min}}{\bar{x}_{i\max} - \bar{x}_{i\min}} \quad \frac{\bar{y}_i - \bar{y}_{i\min}}{\bar{y}_{i\max} - \bar{y}_{i\min}} \quad \frac{\bar{z}_i - \bar{z}_{i\min}}{\bar{z}_{i\max} - \bar{z}_{i\min}} \right]$ ,  $i = 1, 2, \dots, 24$  (2) and mapped to the

interval (0,1).  $[\tilde{x}_i \tilde{y}_i \tilde{z}_i]$  is the feature value after normalization.

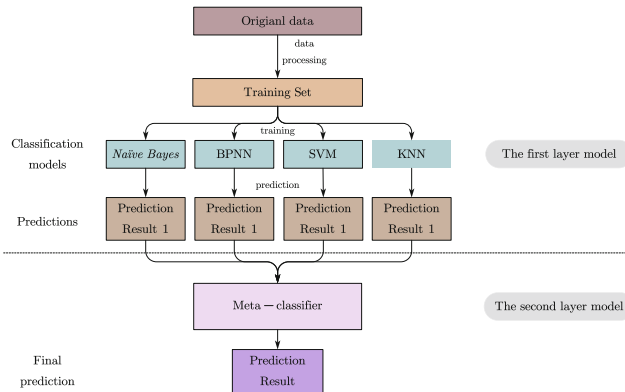
$$[\tilde{x}_i \tilde{y}_i \tilde{z}_i] = \left[ \frac{\bar{x}_i - \bar{x}_{i\min}}{\bar{x}_{i\max} - \bar{x}_{i\min}} \quad \frac{\bar{y}_i - \bar{y}_{i\min}}{\bar{y}_{i\max} - \bar{y}_{i\min}} \quad \frac{\bar{z}_i - \bar{z}_{i\min}}{\bar{z}_{i\max} - \bar{z}_{i\min}} \right], i = 1, 2, \dots, 24 \quad (2)$$

The data obtained through S-G filtering and Min-Max normalization composed the fire dataset in this study. Temperature, smoke concentration, CO concentration, and label are considered the sample of data for training and testing.

### 3 Algorithm Analysis and Evaluation

#### 3.1 Architecture of Algorithm

Ensemble learning is a machine learning method that involves training a series of basic models and processing the output result of each model through the ensemble principle [16, 17]. There are three common types of ensemble learning: bagging, boosting, and stacking. Bagging and boosting are homogeneous ensemble models, which often involve standalone weak learners of the same type. Stacking is a heterogeneous ensemble model, which typically employs standalone learners of different types. The stacking ensemble principle is to combine multiple models in different layers, and iteratively learns the classification deviation of the previous model [18]. Stacking ensemble algorithm can integrate different types of models and classification features, leading to better performances.



**Fig. 2.** Architecture of proposed ensemble learning model.

The architecture of the proposed multi-sensor fusion algorithm based on ensemble learning is illustrated in Fig. 2. The stacking ensemble learning model in this work is designed as a two-layer structure. The first layer has four types of basic classifiers including Naïve Bayes, BPNN, SVM, and KNN models. All of the models were trained on the training set to generate the classification results of each model. The second layer is the LogitBoost ensemble learning model. It combines the four classification results of the

first layer as new feature parameters, and uses the newly constructed feature parameters and labels to train the ensemble model with the same initial weight value. Eventually outputs the final classification result.

### 3.2 Research Methodology

**The First Layer.** The first layer contains 4 base-classifiers, including Naive Bayes, BPNN, SVM and KNN classifiers. Temperature, smoke concentration and CO concentration  $[\tilde{x}_i \tilde{y}_i \tilde{z}_i]$ ,  $i = 1, \dots, 16$  were used as input features, and the fire state  $k_i \in \{1, 2\}$  was used as the label value, where 1 represented the state without fire and 2 represented the state with fire. Four basic classifier models were obtained by training the classifier using the training set respectively. The model was tested using 3227 sets of data from the test set  $[\tilde{x}_i \tilde{y}_i \tilde{z}_i]$ ,  $i = 17, \dots, 24$ , and outputted 4 sets of predictions  $o_{i,1}, o_{i,2}, o_{i,3}, o_{i,4}$ ,  $i = 1, \dots, 3391$ .

A brief description of the principle of each classifier model is as follows.

*Naïve Bayes Model.* Naïve Bayes constructs a classifier using the attribute condition independent assumption, and for a certain feature, the classification output probability can be expressed as (3), where  $k$  is the value of the fire state label,  $P(\tilde{x}|k_i)$ ,  $P(\tilde{y}|k_i)$ ,  $P(\tilde{z}|k_i)$  is conditional probabilities and  $P(k_i) = \frac{k_i}{k_1+k_2}$  is prior probability, both of which can be calculated from the sample data. The Naïve Bayes model outputs the prediction results  $o_{i,1} = [o_{1,1}, \dots, o_{i,1}, \dots, o_{3391,1}]^T$ ,  $i = 1, \dots, 3391$ .

$$P(k_i|\tilde{x}, \tilde{y}, \tilde{z}) = \frac{P(\tilde{x}|k_i)P(\tilde{y}|k_i)P(\tilde{z}|k_i)P(k_i)}{P(\tilde{x})P(\tilde{y})P(\tilde{z})}, i = 1, 2 \quad (3)$$

*BPNN Model.* The output of BPNN uses forward propagation and the errors use back propagation. In this work, the BPNN model has 2 input neurons, 2 hidden layers, and 1 output neuron. The first hidden layer has 10 neurons and the second hidden layer has 4 neurons. The BPNN model outputs the prediction results  $o_{i,2} = [o_{1,2}, \dots, o_{i,2}, \dots, o_{3391,2}]^T$ ,  $i = 1, \dots, 3391$ .

*SVM Model.* The SVM algorithm is one of the most robust classification and regression algorithms. The main objective of SVM algorithm in binary classification is to get the minimum hyperplanes that have maximum distance from the training data set. The SVM model outputs the prediction results  $o_{i,3} = [o_{1,3}, \dots, o_{i,3}, \dots, o_{3391,3}]^T$ ,  $i = 1, \dots, 3391$ .

*KNN Model.* The KNN is based on the training instance categories of  $k$  nearest neighbors and predictions are made through majority voting, etc. In this work, the value of  $k$  is 1. The KNN model outputs the prediction results  $o_{i,4} = [o_{1,4}, \dots, o_{i,4}, \dots, o_{3391,4}]^T$ ,  $i = 1, \dots, 3391$ .

**The Second Layer.** The second layer of the proposed algorithm is an ensemble learning layer, which combines the four basic classifiers of the first layer to construct an integrated fire detection model. The stacking ensemble method is utilized to combine the predictions from multiple well-performing machine learning models to make better predictions than any single model.

The ensemble learning algorithm used in this work is LogitBoost. It addresses the two shortcomings of AdaBoost, which respectively are sensitive to anomalies due to excessive punishment for misclassification points, and cannot predict the probability of a category. LogitBoost was employed as meta-learning model to detect fire, and a general framework of ensemble classifiers was constructed by combining five decision tree based classifiers. LogitBoost uses a logarithmic loss function that iterates Newton-Raphson at each step. The method is applied to the error function to update the model, and the logistic regression model is employed to combine the predictions of all the weak classifiers to get the final prediction result.

Let the prediction results of the four basic classifiers in the first layer be  $o_{3391 \times 4} =$

$$[o_{i,1}, o_{i,2}, o_{i,3}, o_{i,4}] = \begin{bmatrix} o_{1,1} & \cdots & o_{1,4} \\ \vdots & \ddots & \vdots \\ o_{3391,1} & \cdots & o_{3391,4} \end{bmatrix}, i = 1, \dots, 3391. \text{ Each column of the}$$

prediction result matrix is taken as a meta-features vector, and the fire state is taken as the label value. LogitBoost model is used to train the meta-features, and a model mapping from the meta-features to the ground-truth is obtained. The fire detection model is then trained using LogitBoost algorithm.

For training set  $D = \{(o_{1,j}, k_1), \dots, (o_{i,j}, k_i), \dots, (o_{3391,j}, k_{3391})\}$ ,  $i = 1, \dots, 3391, j = 1, 2, 3, 4$ , where  $k_i \in \{1, 2\}$  indicates the fire status label. 1 is no fire state, and 2 is fire state. Let the ensemble function be  $F(o_{i,j})$ , the probability of  $k_i = 1$  is  $p(k_i = 1|o_{i,j}) = \frac{e^{F(o_{i,j})}}{e^{F(o_{i,j})} + e^{-F(o_{i,j})}}$ , the logarithmic loss function is  $L(k_i, F(o_{i,j})) = -k_i \log(p(k_i = 1|o_{i,j})) - (1 - k_i) \log(p(k_i = 1|o_{i,j})) = \log(1 + e^{-2F(o_{i,j})})$ . The goal of the LogitBoost algorithm is to use the decision tree algorithm  $f(o_{i,j})$  as the basic learner to calculate  $F(o_{i,j}) = \arg \min_{F(o_{i,j})} L(k_i, F(o_{i,j}))$ , the specific steps are as follows:

- Step 1. For each set of data, set the initial weight  $w_i = \frac{1}{3391}$ , the initial value of the ensemble function  $F_0(o_{i,j}) = 0$ , and the initial probability of  $k_i = 1$  is  $p(k_i = 1|o_{i,j}) = \frac{1}{2}$ .
- Step 2. For iteration step  $m = 1, \dots, M$ , repeat:
  - 1) For  $i = 1, \dots, 3391$ , calculate working response value  $z_{m,i} = \frac{k_i - p_{m-1}(o_{i,j})}{p_{m-1}(o_{i,j})(1 - p_{m-1}(o_{i,j}))}$  and weight  $w_{m,i} = p_{m-1}(o_{i,j})(1 - p_{m-1}(o_{i,j}))$ ;
  - 2) Fit the function  $f_m(o_{i,j})$  by a weighted least-squares regression of  $z_{m,i}$  to  $o_{i,j}$  with weights  $w_{m,i}$  using the decision tree approach. The derivative of the loss function is  $\frac{\partial L(k_i, F_{m-1}(o_{i,j}))}{\partial F_{m-1}(o_{i,j})} = z_{m,i}$ ,  $\frac{\partial^2 L(k_i, F_{m-1}(o_{i,j}))}{\partial F_{m-1}^2(o_{i,j})} = w_{m,i}$ , thus  $f_m(o_{i,j}) = \arg \min_{f_m(o_{i,j})} L(k_i, F_{m-1}(o_{i,j}))$  can be obtained;
  - 3) For  $i = 1, \dots, 3391$ , update  $F_m(o_{i,j}) \leftarrow F_{m-1}(o_{i,j}) + \frac{1}{2} f_m(o_{i,j})$ ,  $p_m(k_i = 1|o_{i,j}) \leftarrow \frac{e^{F_m(o_{i,j})}}{e^{F_m(o_{i,j})} + e^{-F_m(o_{i,j})}}$ .
- Step 3. Output ensemble classifier  $\text{sign}[F(o_{i,j})] = \text{sign}\left[\sum_{m=1}^M f_m(o_{i,j})\right]$ .



Test the ensemble model using the meta-features of testing data and thus the final prediction results are obtained as  $[\tilde{o}_1, \tilde{o}_2, \dots, \tilde{o}_{3391}]^T$ .

### 3.3 Evaluation Index

By comparing the model predicted label with the true label value, the performances of the algorithms were evaluated on the basis of accuracy, precision, recall, F1 score, receiver operating characteristic (ROC) curve, and area under the ROC curve (AUC).

The confusion matrix is shown in Table 3, where TP and TN respectively denotes the number of correctly predicted positive values and negative values, by analogy FP and FN denote the prediction is false.

**Table 3.** Confusion matrix.

| True value | Predicted value |          |
|------------|-----------------|----------|
|            | Positive        | Negative |
| Positive   | TP              | FP       |
| Negative   | FN              | TN       |

The indexes are as (4)–(7).

$$Accuracy = \frac{TN + TP}{TN + FP + TP + FN} \quad (4)$$

$$Precision = \frac{TP}{TP + FP} \quad (5)$$

$$Recall = \frac{TP}{TP + FN} \quad (6)$$

$$F1score = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (7)$$

Accuracy is the percentage of the correctly predicted samples in the total samples in the whole test set, which intuitively reflects the overall classification ability of algorithms. Precision is the percentage of the correctly predicted positive samples to all actual positive samples, which can reflect the ability to distinguish negative samples. Recall is the percentage of correctly predicted positive samples to all predicted positive samples, which can represent the ability to identify positive samples. F1 score is the harmonic average of precision and recall, which can characterize the comprehensive performance measurement of algorithms in precision and recall. The value of the preceding indicators ranges from 0 to 1. The closer the indicator is to 1, the more robust the model is.

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

$$FPR = \frac{FP}{FP + TN} \tag{9}$$

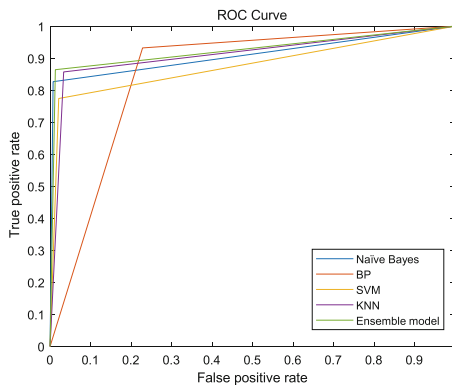
The vertical axis of the ROC curve is true positive rate (TPR), which represents the proportion of the correctly predicted positive samples to all predicted positive samples, as shown in (8). The horizontal axis of the ROC curve is false positive rate (FPR), which stands for the proportion of incorrectly predicted positive samples to all predicted negative samples, as shown in (9). The ROC curve, which is not affected by the imbalance of samples, can embody the classification performance of algorithms at each sample point. The closer the curve is to the top left, the better the performance is. AUC is the area under the ROC curve, which can more accurately reflect the overall classification ability of the algorithms in the form of detailed figures.

### 3.4 Experimental Results

The performance and ROC curve of each algorithm in this work is respectively shown in Table 4 and Fig. 3.

**Table 4.** The performance of each algorithm.

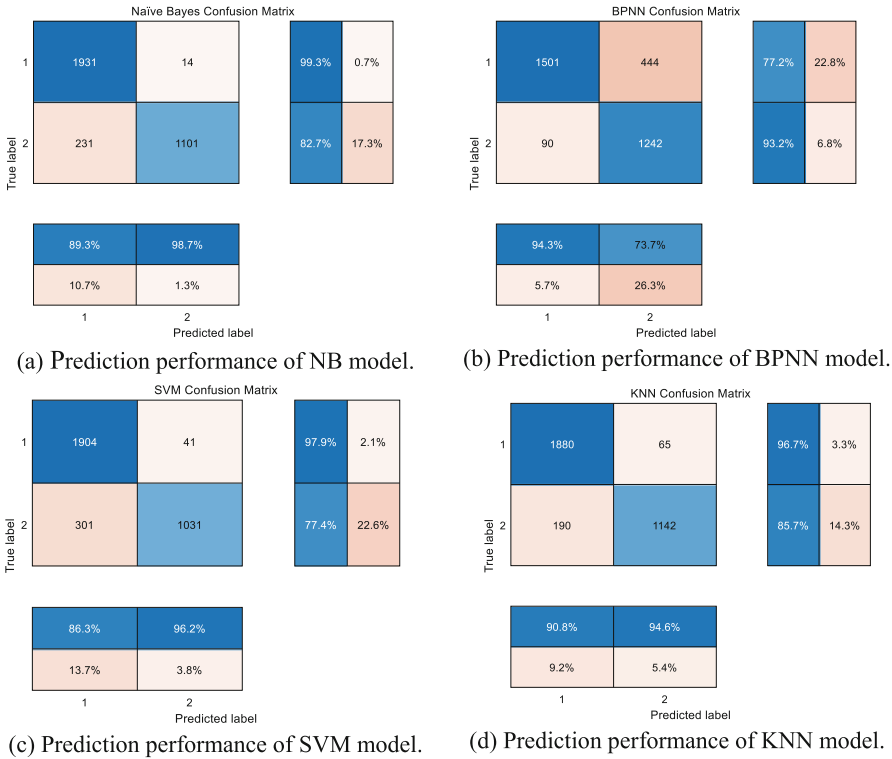
|                | Accuracy | Precision | Recall | F1 score | AUC    |
|----------------|----------|-----------|--------|----------|--------|
| Naïve Bayes    | 0.9252   | 0.8932    | 0.9928 | 0.9403   | 0.9103 |
| BPNN           | 0.8755   | 0.8943    | 0.8961 | 0.8952   | 0.8759 |
| SVM            | 0.8955   | 0.8635    | 0.9789 | 0.9176   | 0.8864 |
| KNN            | 0.9222   | 0.9082    | 0.9666 | 0.9365   | 0.9112 |
| Ensemble model | 0.9371   | 0.9139    | 0.9871 | 0.9491   | 0.9592 |



**Fig. 3.** ROC curve of each algorithm.

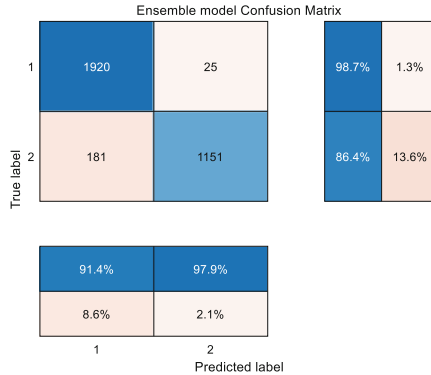
The prediction performance of the single models is shown in Fig. 4 and the prediction performance of ensemble model is as Fig. 5. The upper left is the confusion matrix, with

rows corresponding to real classes and columns corresponding to prediction classes. The upper right graph is the column-normalized column summary, showing the percentage of the number of observations that are correctly and incorrectly classified for each prediction category. The lower-left graph is the row-normalized row summary, showing the percentage of the number of observations that are correctly and incorrectly classified for each real category.



**Fig. 4.** Single model prediction performance.

It can be observed from Table 4 that the performance of Naïve Bayes is the best of the four basic classifiers. It has a high recall reaching 99.28%, however, the precision is only 89.32%, which means the false positive rate is high, leading to potential missing alarm rate. The BPNN model has the best recognition for on-fire state while the recognition accuracy for no-fire state is low, leading to higher false alarm rate. The proposed ensemble model well balanced the precision and recall while improving the multi-faceted performance including accuracy, precision, recall, F1 score, and AUC area. The ROC curve of ensemble model is closest to the upper left corner as Fig. 3 shows, indicating the overall classification effect is remarkable. Meanwhile, the performances of Naïve Bayes and KNN are highly similar, which are more excellent than the BPNN and SVM. The value of the AUC accurately quantifies the above analysis. AUC of ensemble model is 0.9592, higher than the rest of the algorithms.



**Fig. 5.** Prediction performance of ensemble model.

Therefore, compared with other comparison algorithms, the proposed ensemble learning algorithm in this work improves the performance of fire detection in many aspects, including accuracy, precision, recall, F1 score and AUC area. It improves F1 score on the basis of balancing accuracy rate and recall rate, and has the best overall classification performance for the fire detection system.

## 4 Conclusion

Aiming at the problem of indoor fire detection with real-time multi-sensor data, an innovative multi-sensor data fusion algorithm based on LogitBoost ensemble learning model is proposed. Temperature, smoke concentration and CO concentration were selected as fire characteristic parameters, and four single classifiers named Naive Bayes, BPNN, SVM and KNN, as well as the proposed ensemble learning algorithm, were trained and tested using NIST data sets. The results show that the accuracy, precision, recall rate and AUC of proposed ensemble classifier are improved effectively, and the performance is better than that of any single classifier.

## References

1. Fang, X.: A total of 219 000 fire deaths and 625 deaths in the first quarter of China. *China Fire Control* **557**(04), 21 (2022)
2. Qureshi, S., Ekpanyapong, M., Dailey, N., et al.: QuickBlaze: Early fire detection using a combined video processing approach. *Fire Technol.* **52**(5), 1293–1317 (2016)
3. Nakip, M., Güzelış, C.: Multi-Sensor fire detector based on trend predictive neural network. In: 11th International Conference on Electrical and Electronics Engineering, pp. 600–604. Bursa, Turkey (2019)
4. Nakip, M., Güzelış, C., Yildiz, O.: Recurrent trend predictive neural network for multi-sensor fire detection. *IEEE Access* **9**, 84204–84216 (2021)
5. Anand, S, Keetha, K.: FPGA implementation of artificial neural network for forest fire detection in wireless Sensor Network. In: 2nd International Conference on Computing and Communications Technologies, pp. 265–270. Chennai, India (2017)

6. Ziteng, W., Linbo, X., Hongwei, F., Yong, T.: Infrared flame detection based on a self-organizing TS-type fuzzy neural network. *Neurocomputing* **337**, 67–79 (2019)
7. Zianyu, Z., Yi, L., et al.: Fire detection scheme in tunnels based on multi-source information fusion. In: 18th International Conference on Mobility, Sensing and Networking, pp. 1025–1030. Guangzhou, China (2022)
8. Ren, X., Li, C., et al.: Design of multi-information fusion based intelligent electrical fire detection system for green buildings. *Sustainability* **13**(6), 3405 (2021)
9. Premal, E., Vinsley, S.: Image processing based forest fire detection using YCbCr colour model. In: International Conference on Circuits, Power and Computing Technologies, pp. 1229–1237. Nagercoil, India (2014)
10. Mahmoud, M., Ren, H.: Forest fire detection and identification using image processing and SVM. *J. Inform. Process. Syst.* **15**(1), 159–168 (2019)
11. Pritam, D., Dewan, H.: Detection of fire using image processing techniques with LUV color space. In: 2nd International Conference for Convergence in Technology, pp. 1158–1162. Mumbai, India (2017)
12. Saima, M., Fayadh, A.: Attention based CNN model for fire detection and localization in real-world images. *Expert Syst. Appl.* **189**, 116114 (2022)
13. Songbin, L., Qiangdong, Y., Peng, L.: An efficient fire detection method based on multiscale feature extraction, implicit deep supervision and channel attention mechanism. *IEEE Trans. Image Process.* **29**, 8467–8475 (2020)
14. Mukhiddinov, M., Abdusalomov, B., Cho, J.: Automatic fire detection and notification system based on improved YOLOv4 for the blind and visually impaired. *Sensors* **22**(9), 3307 (2022)
15. Bukowski, W., Peacock, D., et al.: Performance of Home Smoke Alarms, Analysis of the Response of Several Available Technologies in Residential Fire Settings. Technical Note [https://tsapps.nist.gov/publication/get\\_pdf.cfm?pub\\_id=100900](https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=100900). Accessed 12 June 2023
16. Arya, N., Saha, S.: Multi-modal classification for human breast cancer prognosis prediction: proposal of deep-learning based stacked ensemble model. *IEEE/ACM Trans. Comput. Biol. Bioinf.* **19**(2), 1032–1041 (2022)
17. Jiang, J., Zhu, W.: Electrical load forecasting based on multi-model combination by stacking ensemble learning algorithm. In: IEEE International Conference on Artificial Intelligence and Computer Applications, pp. 739–743. Dalian, China (2021)
18. Smith, G.: Fire-detection and alarm systems. *Wiring Install. Suppl.* **3**, 9–11 (1977)