EAI Endorsed Transactions
on Transportation Systems and Ocean Engineering Research Article EAI.EU

Development of a Fire Risk Prediction System Based on Deep Learning

Minh-Vuong Nguyen[1,2] and Thanh-Duy Nguyen[1,*]

[1]Artificial Intelligent in Transportation, Ho Chi Minh City University of Transport, Vietnam
[2]Hanata Engineering Construction Company Limited, Vietnam

Abstract

Sensor-based safety monitoring systems play an important role in early detection and prevention of fire and explosion incidents at industrial pumping stations. Modern stations integrate multiple sensors such as temperature, humidity, gas, dust, air quality and fire sensors to assess operational status. In practice, continuous monitoring and coordinated analysis of multi-sensor data conducted by experienced professionals can significantly reduce the risk of fire and explosion. However, sustaining continuous expert-based monitoring is challenging due to high operational costs, manpower demands, and practical constraints. Consequently, automated monitoring and forecasting systems are required to deliver continuous and timely risk assessments while minimizing dependence on manual supervision. However, sensor signals often contain noise that is nonlinear and susceptible to environmental influences, making traditional threshold comparison methods unstable. This paper proposes a fire monitoring and forecasting system based on time series data and deep learning model with three status levels including safe, warning and dangerous. The models used and compared include decision trees, artificial neural networks, and long short-term memory networks with a twelve-step time window. Multi-sensor data are normalized and organized into time series windows to reduce noise and reflect fluctuations in real-world conditions. Experimental results on sensor data collected at the pumping station show that the long short-term memory network achieves higher accuracy and precision than the other two models. Contributing to improving the reliability of safety monitoring at the pumping station and creating a basis for practical implementation.

Keywords: fire alarm system, decision tree, artificial neural networks, long short-term memory network, internet of things.

Received on 07 December 2025, accepted on 15 December 2025, published on 14 January 2026

doi: 10.4108/_______________

1. Introduction

Fire and explosion are among the most dangerous incidents in the industrial environment, which can cause great damage to property, disrupt production and directly threaten human safety. In areas containing flammable fuels or gases, the risk is even higher due to the rapid and difficult-to-control spread, so monitoring and early warning systems play a key role in promptly detecting unusual signs before an incident occurs. In the early stages, fire alarm systems relied mainly on single sensor thresholds, such as modeling the relationship between hot air flow and the thermal response of a detector [1] or optical obscuration thresholds for smoke detectors [2-3]. The physical properties of smoke including particle size, optical density and aging process have been analyzed in [4], and even in large spaces, threshold-based binary classification method is still widely applied [5]. However, threshold-based models are susceptible to environmental background noise and increase the false alarm rate.

As the need to reduce false positives became more urgent, more traditional machine learning methods began

*Corresponding author. Email: nguyenthanhduy919@gmail.com

to be investigated. For example, Support Vector Machines (SVM) [6] allow the combination of multiple measurement sources including heat, smoke, toxic gases, and humidity to improve accuracy. High-dimensional sensor fusion [7], fuzzy inference [8], and evidence fusion [9] methods help smooth the response in the transition region. Reviews such as building research establishment (BRE) [10] and studies in [11] show that machine learning can significantly reduce false positives, although most systems still stop at binary classification. Decision tree models are exploited in [12] to describe risk variation with environmental conditions and in [13] as an explanatory tool to identify abnormal trends before thresholds are exceeded. The development of the Internet of Things opens up the possibility of continuous sensor data collection, thereby facilitating the application of deep learning models in time series analysis. Research [14] shows that long-short-term memory networks have long-term memory capabilities and limited gradient loss, thus suitable for describing temporal dependencies in fire signals. Extensive studies [15-16] also demonstrate that long short-term memory networks are more stable than traditional Recurrent Neural Network (RNN) variants when the environment contains a lot of noise. In Vietnam, most fire alarm systems are still based on thresholds [17], although TCVN 5738:2021 [18] standard has raised the requirements for detector performance. Recent studies using ESP32 and ESP8266 microcontroller platforms [19-20] have begun to build continuous sensor data warehouses for deep learning models. In parallel, decision tree models used in forest fire spatial risk classification [21-22], multi-layer feedforward models for nonlinear classification problems [23], autoencoders in unlabeled anomaly detection [24], and simple RNN variants in [25] also demonstrated the ability to predict trends before thresholds are exceeded.

In summary, although there have been many different approaches, most studies still focus on binary classification or have not fully exploited the time series nature of sensor data. This raises the need to develop a more stable multi-risk monitoring and forecasting system in noisy environments. On that basis, the paper focuses on developing a solution to monitor and forecast fire and explosion risks for fuel pumping stations, exploiting real-time sensor data combined with deep learning models to identify abnormal conditions early, thereby supporting timely warnings and minimizing risks in operations. The main contributions of the paper include:

(1) Building an experimental model of a pumping station and a multi-sensor Internet of Things system to collect real-time data.
(2) Propose a process for processing and organizing time series data, and develop decision tree, artificial neural network and short-term memory classification models for three risk levels.
(3) Organize training and experimental evaluation based on collected results with many criteria to clearly show the effectiveness of each model, and at the same time propose a model combination direction to improve the reliability of the system.

The structure of this paper is distributed as follows: Section 2 presents the methodology, including data collection and storage, the overview of the proposed model, the model evaluation criteria, and quantitative thresholding. Section 3 provides the algorithm of the training models; experimental and evaluation results are provided in section 4. Section 5 will give conclusions for the research.

2. Proposed method

Traditional fire alarm studies are mainly based on response models of heat and smoke detectors, in which thresholds are set based on optical obscuration, smoke density, or hot air flow propagation characteristics. Although suitable in simple environments, these threshold-based systems are susceptible to environmental noise, causing false alarm rates to increase significantly when background conditions change or smoke signals are weak and unevenly dispersed. The emergence of Internet of Things systems makes sensor data collection more continuous and richer, facilitating the application of machine learning methods such as SVM, decision trees, fuzzy inference, and evidence fusion. The development of deep learning, especially RNN and LSTM models, opens up the possibility of modeling long-term dependencies in noisy and complex data. However, most fire alarm systems are still threshold-based or employ simple machine learning models.

On that basis, the paper proposes a fire and explosion risk monitoring and forecasting system for fuel pumping stations based on real-time sensor data and deep learning models. The multi-sensor data mining system is organized in the form of time series and compares three methods DT, ANN and LSTM to classify risks into three levels including safety, warning and danger. This method aims to increase the accuracy and stability of the warning system under real operating conditions. The system is deployed on a miniature industrial pumping station model, including pump cluster, pipeline, fuel storage area and control cabinet in Figure 1. Sensors are strategically installed at critical leak-prone locations, such as control valve joints, pump heads, flexible joints, and pump room ceilings, to enable continuous acquisition of environmental parameters: temperature and humidity reflect the risk of overheating, MQ2 and MP2 record changes in combustible gas, pressure or fluctuations related to leaks or smoldering, MQ135 serves to assess air quality and flame sensors detect infrared radiation.

Signals from sensors are read periodically through the ATmega328 microcontroller, pre-processed and transmitted to the computer via the ESP8266 module. The computer performs two main functions including training DT, ANN and LSTM models from historical data and real-time inference to determine the risk level, thereby triggering warnings such as buzzers, display interfaces or sending messages, emails and phone calls. To apply to deep learning models, the collected data is grouped and organized into a three-dimensional data matrix including time axis, sensor axis and number of samples. This matrix form supports well the preprocessing steps and time series analysis, especially for LSTM models that need to group data in consecutive windows. Data collected continuously from sensors is saved as CSV files and data processing is done in Python language on PyCharm software platform with Pandas, Scikit learn and TensorFlow Keras libraries.

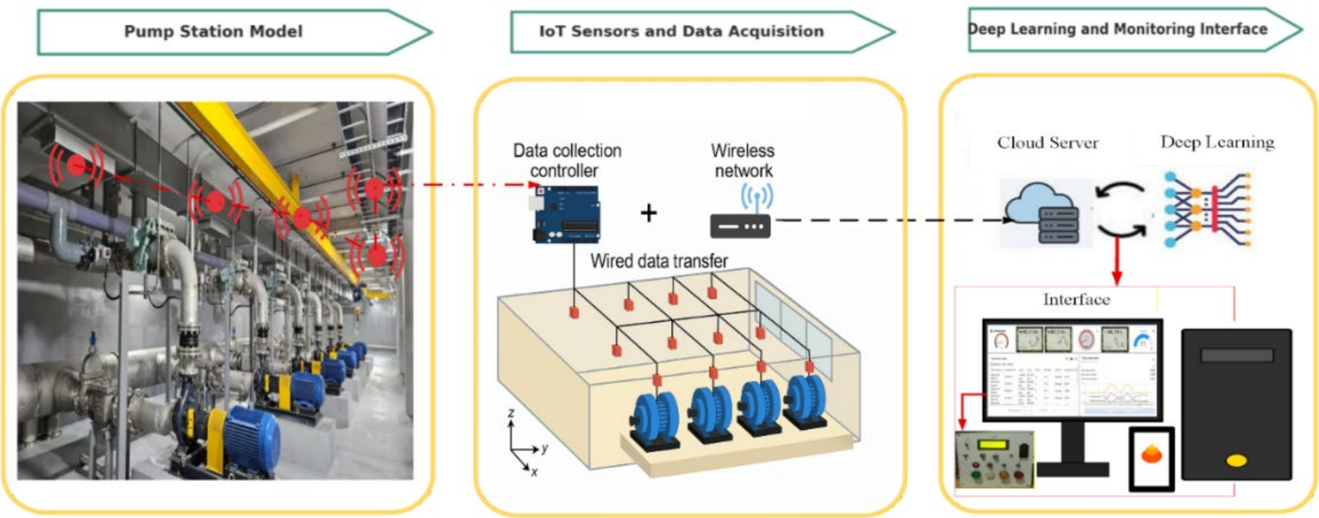

Figure 1. Fire monitoring and forecasting system model

2.1. Collect and store data

Sensor data were collected from a miniature fuel station equipped with multiple environmental sensors, including temperature, humidity, flammable gas, smoke, air quality, and flame detection sensors. These sensors were interfaced with embedded nodes based on Arduino and ESP8266 platforms, which periodically acquired sensor readings and transmitted the data wirelessly to a central processing computer.

At the receiving end, the incoming data stream is recorded and continuously stored in a structured file format together with corresponding timestamps. This storage strategy ensures the integrity and traceability of the time-series data and provides a reliable data source for subsequent preprocessing, data partitioning, and model training procedures described in the following sections.

2.2. Collect and store data

The pre-processing process includes the following main steps:

- Raw preprocessing: normalize column names, convert flame sensor signals to binary, cast data types, and remove missing, duplicate, or outlier values.
- Feature normalization: sensor signals are normalized by z-score or MinMaxScaler to bring the data to the same scale and help the learning model to be more stable.
- Time series organization: data is cut into sliding windows. For DT and ANN, a one-step window is used; for LSTM, a twelve-step window times six features are used. Labels are assigned according to the last sample of each series with three levels including Safe 0, Warning 1, and Danger 2.
- Data splitting: data is split into 70 % training, 15 % validation, and 15 % testing, ensuring uniform class distribution.
- Setting the warning threshold: from the predicted probabilities p0, p1 and p2 on the testing set, the quantity p unsafe is constructed to select the tau threshold to balance between low false alarm rate and good recall ability for dangerous situations.

After preprocessing, the data is organized into vectors or time series depending on the model used. With DT and ANN, each input sample is a vector of 6 features:

$$x = (temp, hum, mq2, mp2, mq135, flame) \qquad (1)$$

For the LSTM model, the data is formed into 12 x 6 time series windows. The sensor system operates with a fixed sampling interval; in the experiments, the sampling period was set to approximately 2 seconds per sample. Consequently, the 12-step input window of the LSTM model corresponds to an observation duration of about 24 seconds, enabling the model to capture temporal trends and risk accumulation over time while maintaining a prediction latency suitable for early-warning applications. The label of each sample belongs to one of three levels including Safe 0, Warning 1 and Danger 2. The dataset is divided in the ratio of 70 % for training, 15 % for validation and 15 % for testing.

2.3. Collect and store data

The performance of the proposed models was evaluated using standard classification metrics, including accuracy, precision, recall, and confusion matrix analysis, in combination with time-series smoothing techniques and hysteresis rules. These metrics are well suited for assessing multi-level risk classification performance under noisy sensor data conditions.

- Accuracy:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (2)$$

In there, TP is the number of samples belonging to the class to be detected and predicted correctly, TN is the number of samples not belonging to that class and predicted correctly, FP is the number of samples not belonging to the class but predicted incorrectly, and FN is the number of samples belonging to that class but missed. In the multi-class classification of Safety, Warning and Danger, when considering class i, we always consider class i as positive and the other two classes as negative.

- Precision of class i:

$$Precision_i = \frac{TP_i}{TP_i + FP_i} \quad (3)$$

In there, TP_i is the number of correctly predicted samples of class i, and FP_i is the number of samples that do not belong to class i but are mistakenly predicted to belong to class i.

- Recall of class i:

$$Recall_i = \frac{TP_i}{TP_i + FN_i} \quad (4)$$

In there, FN_i is the number of missed samples of class i.

- Confusion Matrix:

$$CM = \begin{bmatrix} N_{00} & N_{01} & N_{02} \\ N_{10} & N_{11} & N_{12} \\ N_{20} & N_{21} & N_{22} \end{bmatrix} \quad (5)$$

In there, N_{ij} is number of actual samples belonging to class i, but the model predicted class j, N_{01} is Safe samples predicted to be Warning, and N_{22} is Dangerous samples predicted correctly.

- Smoothing time series:

$$p^{smooth}(t) = \alpha\, p(t) + (1 - \alpha) p^{smooth}(t - 1) \quad (6)$$

In there, $p(t)$ is the predicted probability at time t, $p^{smooth}(t)$ is the probability after smoothing, α is the smoothing coefficient in the range zero point ten to zero point three, and low alpha values smooth more, high alphas react faster.

- Hysteresis rules for risk labeling:

$$\hat{y}(t) = \begin{cases} 2 & if\ p_{unsafe}(t) > \tau_2 \\ 1 & if\ p_{unsafe}(t) > \tau_1 \\ 0 & opposite \end{cases} \quad (7)$$

In there, $\hat{y}(t)$ is the risk label at time t, $p_{unsafe}(t)$ is the probability that the sample belongs to the unsafe group. τ_2 is the threshold for triggering danger, τ_1 is the threshold for triggering warning, and hysteresis avoids continuous label fluctuations in transition regions.

2.4. Quantitative thresholding

In this study, each time-series window is labeled according to the risk state at its final sample to facilitate early risk prediction. A safe state is assigned when all sensor signals at this time remain within normal operating ranges and no flame signal is detected, whereas an alert state is assigned when at least one sensor exceeds the alert threshold without reaching a dangerous level. In transitional regions, where sensor values fluctuate near decision thresholds or demonstrate an increasing temporal trend, a priority-based labeling principle is applied.

The labeling rules are implemented as follows:

- Safe state:

$$y(t) = 0 \text{ if } (\forall i: x_i(t) < T_i^{warn}) \wedge flame(t) = 0 \quad (8)$$

- Warm state:

$$y(t) = 1 \text{ if } \left(\exists i: T_i^{warn} \le x_i(t) < T_i^{danger}\right) \wedge flame(t) = 0 \quad (9)$$

- Dangerous state:

$$y(t) = 2 \text{ if } \left(\exists i: x_i(t) \ge T_i^{danger}\right) \vee flame(t) = 1 \quad (10)$$

In there, $x_i(t)$ is the measurement of the i-th sensor at the last time step, T_i^{warn} denotes the warning threshold of the i-th sensor, T_i^{danger} denotes the danger threshold of the i-th sensor, $flame(t) \in \{0,1\}$ denotes the flame sensor signal at the final sample, and y(t) denotes the risk label of the time series.

3. Fire monitoring and forecasting system model

3.1. Decision tree model

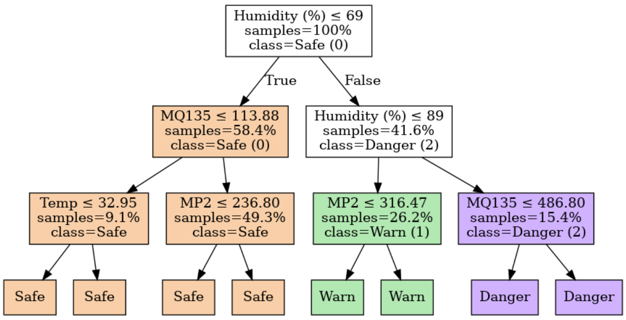

Figure 2. Decision tree structure after training

The decision tree model is chosen for its simplicity, fast inference speed and intuitive interpretation, suitable for real-time warning systems. Each data sample is classified by going through the branching nodes, feature selection model and splitting threshold to reduce the chaos of the data set at each node.

The index used is the Gini measure, which is defined as follows:

$$G = 1 - \sum_{i=1}^{C} p_i^2 \tag{11}$$

In which, p_i is the proportion of samples belonging to class i at the node under consideration.

The model receives as input 6 sensor features and 3 class labels. The decision tree is trained using the classification and regression trees (CART) algorithm with the maximum depth parameter adjusted during testing to avoid overfitting. After training, the tree structure allows for visual identification of the conditions leading to each risk level, supporting the interpretation of the model's decisions in an industrial context, as presented in Figure 2 and Algorithm 1.

Algorithm 1: *Training the DT model*	
Input	The training dataset consists of feature vectors and labels.
Output	Trained DT model and evaluation metrics.
1.	**Initialization:** The root node contains all the data.
2.	Calculate the Gini index ← (11).
3.	Try possible separation thresholds for each feature.
4.	Choose the threshold that gives the largest Gini reduction.
5.	Split the data into two sub-branches.
6.	Stop when the node reaches homogeneity or the depth exceeds a threshold.
7.	Output model evaluation results, model and test set predictions.
8.	**End**

3.2. Artificial Neural Network

Artificial neural networks are used to model the nonlinear relationship between sensor signals and the safety status of the pumping station. Unlike the decision tree model based on discrete thresholds, artificial neural networks allow continuous feature extraction and learn complex variation patterns in sensor data. In this study, the network is designed in a multi-layer feed-forward structure.

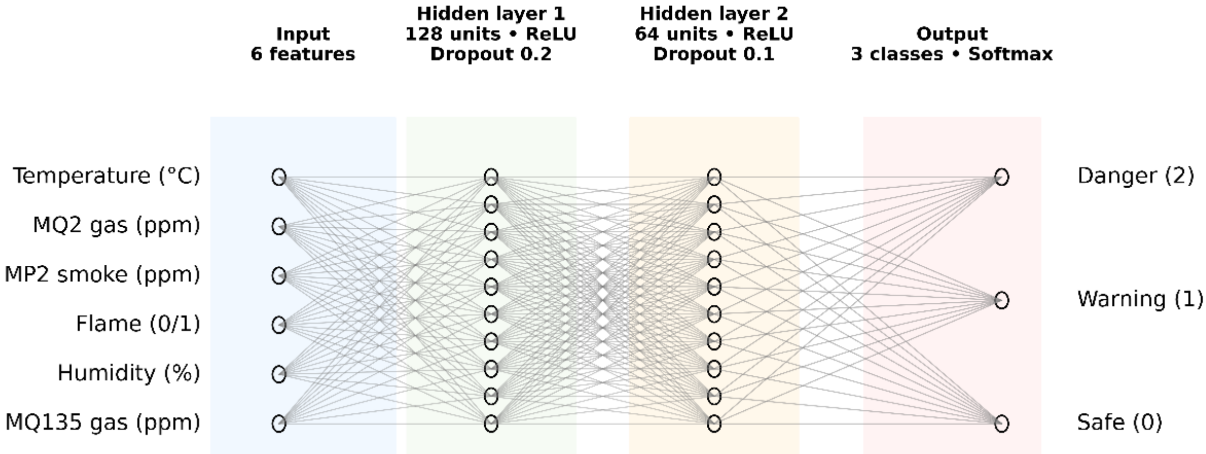

Figure 3. Structure of artificial neural network after training

At any neuron of the hidden layer, the pre-activation signal is computed as the weighted sum of the previous layer input plus a bias coefficient with the following general:

$$I_j = \theta_j + \sum_{i=1}^{m} w_{ji} x_i \tag{12}$$

In there, w_{ji} is the link weight between the i input and j neuron, θ_j is the bias value, m is the number of input features.

- In the experiment, the ReLU function is applied to the hidden layers:

$$y_j = f(I_j) = max(0, I_j) \tag{13}$$

- Calculate logit at the output layer:

$$z_k = \theta_k + \sum_j w_{kj}\, y_j \tag{14}$$

- Normalize the probability using Softmax:

$$\hat{y} = \frac{exp(z_k)}{\sum_{i=1}^{3} exp(z_i)} \tag{15}$$

- Training and backpropagation:

$$\Delta w_{j_i} = \frac{\partial l}{\partial w_{ji}} \tag{16}$$

$$\Delta \theta_j = \frac{\partial l}{\partial \theta_j} \tag{17}$$

- Adam optimization algorithm with updated parameters:

$$w_{ji}^{(new)} = w_{ji}^{(old)} - \eta \Delta w_{ji} \tag{18}$$

After the input layer consisting of six features including temperature, MQ2, MP2, flame sensor, humidity and MQ135, the model uses three hidden layers, combined with the ReLU activation function to increase the nonlinear learning ability shown in Figure 3. Dropout layers with the ratio of zero point fifteen, zero point ten and zero point zero five are placed between the hidden layers to reduce the overfitting phenomenon when the model is trained on noisy data. The output layer consists of three softmax nodes, corresponding to the three states including Safe, Warning and Danger. The total number of parameters of the model is about forty-three thousand parameters, which is enough to represent but still maintain fast inference speed. The training process uses the Adam optimizer and the Categorical Cross Entropy loss function. Model performance is evaluated on the validation set using standard classification metrics, including accuracy, precision, and recall. Early stopping is used to stop training when the error no longer improves, avoiding the model memorizing noise in the data, in Table 1 and Algorithm 2.

Table 1. ANN model training parameters

Parameter	Value
Total samples	≈ 38,040
Training samples	≈ 26,628 (70 %)
Validation samples	5,706 (15 %)
Test samples	5,706 (15 %)

Parameter	Value
Preprocessing	Data cleaning, Standard Scaler normalization
Feature order	[temp, mq2, mp2, flame, hum, mq135]
Gaussian Noise layer	Gaussian Noise (0.01)
Architecture	Dense 256 → 128 → 64
Activation	ReLU
Dropout	0.15 – 0.10 – 0.05
Max epochs	50
Optimizer	Adam (learning rate =0.001)
Loss	Categorical Cross-Entropy
Early Stopping	Patience = 5

Thanks to its compact structure and strong nonlinear learning ability, the artificial neural network gives stable results in many practical operating scenarios, especially at fast-varying signal segments. The concurrent model has the advantage of being easily deployed on Internet of Things systems or devices with limited computational resources.

Algorithm 2: *Training the ANN model*

```
Input    Historical data of collected sensors.
Output   Trained ANN model and evaluation metrics.
1.   Initialization: Split the dataset (train 70 %, validation 15 % and test 15 %).
2.   Initialization: Setting up the architecture and training parameters of the ANN model.
3.   While epochs ≤ EpochMax do
4.      Training ANN model.
5.      Model evaluation and comparison stops when performance reaches the optimal threshold.
6.      if latest profits that meet the termination conditions then
7.         Save the evaluation model.
8.      else
9.         epoch ← epoch+1 end go to Step 3.
10.     end if
11.  end while
12.  Output model evaluation results, model and test set predictions.
13.  End
```

3.3. Long Short-Term Memory

LSTM is a variant of RNN designed to process time series data with long-term dependencies, in Figure 4. Thanks to the structure of forget gates, input gates and output gates, LSTM is able to select important information and remove noise over time, helping to accurately model the fluctuations of sensor signals in fire scenarios.

At time t, an LSTM unit receives three inputs including x_t which is the feature vector at the current time, h_{t-1} which is the hidden state of the previous step and c_{t-1} which is the memory state of the previous step. The three gates in LSTM are defined as follows:

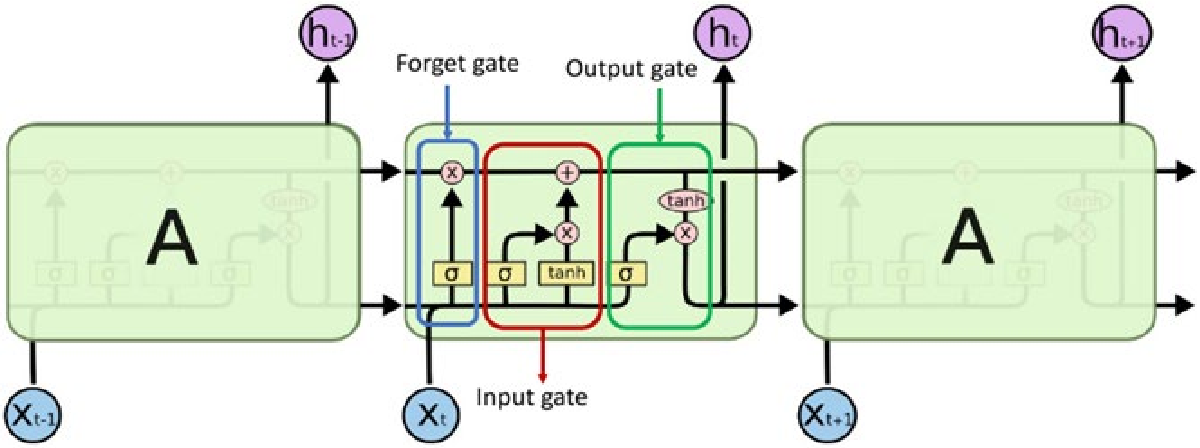

Figure 4. Internal unit structure of LSTM network

Forget gate is the information in the previous cell state that needs to be retained:

$$f_t = \sigma\left(W_f[h_{t-1}, x_t] + b_f\right) \tag{19}$$

In there, $\sigma(\cdot)$ is the sigmoid function for values in [0, 1], W_f is the weight matrix of the forget gate and b_f is the bias vector.

Input gate determines the amount of new information that needs to be written into the memory cell, including:

- Input trigger port:

$$i_t = \sigma\left(W_i\left[h_{t-1}, x_t\right] + b_i\right) \tag{20}$$

- Candidate state of memory cell:

$$\tilde{c_t} = \tanh\left(W_c\left[h_{t-1}, x_t\right] + b_c\right) \tag{21}$$

- Update the memory state at time t combining the old information selected by f_t and the new information determined i_t:

$$c_t = f_t \odot c_{t-1} + \odot \tilde{c_t} \tag{22}$$

In this problem, the LSTM input is a twelve-step time series window, each step consists of six features including temperature, MQ2, MP2, flame sensor, humidity and MQ135. The model uses a LSTM layer with 64 units to encode the entire measurement series into a hidden vector, then passes it through a 32 Dense layer with ReLU activation function and a three-dimensional softmax layer to predict the probability corresponding to three states including safe, warning and danger, in Table 2 and Algorithm 3.

Table 2. LSTM model training parameters

Parameter	Value
Total samples	≈ 38,040
Training samples	≈ 26,628 (70 %)
Validation samples	5,706 (15 %)
Test samples	5,706 (15 %)
Preprocessing	Data cleaning and Standard Scaler on 6 features
Sequence length	12 consecutive time steps
Input features	6 features (temp, mq2, mp2, flame, hum, mq135)
Architecture	LSTM (64) → Dense (32, ReLU) → Dense (3, Softmax)
LSTM units	64
Hidden dense units	32 (ReLU)
Output layer	Dense (3) with Softmax (3 classes)
Max epochs	50
Optimizer	Adam (learning rate=0.001)
Loss	Categorical Cross-Entropy
Early Stopping	Early Stopping, patience = 5

The total number of parameters of the model is about 38,040 samples, which is smaller than the ANN model but more effective in capturing trends and context over time. The model is trained using the Adam optimizer and the Categorical Cross Entropy loss function, with an early stopping mechanism to avoid overfitting. To ensure stable training of deep sequential models on noisy time-series data, the Adam optimization algorithm was adopted, owing to its adaptive learning-rate mechanism and robust convergence properties. Thanks to the ability to learn dynamic dependencies and reduce sensitivity to noise, LSTM achieves the most accurate and stable results in experimental scenarios. In particular, LSTM responds well. The classifier allows for three-level risk prediction with real-time inference capabilities.to long stunts and continuously varying signal segments, which are difficult for DT and ANN models to represent accurately.

From a system design perspective, the models demonstrate distinct trade-offs between computational complexity and predictive capability. In particular, sequential architectures exhibit superior capability in capturing the temporal characteristics of dynamically evolving sensor data, thereby enhancing the reliability of fire and explosion risk predictions under unstable operating conditions.

Algorithm 3: *Training the LSTM model*

```
Input     Historical data set obtained from the sensor.
Output    Model evaluation and prediction results on the test set.
1.   Initialization: Data preprocessing.
2.   Initialization: Split the dataset (train 70 %, validation 15 % and test 15 %).
3.   Initialization: LSTM model architecture and training parameters setup.
4.   While epochs ≤ EpochMax do
5.     Training LSTM network model.
6.     Evaluate the model to stop the training process early.
7.     if loss function no longer decreases after 5 epochs then
8.          Save evaluation model, stop training early.
9.     else
10.         epoch ← epoch+1 end go to Step 4.
11.    end if
12.  end while
13.  Output model evaluation results, model and test set predictions.
14.  End
```

4. Results and evaluation

4.1. Experimental model

The overall architecture of the fire monitoring and forecasting system on the pumping station model, including the main functional blocks and the sensor arrangement at the scene is described in Figure 5. At the input stage, environmental sensors and smoke fire sensors are connected via IO and ESP modules to collect real-time data. The data is then passed to the processing block, where cleaning, normalization, and windowing of the time series are performed before being passed to the classification module using a machine learning model. The lower part of the figure shows the sensor installation locations at the pumping station, including the flame sensor cluster and multi-channel environmental sensors, serving the continuous monitoring of key areas. This empirical model provides a basis for evaluating system reliability under various operating conditions.

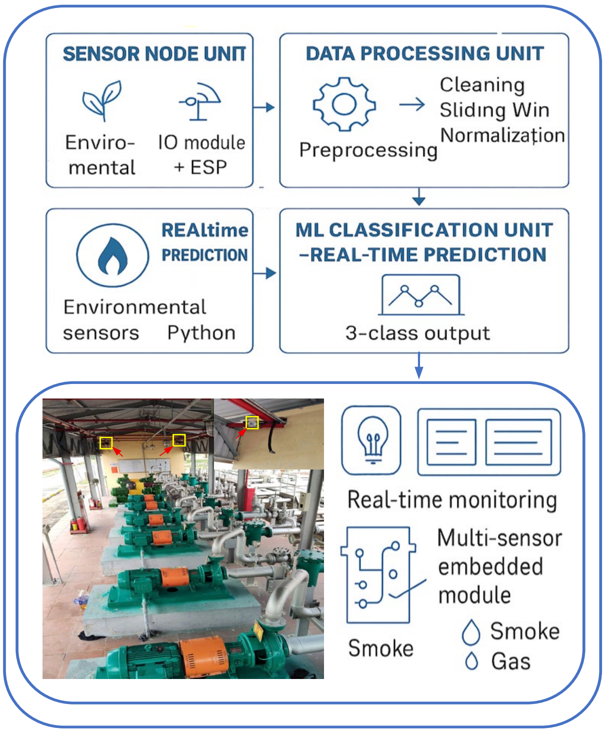

Figure 5. Experimental model and configuration of multi-sensor monitoring system at pumping station

4.2. Training results

Based on the results in Table 3 and from Figure 6 to Figure 8, it can be seen that the LSTM quantitative results achieved the highest indexes: accuracy around 0.94, precision 0.93, ROC-AUC, and PR-AUC are both above 0.94. ANN achieves accuracy around 0.92, precision is 0.90, while DT achieves around 0.88 and lower precision. Confusion matrix shows that all three models recognize the Safety class well and the main difference lies in the Warning and Danger classes. LSTM gives a higher and more balanced prediction rate between these two classes; ANN is close but still tends to make mistakes in part of the samples at the edge while DT makes more mistakes when the data is noisy or rapidly changing. The summary tables show that LSTM has a higher mean p_{unsafe}, showing high sensitivity to risky regions, DT is more conservative with large mean p_0, while ANN is in the middle.

Table 3. Parameters of the marine diesel engine

Model	Accuracy	Precision	Recall	ROC-AUC	PR-AUC
DT	0.88	0.86	0.84	0.91	0.89
ANN	0.92	0.90	0.88	0.94	0.93
LSTM	0.94	0.93	0.91	0.96	0.95

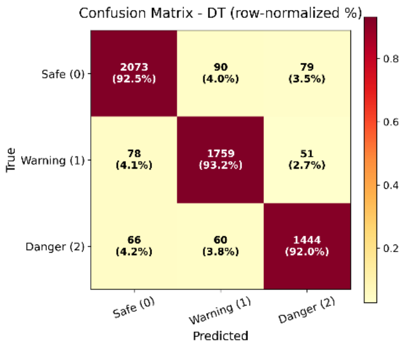

Figure 6. Confusion matrix of DT models

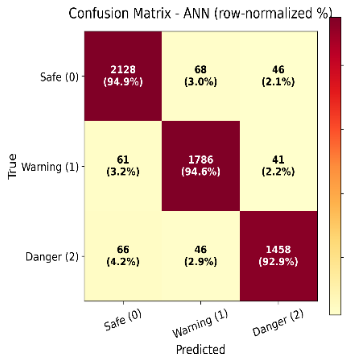

Figure 7. Confusion matrix of ANN models

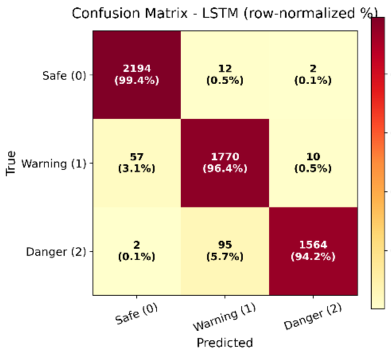

Figure 8. Confusion matrix of LSTM models

4.3. Experimental results and evaluation of results

Figure 9 depicts the variation of the dangerous probability p_{unsafe} over time for three classification models including DT, ANN and LSTM on the experimental data series. The results show that DT generates discrete signals with large oscillation amplitudes and is very sensitive to sensor noise, leading to strong variations of p_{unsafe} when the input conditions change slightly, similar to study [26]. The binary branching characteristic of decision trees causes the output to be hard segmented and amplify local noise, so the signal needs to be filtered to properly reflect the risk trend over time. Meanwhile, ANN gives a more stable signal, showing well the phases of increasing or decreasing risk in the data, but there are still slight fluctuations and delays in the transition zone when environmental conditions change suddenly. LSTM gives the most stable results of the three models: the output signal is relatively smooth even without filtering and follows long-term risk evolution phases thanks to its ability to exploit time-dependent information in the data. This shows that LSTM is more suitable for continuous time series forecasting problem, while ANN achieves intermediate level and DT is less stable when working with noisy data.

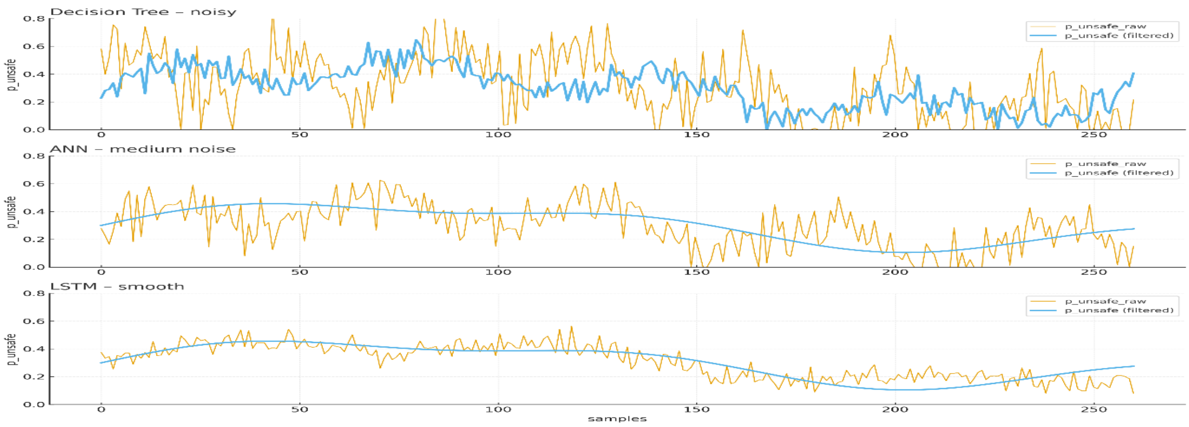

Figure 9. Comparison of p_{unsafe} over time of three models DT, ANN and LSTM on experimental series

Table 4. Compare statistical summary of prediction results

Index	DT	ANN	LSTM
Total predicted samples	Long sequence (stable background data)	43 consecutive samples	2019 consecutive samples
Risk level distribution (Safe – Warning – Dangerous)	≈ 100 % – 0 % – 0 %	≈ 35 % – 65 % – 0 %	≈ 87 % – 12 % – 0,4 %
Number of transitions between risk levels	0 (none observed)	4 transitions (2 Safe → Warning, 2 Warning → Safe)	59 transitions over the entire sequence

Typical confidence characteristics	p_0 stable around ≈ 0.67; p_2 baseline ≈ 0.33; p_1 ≈ 0	Average confidence ≈ 0.70; high for Safe, lower for Warning	Strongly concentrated probabilities

Table 4 presents the summary statistics of the prediction results of three models DT, ANN and LSTM on experimental time series data. The DT model shows high stability but hardly reflects different risk levels because the forecasts are mostly in the Safety class. ANN provides a more balanced risk distribution and can observe state transitions between levels, however the confidence level over time still fluctuates and is not really prominent in dangerous stunts. LSTM gives the best results with the ability to follow long risk segments, large total number of prediction samples and clear number of state transitions. The confidence level of LSTM is strongly concentrated at the Safe and Danger levels, showing that the model is good at identifying risky regions in the time series. In the proposed system, a sensor error detection mechanism is incorporated into the real-time prediction phase by monitoring missing values, physical boundary violations, abrupt signal changes, and signal stagnation for each sensor. Upon error detection, the affected data sample is flagged to notify the operator and excluded from the LSTM input window, thereby preventing distortion of the prediction outcomes.

To clarify the novel aspects and assess the ability to address practical problems, the results of this study were compared with previously published works to ensure objectivity in evaluating the effectiveness of the proposed solution, as presented in Table 5. The comparison results indicate that, while previous studies mainly focused on fire detection based on static criteria and fixed thresholds, the proposed approach extends toward multi-sensor time-series data analysis, enabling earlier detection and multi-level assessment of fire and explosion risk with improved stability and reliability.

Table 5. Comparison of the proposed approach with previously published studies

Study	Platform & Sensors	Method	Detection Objective	Key Features
Study [27]	ESP32, basic environmental sensors (temperature, gas, humidity)	Simple threshold comparison	Forest fire detection	Simple IoT system, easy to deploy, low cost
Study [28]	ESP32, wireless sensor network (temperature, humidity, pressure, gas concentration)	Statistical analysis	Early fire detection	Fast detection time, includes reliability assessment of devices
This study	Arduino, ESP8266, multi-sensor system (temperature, gas, smoke, humidity, flame, composite gas)	Deep learning and time-series analysis	Early detection and multi-level risk warning	Exploits dynamic relationships among sensors, noise reduction, multi-level risk assessment based on trained deep learning models

5. Conclusion

This study proposed and implemented a fire risk monitoring and forecasting system based on real-time sensor data, tested on a miniature industrial pumping station model. By integrating various environmental sensors and fire/smoke sensors, the system generates multi-channel data streams that fully reflect the operating states from safe background to dangerous activation phases. On that data, three machine learning models including DT, ANN and LSTM are trained and evaluated according to the same standard process, ensuring objectivity and comparability. The results show that the LSTM model has the best performance: high accuracy, precision and good tracking ability in long-term danger zones. ANN has quite good signal smoothness but is less stable in the warning layer, while DT reacts quickly but is susceptible to noise and depends heavily on pre-processing conditions. The comparison of p_{unsafe} signals over time also confirms that LSTM is most suitable for continuously operating warning systems, where stability and high sensitivity to abnormal signs are required.

The experimental model has demonstrated the feasibility of applying deep learning in monitoring fire and explosion risks at pumping stations, and at the same time provides a basis for expansion to more complex industrial environments. In the future, the system can be developed in the direction of increasing the number of sensors, improving data resolution, incorporating more hybrid models, or applying reinforcement learning techniques to improve accuracy and reduce false alarms. The obtained results show that this approach has high application potential and can become the foundation for smart warning systems in practice.

References

[1] Jang H.-Y and Hwang C.-H, “Configuration Approaches of CFAST for Prediction of Smoke and Heat Detector Activation Times in Corridor Fires,” Applied Sciences, vol. 13, no. 24, p. 13161, 2023.

[2] H. Y. Jang and C. H. Hwang, "Obscuration Threshold Database Construction of Smoke Detectors for Various Combustibles," Sensors, vol. 20, no. 21, p. 6272, 2020.
[3] M. Ali et al., "A Comprehensive Review of Advanced Sensor Technologies for Fire Detection with a Focus on Gasistor-Based Sensors," Chemosensors, vol. 13, no. 7, p. 230, 2025.
[4] J. Fonollosa, A. Solórzano, and S. Marco, "Chemical Sensor Systems and Associated Algorithms for Fire Detection: A Review," Sensors, vol. 18, no. 2, p. 553, 2018.
[5] L. Wang, X. Zhang, L. Li, B. Li, and Z. Mei, "Experimental Study on Early Fire Smoke Characteristics in a High-Volume Space: A Fire Detection Perspective," Fire, vol. 7, no. 9, p. 298, 2024.
[6] S. Chen, J. Ren, Y. Yan, M. Sun, F. Hu, and Zhao, "University of Dundee Multi-sourced sensing and support vector machine classification for effective detection of fire hazard in early stage," Computers & Electrical Engineering, vol. 104, 2022.
[7] Wang. L et el., "Multi-Height and Heterogeneous Sensor Fusion Discriminant with LSTM for Weak Fire Signal Detection in Large Spaces with High Ceilings," Electronics, vol. 13, no. 13, 2024.
[8] N. K. Vaegae, V. Annepu, K. Bagadi, and F. Dahan, "Multisensor Fuzzy Logic Approach for Enhanced Fire Detection in Smart Cities," Journal of Optimization, vol. 2024, no. 1, p. 8511649, 2024.
[9] H. Wang, Y. Shi, L. Chen, and X. Zhang, "A Tunnel Fire Detection Method Based on an Improved Dempster-Shafer Evidence Theory," Sensors, vol. 24, no. 19, p. 6455, 2024.
[10] R. Chagger, "The performance of multi-sensors in fire and false alarm tests Briefing Paper," Watford, U.K.: BRE, 2018.
[11] F. Khan et al., "Recent advances in sensors for fire detection," Sensors, vol. 22, no. 9, p. 3310, 2022.
[12] F. Abid and N. Izeboudjen, "Predicting Forest Fire in Algeria Using Data Mining Techniques: Case Study of the Decision Tree Algorithm," Advances in Intelligent Systems and Computing, vol. 1105 AISC, pp. 363–370, 2020.
[13] A. R. Devi, N. M. Balaji, and S. E. Raj, "Iot based automatic forest fire detection based on machine learning," Alochana Journal, vol. 14, pp. 2231–6329, 2025.
[14] B. Lindemann, T. Müller, H. Vietz, N. Jazdi, and M. W.-P. Cirp, "A survey on long short-term memory networks for time series prediction," Procedia CIRP, vol. 99, pp. 650-655, 2021.
[15] D. G. Bhuvaneswari et al., "Early fire hazard prediction framework in smart cities using deep learning with antlion optimization algorithm," Journal of Theoretical and Applied Information Technology, vol. 103, no. 15, 2025.
[16] S. H. Noh and H. J. Moon, "Anomaly Detection Based on LSTM Learning in IoT-Based Dormitory for Indoor Environment Control," Buildings, vol. 13, no. 11, p. 2886, 2023.
[17] T. Chi, N. D. Thach, and H. Khanh, "Research and Design of a Fire Alarm System for Ships," Journal of Marine Science and Technology, no. 70, pp. 48–53, 2022.
[18] Directorate for Standards, Metrology and Quality, "Fire Detection and Alarm System – Technical Requirements," TCVN 5738, pp. 1–26, 2021.
[19] V. H. Thai, "Performance Evaluation of MQTT and HTTP Protocols in a Real-Time IoT System," Journal of Science and Technology – The University of Danang, vol. 20, no. 3, pp. 51–56, 2022.
[20] H. M. Dung and K. D. A. Khang, "Forecasting Air Quality by the LSTM-MA Model, Using Data at the Gieng Nuoc Intersection Automatic Monitoring Station, Ba Ria – Vung Tau Province," Vietnam Journal of Hydrometeorology, vol. 765, pp. 75–89, 2024.
[21] P. N. T. Doan, H. L. Trinh, T. V. Nguyen, H. T. T. Le, and P. V. Le, "Forest fire risk prediction using geospatial data and machine learning techniques, a case study in the western region of Nghe An province," Journal of Mining and Earth Sciences, vol. 65, no. 5, pp. 50–60, 2024.
[22] P. N. Pha, N. T. Hieu, N. V. Thang, N. T. Son, Q. C. Hoang, and P. M. Trien, "Design and Implementation of Fire Early Warning System in In-Building Infrastructure Based on Deep Learning," Journal of Science and Technology, vol. 56, no. 2, pp. 48–53, 2020.
[23] K. T. Tran, C. T. N. Duong, and M. H. Vo, "Applying Neural Network for Rain Forecast with Wireless Sensor Network," Journal of Technical Education Science, vol. 65, pp. 10–20, 2021.
[24] M. Truong, "Application of IoT Technology and Sensor Networks in Environmental Management and Improving the Shrimp–Rice Farming Model Efficiency in An Bien District, Kien Giang Province," vol. 58(SDMD), pp. 31-41 2022.
[25] N. N. Trung and B. T. Duyen, "Behavioral Pattern Prediction Method Based on IoT Data Analysis Using Deep Neural Network for Automatic Control," Journal of Energy Science and Technology – Electric Power University, vol. 29, pp. 110–121, 2022.
[26] A. A. Alharbi, "Classification Performance Analysis of Decision Tree-Based Algorithms with Noisy Class Variable," Discrete Dyn Nat Soc, vol. 2024, no. 1, p. 6671395, 2024.
[27] Y. Subbarayudu, G. V. Reddy, J. Bhargavi, and K. Latha, "An Efficient IoT-Based Novel Approach for Fire Detection Through Esp 32 Microcontroller in Forest Areas," MATEC Web of Conferences, vol. 392, p. 01109, 2024.
[28] C. F. I. Garcia and J. B. G. Ibarra, "Efficiency and Performance Evaluation of an Early Fire Detector Device Using an ESP32 Wireless Sensor Network," 2023 2nd International Conference on Vision Towards Emerging Trends in Communication and Networking Technologies (ViTECoN), Vellore, India, pp. 1-6, 2023.