Chinese Journal of Chemical Engineering 85 (2025) 182–188

Contents lists available at ScienceDirect

Chinese Journal of Chemical Engineering

journal homepage: www.elsevier.com/locate/CJChE

Full Length Article

Consequence prediction using variable-length concentration time series for gas turbine enclosure

Shikuan Chen , Wenli Du *, Chenxi Cao , Bing Wang

Key Laboratory of Smart Manufacturing in Energy Chemical Process, Ministry of Education, East China University of Science and Technology, Shanghai 200237, China

ARTICLE INFO

Article history:
Received 6 October 2024
Received in revised form
19 April 2025
Accepted 20 April 2025
Available online 14 May 2025

Keywords:
Consequence prediction
Variable-length time series
Deep learning
Computational fluid dynamics

ABSTRACT

Flammable gas leakage in a semi-enclosed scenario can lead to catastrophic consequences, such as vapor cloud explosions. To reduce casualties and environmental damage, predicting the consequences based on the initial concentration time series monitored by sensors is of paramount importance. This paper proposes a consequence prediction model based on deep learning using variable-length concentration time series. Incomplete concentration values are padded and then passed through a masking layer, enabling the network to focus exclusively on valid data. The temporal correlations are extracted using a long short-term memory (LSTM) network, and the final prediction results are obtained by passing these features into a feedforward neural network (FNN). Computational fluid dynamics (CFD) software was used to simulate the leakage of hydrogen-mixed natural gas. Experiments were carried out for nine distinct prediction targets, derived from combinations of the mass and centroid coordinates of vapor clouds formed by various gases. These prediction targets were modeled using both fixed-length and variable-length input sequences. The high accuracy of the experimental results validates the effectiveness of the proposed method.

1. Introduction

In today's rapidly developing industrialized world, the leakage of flammable and explosive gases can cause severe accidents and substantial environmental damage [1–4]. In enclosed spaces, such as turbine enclosures, if the concentration of leaked explosive gases reaches the range between the lower and upper explosive limits, vapor cloud explosions (VCE) may occur, resulting in significant casualties [5,6]. To reduce such losses, it is crucial to accurately estimate the spatiotemporal concentration distribution of gases based on the time series of concentrations detected by sensors during the initial stages of leakage. Additionally, it is important to estimate the mass and centroid coordinates of the vapor cloud formed after the gases reach a steady-state diffusion.

Various methods have been proposed for predicting the consequences of VCE, each with its own strengths and limitations. These approaches can be broadly classified into three categories: empirical models, numerical simulations, and data-driven models [7,8]. The classical empirical models, including the TNO multi-energy method [9,10], the trinitrotoluene equivalency model [11,12], and the Baker–Strehlow–Tang model [13,14], often suffer from significant limitations due to their reliance on simplified assumptions and empirical correlations. These methods are particularly inadequate in scenarios involving obstacles, where they fail to accurately capture complex interactions and spatial variations. Numerical methods, primarily referring to computational fluid dynamics (CFD) tools, can overcome these limitations and have been proven effective [15–20]. However, CFD simulations are computationally expensive. Data-driven approaches have benefited from the rapid development of machine learning methods in recent years [21,22]. Machine learning models not only excel at capturing complex non-linear relationships between variables and exhibiting self-learning capabilities, but they also provide results more quickly [23]. Xu *et al.* [24] presented a beetle antennae search improved back-propagation neural network model for predicting gas explosion pressures. Zhou *et al.* [25] proposed a novel approach for predicting explosion overpressure consequences in petrochemical industry vapor cloud explosions by combining numerical simulation with neural network technology. Since the initial concentration sequence of a leak can determine the distribution of steady-state concentration, thereby affecting the vapor cloud's mass and centroid position, it is more practical to predict consequences based on the initial leak concentration sequence. However, few existing

* Corresponding author.
E-mail address: wldu@ecust.edu.cn (W. Du).

https://doi.org/10.1016/j.cjche.2025.04.013

works utilize the initial leak concentration sequence for consequence prediction. This paper aims to address this issue by proposing a predictive model that adapts to varying lengths of sensor data over time.

LSTM is widely used in processing time series data in chemical processes, and it plays an important role in capturing long-term dependencies [26,27]. However, for variable-length inputs, a single LSTM model may struggle to adapt effectively to different input lengths. In the field of neural machine translation, sentences to be translated are typically of varying lengths [28]. Inspired by this, this paper draws an analogy between concentration sequences of different lengths and sentences of different lengths in translation. Padding is used to align the concentration sequences, while a masking layer is applied to disregard the padded, non-informative data [29–31]. This approach enables the model to effectively handle variable-length inputs for future consequence prediction.

The main contributions of this paper are as follows:

(1) To tackle the real-time consequence prediction problem, a deep learning-based model using variable-length concentration series data is proposed.
(2) A masking layer is added to the variable-length input after the padding operation. An LSTM component is used to extract the temporal high-dimensional features, which are subsequently passed into a feedforward neural network (FNN) to obtain the final results.
(3) CFD simulations are performed to prepare the experimental data, considering different prediction targets. The high accuracy of the results indicates the effectiveness of the proposed method.

2. Scenario Description and Problem Statement

2.1. Scenario description

In this study, we employ the fire dynamics simulator (FDS) as the CFD software, specifically designed to simulate heat transfer, smoke propagation and combustion processes during fire incidents [32]. FDS has been rigorously validated for its application in modeling indoor diffusion processes, and its high precision, flexibility and open-source nature render it an indispensable tool for advancing safety measures [33–35]. Regarding its underlying principles, FDS utilizes numerical techniques to solve the Navier-Stokes equations, encompassing the continuity equation, momentum balance equation and energy balance equation, which are respectively delineated as follows [36]:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \boldsymbol{U}) = 0 \tag{1}$$

$$\frac{\partial (\rho \boldsymbol{U})}{\partial t} + \boldsymbol{U} \cdot \nabla (\rho \boldsymbol{U}) = -\nabla P + \nabla \cdot \boldsymbol{\tau} \tag{2}$$

$$\frac{\partial (\rho E)}{\partial t} + \nabla \cdot (\rho E \boldsymbol{U}) = \nabla \cdot \left(k \nabla T_p\right) + S \tag{3}$$

where ρ represents the density, $\boldsymbol{U}$ denotes the velocity vector, P signifies the pressure, $\boldsymbol{\tau}$ refers to the stress tensor, E corresponds to the total energy per unit mass, k indicates the thermal conductivity, T_p represents the temperature and S accounts for any heat sources or sinks.

In our study, we use a semi-enclosed gas turbine enclosure as the scenario, as shown in Fig. 1, with an approximate size of 17 m × 16 m × 13 m. The outer enclosure has ventilation windows and an exhaust fan above, both of which ensure a stable airflow inside the enclosure.

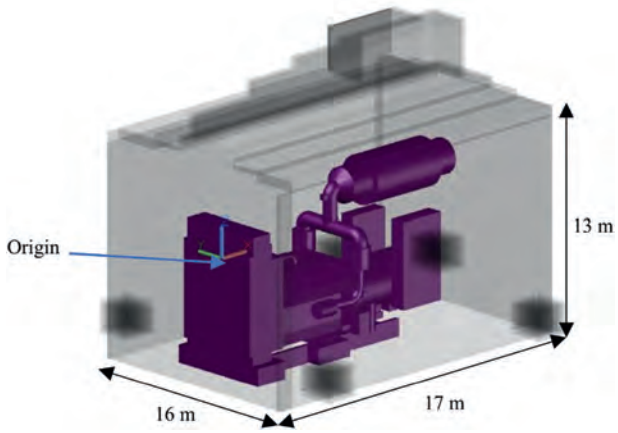

Fig. 1. Schematic layout of the experimental scenario.

A total of 89 different leakage scenarios were simulated using FDS. Methane and hydrogen are mixed at a 9:1 ratio and released from 8 possible leakage points over a duration of 300 s. The leakage rates include 0.25, 0.5, 1 and 2 $kg \cdot s^{-1}$, while the fan speeds include 1, 2 and 3 $m \cdot s^{-1}$. The grid size is set to 0.25 m. It is worth noting that these possible conditions are calculated based on certain parameters from the actual scenario. A total of 10 sensors are randomly distributed inside the enclosure, and each sensor simultaneously recorded the concentrations of both hydrogen and methane. The specific coordinates of the leakage sources and sensors are listed in Table 1 and Table 2, respectively. These eight leakage sources in Table 1 are strategically positioned at pipeline corners, where mechanical stress and operational conditions make them more prone to leaks. Furthermore, leaks at these locations are more likely to result in gas accumulation, potentially forming explosive vapor clouds. Similarly, the ten sensors in Table 2 are placed within the enclosure at positions where vapor clouds are likely to accumulate.

2.2. Problem statement

In this paper, we aim to predict the mass and centroid location of the vapor cloud using the initial concentration values before reaching steady state. The mass and centroid location of the vapor cloud determine the level of overpressure and its associated hazards from a VCE. The concentration values detected by sensors at the early stage of the leak can assist decision-makers in promptly implementing measures to mitigate losses. The mathematical formulation is presented in Eq. (4):

Table 1
Coordinates of leak positions in the experimental setup.

Leak No.	Coordinates/m	Leak No.	Coordinates/m
1	(6.25, 0, 1.25)	5	(4, −1.75, −1.5)
2	(1.5, 1.25, 0.25)	6	(10, −0.75, −2.25)
3	(7, −1.75, −3)	7	(−0.5, 0, −2.25)
4	(7, 1.75, −3)	8	(7.25, 0, 0)

Table 2
Coordinates of the sensor positions in the experimental setup.

Leak No.	Coordinates/m	Leak No.	Coordinates/m
1	(14.25, 2.25, 2.75)	6	(3, −1.25, 2)
2	(5, −2.25, 2.75)	7	(12.5, −2, 3.75)
3	(13.25, 2.75, 3)	8	(−0.25, −4.25, 2)
4	(6.75, −0.75, 3.75)	9	(13.25, 5, 2.75)
5	(9.25, 3.25, 3.75)	10	(12.75, 4, 3)

$$M_{\text{vce}}, (x, y, z)_{\text{vce}} = F_\theta(c_1, \cdots, c_T) \tag{4}$$

where M_{vce} represents the mass of the vapor cloud, $(x, y, z)_{\text{vce}}$ denotes the 3-dimensional coordinates of the vapor cloud centroid, F represents the prediction model, θ denotes the model parameters, and $c_1, \cdots, c_T$ represents the concentration series of length T.

In this paper, we consider two types of prediction problems: fixed-length input and variable-length input (depending on whether T is fixed or variable). The former is suited for offline scenarios, and the latter is suited for online scenarios involving the gradual acquisition of effective input concentration.

3. Method

3.1. Workflow

The workflow for consequence prediction is illustrated in Fig. 2.

A. Collect scenario information, including the gas turbine and the size of its enclosure. It is important to note that this paper primarily aims to explore scientific methods, so specific details of the actual system are omitted for simplicity. After setting the simulation parameters, use FDS software to perform diffusion simulation calculations.
B. Extract concentration data at sensor locations and preprocess the data for model training. This includes calculating the centroid and mass of the vapor cloud and normalizing the data.
C. Determine whether to apply padding operations and add masking layers based on whether the input is variable-length or not.
D. Since the gas mixture is hydrogen-methane, predict the centroid and mass of the vapor cloud for various gas mixtures. Nine distinct prediction targets are available.
E. After randomly dividing the dataset, use the training set data to train the model.
F. Input the test set data into the trained model and evaluate its accuracy to assess the effectiveness of the method.

3.2. Data preprocessing

Since the concentration sequence is obtained from the diffusion scenario without a corresponding vapor cloud explosion simulation, we indirectly determine the consequences of the vapor cloud explosion by calculating the vapor cloud mass and centroid coordinates based on the steady-state concentration.

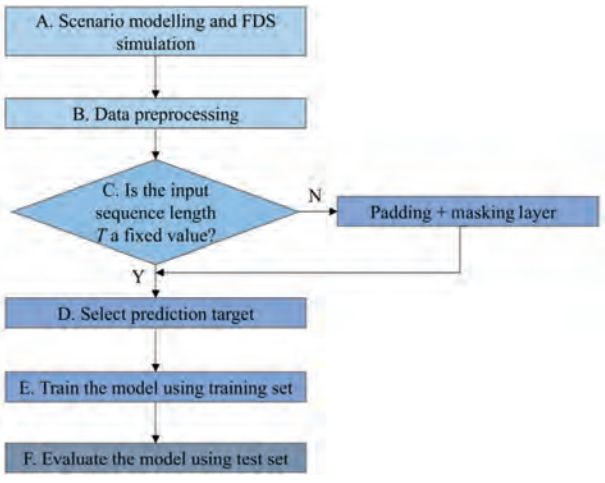

Fig. 2. Flowchart of the consequence prediction process.

a. First, calculate the volume fraction based on the grid concentration values.

$$v_{\text{f}} = \frac{cRT}{\rho P} \tag{5}$$

b. Based on the lower and upper explosion limits, filter out the grid concentration values that meet the explosion conditions, *i.e.*, the volume fractions greater than or equal to the lower explosion limit (LEL) and less than or equal to the upper explosion limit (UEL). Denote the concentration values that meet these conditions as c_e.

$$\text{LEL} \le c_e \le \text{UEL}, c_e \in v_{\text{f}} \tag{6}$$

c. Multiply the grid concentration values that meet the conditions by the grid volumes and sum them up to obtain the vapor cloud mass.

$$M_{\text{vce}} = \sum_e c_e V_{\text{grid}} \tag{7}$$

d. Calculate the centroid coordinates of the vapor cloud based on the concentration values and the corresponding coordinate values that meet the explosion conditions:

$$\begin{aligned} x &= \frac{c_1 x_1 + c_2 x_2 + \cdots + c_n x_n}{c_1 + c_2 + \cdots + c_n} \\ y &= \frac{c_1 y_1 + c_2 y_2 + \cdots + c_n y_n}{c_1 + c_2 + \cdots + c_n} \\ z &= \frac{c_1 z_1 + c_2 z_2 + \cdots + c_n z_n}{c_1 + c_2 + \cdots + c_n} \\ c_1, c_2, \cdots, c_n &\in c_e \end{aligned} \tag{8}$$

where v_{f} is the volume fraction, c is the concentration value, ρ is the gas density, R is the ideal gas constant, T is the temperature, P is the pressure value, c_e is the grid concentration value that meets the explosion conditions, V_{grid} is the unit volume of the grid.

e. Given the inherent variability in the calculation results, take the average values of the vapor cloud mass and centroid coordinates over the last 10 s of the steady-state concentration sequence as the target values for the prediction task, as shown in Eq. (9):

$$\begin{aligned} \overline{M}_{\text{vce}} &= \frac{1}{10} \sum_{i=290}^{299} M_{\text{vce}}^{(i)} \\ \overline{x} &= \frac{1}{10} \sum_{i=290}^{299} x^{(i)} \\ \overline{y} &= \frac{1}{10} \sum_{i=290}^{299} y^{(i)} \\ \overline{z} &= \frac{1}{10} \sum_{i=290}^{299} z^{(i)} \end{aligned} \tag{9}$$

f. Normalization. After calculating the mass and centroid of the vapor cloud, the input data and output data should be min-max normalized. The whole dataset is randomly split into a training set and a test set with a ratio of 9:1 for subsequent model training and evaluation.

$$
\begin{aligned}
c_{\text{norm}} &= \frac{c-\min(c)}{\max(c)-\min(c)} \\
M_{\text{norm}} &= \frac{M-\min(M)}{\max(M)-\min(M)} \\
x_{\text{norm}} &= \frac{x-\min(x)}{\max(x)-\min(x)} \\
y_{\text{norm}} &= \frac{y-\min(y)}{\max(y)-\min(y)} \\
z_{\text{norm}} &= \frac{z-\min(z)}{\max(z)-\min(z)}
\end{aligned}
\tag{10}
$$

3.3. Consequence prediction model using concentration time series data

Since the vapor cloud mass and centroid are related to the steady-state concentration, which is influenced by the initial concentration during the leakage phase, this paper employs LSTM to learn the temporal features of the initial fixed-length concentration sequence. The LSTM equations are presented in Eq. (11):

$$
\begin{aligned}
f_t &= \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right) \\
i_t &= \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \\
\tilde{C}_t &= \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \\
C_t &= f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t \\
o_t &= \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \\
h_t &= o_t \cdot \tanh(C_t)
\end{aligned}
\tag{11}
$$

where f_t is the forget gate at time step t, i_t is the input gate activation at time step t, $\tilde{C}_t$ is the candidate cell state at time step t, C_t is the cell state at time step t, o_t is the output gate activation at time step t, h_t is the hidden state at time step t, W_f, W_i, W_c, and W_o are weight matrices, and b_f, b_i, b_C, and b_O represent bias terms. h_{t-1} is the hidden state from the previous time step, x_t is the input at the current time step, C_{t-1} is the cell state from the previous time step, and σ, tanh denote the sigmoid and hyperbolic tangent activation functions, respectively.

The learned high-dimensional abstract features are then fed into a FNN to directly predict the outcomes. The schematic diagram of the fixed-length-input prediction model is shown in Fig. 3(a). For variable-length inputs, since neural networks require fixed input shapes, padding operations are applied, specifically zero-padding the areas without concentration data. In other words, even for the variable-length input model, the input has a fixed maximum length. Then, a masking layer is added, which enables the network to focus on the relevant data while ignoring the zero-padded portions. The schematic diagram of the variable-length-input model can be seen in Fig. 3(b). The number of neurons in the output layer of the FNN depends on the mass or centroid position of the vapor

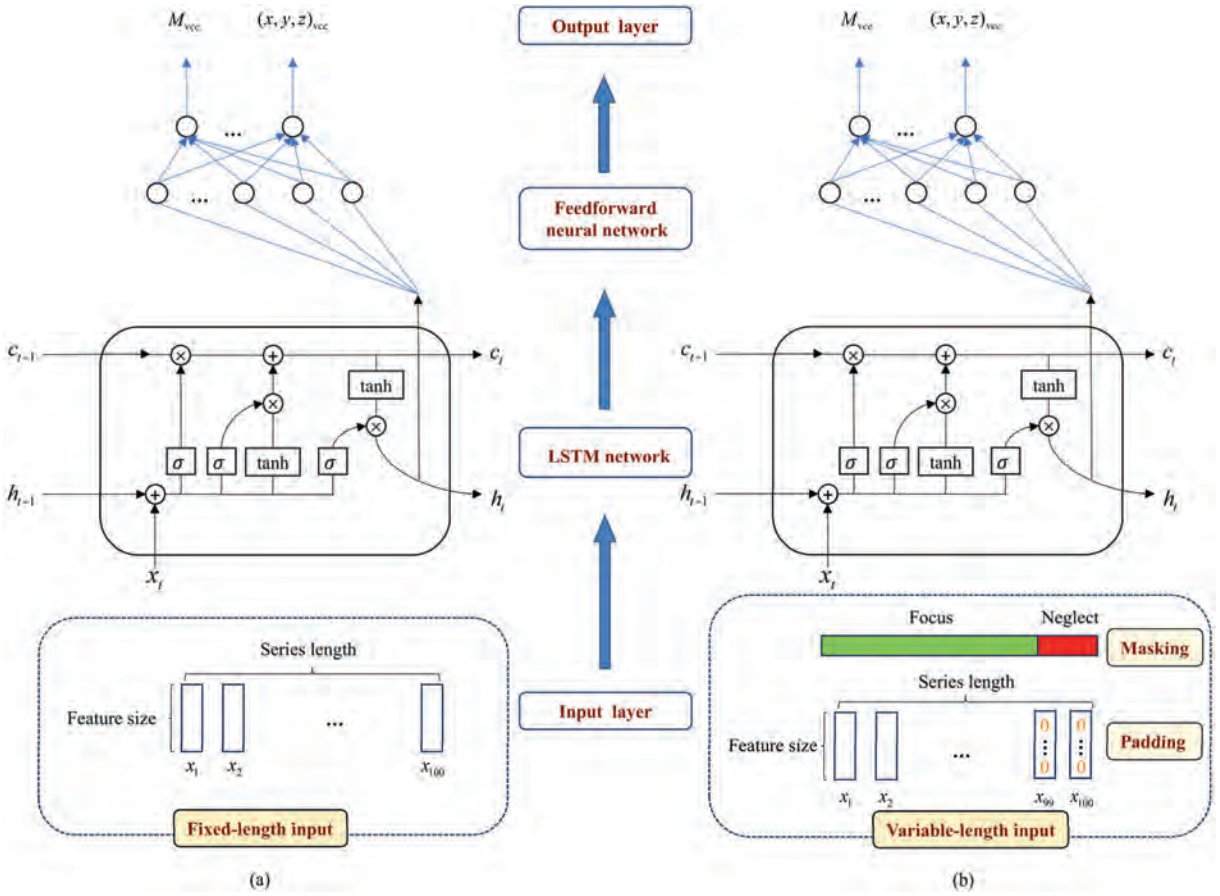

Fig. 3. Schematic diagram of the consequence prediction model: (a) fixed-length input (b) variable-length input.

Table 3
Different prediction targets and corresponding neurons in the output layers.

Target No.	Neurons	Target description
1	1	M_{vce} of methane
2	3	$(x,y,z)_{vce}$ of methane
3	1	M_{vce} of hydrogen
4	3	$(x,y,z)_{vce}$ of hydrogen
5	2	M_{vce} of both methane and hydrogen
6	6	$(x,y,z)_{vce}$ of both methane and hydrogen
7	4	M_{vce} and $(x,y,z)_{vce}$ of methane
8	4	M_{vce} and $(x,y,z)_{vce}$ of hydrogen
9	8	M_{vce} and $(x,y,z)_{vce}$ of both methane and hydrogen

Table 4
Summary of hyperparameter values.

Hyperparameters	Values
Batch size	128
Epochs	500
LSTM units	64
Learning rate	0.001
Loss function	Mean squared error
Activation function	Sigmoid, tanh
Optimizer	Adam

cloud formed by different gases. There are nine different prediction targets in total, as detailed in Table 3.

4. Experiments and Analysis

In this study, the construction, training, and prediction of the experimental model are implemented using the Python Keras library 2.4.0 with only CPU. The CPU configuration of the running platform is Intel(R) Xeon(R) Gold 6242R @ 3.10 GHz. To clearly demonstrate the selection of hyperparameters, we have summarized the values of hyperparameters used in the experiments, as shown in Table 4. Since this is a regression task, the evaluation metric for testing the model is R^2, also known as the coefficient of determination, which measures how well the model fits the observed data. The value of R^2 ranges from 0 to 1, with values closer to 1 indicating a better fit and values closer to 0 indicating a poorer fit. The formula for calculating R^2 is as follows:

$$R^2 = 1 - \frac{\sum_{i=1}^{n} (y_i - \widehat{y}_i)^2}{\sum_{i=1}^{n} (y_i - \overline{y})^2} \tag{12}$$

where y_i is the i-th target value, $\widehat{y}_i$ is the i-th predicted value, $\overline{y}$ is the mean of the target value, n is the number of samples.

4.1. Consequence prediction using fixed-length time series data

A 2-s sampling interval was selected as it balances computational cost with accurately capturing the overall concentration trend. This study experimented with six different fixed-length concentration sequences: 15, 30, 40, 50, 60, and 100 s. For example, the sequence length of 15 s corresponds to data from 0 to 30 s, sampled every 2 s. For each of the nine prediction targets, a total of 54 experimental tests were conducted. Table 5 presents the R^2 values for each prediction target at different fixed sequence lengths. The best results for each target are highlighted in bold, representing the optimal performance. It is evident that when the sequence length is set to 40 s, the model achieves near-optimal prediction performance across most targets. However, for shorter sequences (15 s) and longer sequences (100 s), the model's predictive performance is significantly lower, indicating less effective results for these durations.

Table 5
Prediction results of different fixed-length inputs across various prediction targets.

Target No.	15 s	30 s	40 s	50 s	60 s	100 s
1	0.894	0.984	**0.999**	0.997	**0.999**	0.991
2	0.889	0.917	**0.926**	0.873	0.859	0.856
3	0.981	**0.994**	0.984	**0.994**	0.970	0.991
4	0.874	0.946	**0.953**	0.906	0.891	0.878
5	0.914	0.979	0.990	0.991	**0.992**	0.986
6	0.868	0.921	**0.923**	0.890	0.867	0.876
7	0.889	0.934	0.935	**0.943**	0.925	0.885
8	0.878	0.941	**0.957**	0.932	0.925	0.883
9	0.870	0.929	**0.944**	0.909	0.893	0.882

For the 15-s length, the number of features is minimal, thus providing the least amount of useful information. Additionally, it is at the early stage of the leak, far from the steady-state concentration period, resulting in poor performance. For the 100-s concentration sequence, data features may introduce redundancy, affecting the model's judgment. Furthermore, LSTM faces challenges when dealing with very long sequences due to issues such as vanishing gradients, which can hinder its ability to maintain information over extended time spans, resulting in poorer prediction performance for the 100-s sequence.

Regarding different prediction targets, the prediction accuracy for the vapor cloud mass is better than that for the vapor cloud centroid coordinates. This is because the model's input is concentration features, and the vapor cloud mass can be calculated directly from the concentration values. In contrast, predicting the vapor cloud centroid coordinates also depends on the coordinates of each grid node, which are not included in the network's input, leading to less accurate predictions.

In terms of computational cost, slight differences arise due to the varying number of neurons in the output layer of the neural network for different prediction targets. Taking the 30-s input model designed for coordinate prediction as an illustration, it demonstrates an inference time of 0.3229 s on the test set. The model comprises 21955 parameters, which amounts to a total of 286 kB in size. These results demonstrate that the consequence prediction model proposed in this paper is capable of meeting practical requirements for rapid response and is well-suited for deployment.

4.2. Consequence prediction using variable-length time series data

The variable-length series data are padded to a length of 100. Considering all possible sequence lengths, a sequence length of 100 expands the total sample size from 89 to 8900. Table 6 shows the R^2 results of the variable-length sequence prediction model. It can be observed that the model performs well and consistently across different prediction targets.

When comparing the accuracy of the variable-length sequence prediction model to that of the fixed-length sequence prediction models, it is found that the accuracy of the variable-length model is almost always higher than the 15-s and 100-s fixed-length

Table 6
Prediction results of variable-length input across various prediction targets.

Target No.	R^2	Target No.	R^2
1	0.967	6	0.979
2	0.979	7	0.974
3	0.979	8	0.978
4	0.979	9	0.976
5	0.973		

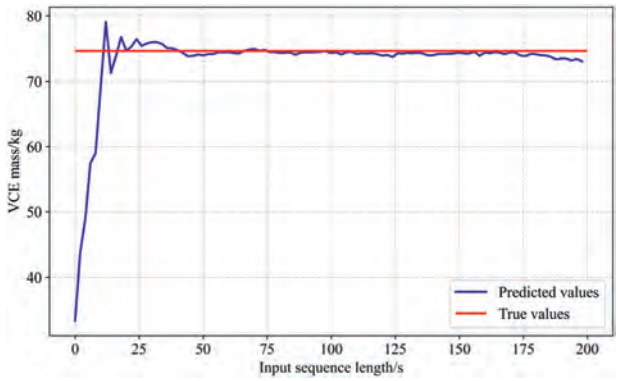

Fig. 4. Mass prediction across different input lengths.

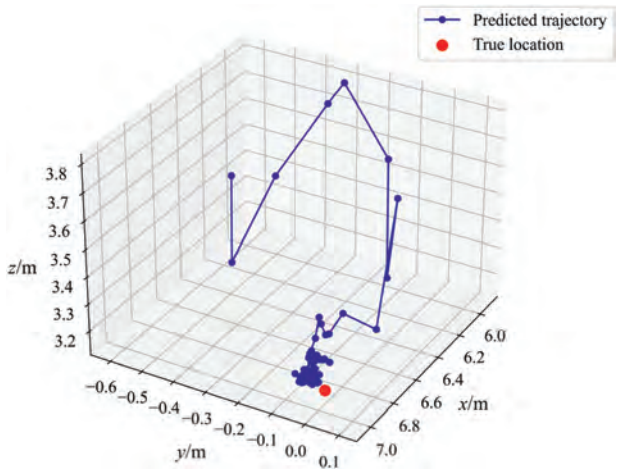

Fig. 5. The predicted trajectory of centroid location.

models, but slightly lower than those of the 30-s and 60-s models. This is because the variable-length model learns the features of sequences of different lengths during training, thereby also learning the features of the 30-s and 60-s sequences, which leads to better results compared to the lower-accuracy fixed-length sequences. However, because it encompasses features from all sequence lengths, it has a certain degree of redundancy, making it slightly less accurate than the higher-accuracy fixed-length sequences.

Figs. 4 and 5 respectively show the dynamic prediction results of the vapor cloud mass and centroid position in a single scenario as the effective length varies until the maximum input length. It can be seen that in the early stages, due to the limited number of effective concentration features, the predictions differ significantly from the true values. However, as the effective length increases, the predictions gradually approach the true values. This result can assist decision-makers in performing consequence analysis using minimal effective concentration information while maintaining accuracy.

4.3. Impact of the train-test split ratio

The split ratio between the training set and the test set can influence the accuracy of the model's performance. In this subsection, we conducted fixed-length input experiments to test the impact of three different ratios with a 30-s input: 9:1, 8:2, and 7:3. To partition the dataset, a 9:1 random split was first applied to create the training and testing sets. Then, a portion of the training set was randomly selected and added to the testing set to form an 8:2 split, and the process was repeated to achieve a 7:3 ratio. This approach was designed to reduce the randomness in the testing set across different splits by gradually increasing the size of the testing set, rather than fully re-randomizing it at each step. The experimental results are detailed in Table 7. As shown in Table 7, the 9:1 ratio produced the best results, followed by 8:2, with the 7:3 ratio yielding the worst performance. The larger training set in the 9:1 ratio allows the model to access more data and gain more opportunities to learn features from the data, leading to the best performance. However, the differences between the ratios are not significant, indicating that the model has already captured most of the features. Additionally, the R^2 values in the 7:3 case are mostly above 0.9, further demonstrating the model's strong generalization ability.

Table 7
Prediction results for different train-test split ratios.

Target No.	9:1	8:2	7:3
1	0.984	0.968	0.911
2	0.917	0.865	0.901
3	0.994	0.964	0.943
4	0.946	0.930	0.925
5	0.979	0.952	0.923
6	0.921	0.906	0.899
7	0.934	0.888	0.888
8	0.941	0.932	0.941
9	0.929	0.886	0.877

5. Conclusions

In this paper, we propose a deep learning-based method to predict the mass and centroid location of the vapor cloud explosion. Eighty-nine different CFD leakage simulations were conducted to provide experimental data. In this study, deep learning prediction models were established for both fixed-length sequence inputs and variable-length sequence inputs, based on various consequence prediction targets. The experimental results demonstrate the effectiveness of the proposed method. Additionally, the variable-length sequence model can adapt to inputs of different effective lengths for dynamic prediction. Decision-makers can balance the available sensor data and accuracy to make timely decisions, which has practical significance.

However, this study has certain limitations that warrant further investigation: (1) The sensors are strategically placed in areas where vapor clouds are likely to accumulate, the optimal number and placement of sensors are not extensively explored. The current setup may include redundant sensors or overlook the potential for more efficient configurations that could enhance prediction accuracy. Future research would focus on effective feature selection methods to identify the most influential sensors and determine the minimum number of sensors required for accurate predictions. (2) The study relies entirely on synthetic data from FDS simulations. While this serves as a reasonable proof-of-concept, it limits the generalizability of the findings to real-world scenarios. We intend to incorporate empirical sensor data from experimental or field studies for a comprehensive case study or comparative analysis in future work. (3) The proposed method may not be applicable to sensor failure scenarios that could occur in practice, potentially disrupting the regular time series required for LSTM processing. We will explore methods from the domain of irregular multivariate time series to solve this problem.

CRediT Authorship Contribution Statement

Shikuan Chen: Writing − original draft, Visualization, Methodology, Formal analysis, Data curation, Conceptualization. Wenli Du: Writing − review & editing, Supervision, Funding acquisition, Formal analysis. Chenxi Cao: Writing − review & editing, Data curation. Bing Wang: Writing − review & editing, Software.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work was supported by the National Key Research and Development Program of China (2022YFB3305900), National Natural Science Foundation of China (62373153,62173147), the Programme of Introducing Talents of Discipline to Universities (the 111 Project) under Grant B17017 and Fundamental Research Funds for the Central Universities (222202517006).

References

[1] E. Palazzi, F. Currò, B. Fabiano, A critical approach to safety equipment and emergency time evaluation based on actual information from the bhopal gas tragedy, *Process*, *Saf. Environ. Prot.* 97 (2015) 37−48.
[2] J.E. Ten Hoeve, M.Z. Jacobson, Worldwide health effects of the Fukushima Daiichi nuclear accident, *Energy Environ. Sci.* 5 (9) (2012) 8743−8757.
[3] X. Luo, X.Y. Feng, X. Ji, Y.G. Dang, L. Zhou, K.X. Bi, Y.Y. Dai, Extraction and analysis of risk factors from Chinese chemical accident reports, *Chin. J. Chem. Eng.* 61 (2023) 68−81.
[4] S. Lyu, S.H. Zhang, X.M. Huang, S.N. Peng, J. Li, Investigation and modeling of the LPG tank truck accident in Wenling, China, *Process*, *Saf. Environ. Prot.* 157 (2022) 493−508.
[5] M. Hutchinson, H. Oh, W.H. Chen, A review of source term estimation methods for atmospheric dispersion events using static or mobile sensors, *Inf. Fusion* 36 (2017) 130−148.
[6] D. He, W.P. Yan, Influences of different diluents on ignition delay of syngas at gas turbine conditions: a numerical study, *Chin. J. Chem. Eng.* 25 (1) (2017) 79−88.
[7] A.M. Idris, R. Rusli, M.E. Mohamed, A.F. Ramli, M.S. Nasif, J.S. Lim, Explosion pressure and duration prediction using machine learning: a comparative study using classical models with Adam-optimized neural network, *Can. J. Chem. Eng.* 103 (1) (2025) 137−152.
[8] A. Filice, M. Mynarz, R. Zinno, Experimental and empirical study for prediction of blast loads, *Appl. Sci.* 12 (5) (2022) 2691.
[9] A.C. van den Berg, The multi-energy method A framework for vapour cloud explosion blast prediction, *J. Hazard. Mater.* 12 (1) (1985) 1−10.
[10] R.I. Tsukada, D.A. Shiguemoto, S.S.V. Vianna, The TNO multi-energy method combined to mathematical programming and computational fluid dynamics for optimisation of gas detectors, *J. Loss Prev. Process. Ind.* 83 (2023) 105035.
[11] H. Zhou, C. Zheng, X.S. Yue, Z.H. Zhu, A.G. Lu, X.S. Kong, W.G. Wu, TNT equivalency method in confined space based on steel plate deformation, *Int. J. Impact Eng.* 178 (2023) 104587.
[12] W.F. Xiao, M. Andrae, N. Gebbeken, Air blast TNT equivalence factors of high explosive material PETN for bare charges, *J. Hazard. Mater.* 377 (2019) 152−162.
[13] R.A. Strehlow, W.E. Baker, The characterization and evaluation of accidental explosions, *Prog. Energy Combust. Sci.* 2 (1) (1976) 27−60.
[14] X. Rocourt, I. Sochet, B. Pellegrinelli, Application of the TNO multi-energy and Baker-Strehlow-Tang methods to predict hydrogen explosion effects from small-scale experiments, *J. Loss Prev. Process. Ind.* 81 (2023) 104976.
[15] P. Lei, Z. Qi, Influence of vapor cloud shape on temperature field of unconfined vapor cloud explosion, *Chin. J. Chem. Eng.* 18 (1) (2010) 164−169.
[16] Z.R. Lin, S.F. Wang, S.X. Fu, J.P. Huo, Numerical study on effects of the cofferdam area in liquefied natural gas storage tank on the leakage and diffusion characteristics of natural gas, *Chin. J. Chem. Eng.* 29 (2021) 228−241.
[17] N.H. Alam, E. Kashi, R. Habibpour, Computational fluid dynamics simulation of gas dispersion in complex facilities using Kit Fox field experiments: validation and statistical evaluation, *Chin. J. Chem. Eng.* 44 (2022) 412−423.
[18] O.R. Hansen, E.S. Hansen, CFD-modelling of large-scale LH2 release and explosion experiments, *Process*, *Saf. Environ. Prot.* 174 (2023) 376−390.
[19] G. Momferatos, S.G. Giannissi, I.C. Tolias, A.G. Venetsanos, A. Vlyssides, N. Markatos, Vapor cloud explosions in various types of confined environments: CFD analysis and model validation, *J. Loss Prev. Process. Ind.* 75 (2022) 104681.
[20] M. Dhiman, A. Zambare, P. Sathiah, V.D. Narasimhamurthy, CFD simulations of vapour cloud explosions using PDRFoam, *J. Loss Prev. Process. Ind.* 85 (2023) 105164.
[21] D.L. Ma, Z.X. Zhang, Contaminant dispersion prediction and source estimation with integrated Gaussian-machine learning network model for point source emission in atmosphere, *J. Hazard. Mater.* 311 (2016) 237−245.
[22] K. Zhang, K. Zhang, R. Bao, Prediction of gas explosion pressures: a machine learning algorithm based on KPCA and an optimized LSSVM, *J. Loss Prev. Process. Ind.* 83 (2023) 105082.
[23] Q.L. Wang, J.Q. Han, F. Chen, X. Zhang, C. Yun, Z. Dou, T.J. Yan, G.A. Yang, Real-time risk prediction of chemical processes based on attention-based Bi-LSTM, *Chin. J. Chem. Eng.* 75 (2024) 131−141.
[24] Y. Xu, Y.M. Huang, G.W. Ma, A beetle antennae search improved BP neural network model for predicting multi-factor-based gas explosion pressures, *J. Loss Prev. Process. Ind.* 65 (2020) 104117.
[25] S.N. Zhou, Z.Q. Wang, Q.Z. Li, A fusing NS with NN model for the consequence prediction of vapor cloud explosion, *Process*, *Saf. Environ. Prot.* 149 (2021) 698−710.
[26] Y.M. Bai, S.Y. Xiang, F.F. Cheng, J.S. Zhao, A dynamic-inner LSTM prediction method for key alarm variables forecasting in chemical process, *Chin. J. Chem. Eng.* 55 (2023) 266−276.
[27] S.W. Xiong, L. Zhou, Y.Y. Dai, X. Ji, Attention-based long short-term memory fully convolutional network for chemical process fault diagnosis, *Chin. J. Chem. Eng.* 56 (2023) 1−14.
[28] I. Sutskever, O. Vinyals, Q.V. Le, Sequence to sequence learning with neural networks, *Adv. Neural Inf. Process. Syst.* 4 (2014) 3104−3112.
[29] D. Bahdanau, K. Cho, Y. Bengio, Neural machine translation by jointly learning to align and translate, in: ICLR 2015, San Diego, CA, USA, 2015.
[30] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, in: Proceedings of the 31st International Conference on Neural Information Processing Systems, Long Beach California, USA, 2017.
[31] Z.L. Yang, Z.H. Dai, Y.M. Yang, J. Carbonell, R. Salakhutdinov, Q.V. Le, XLNet: Generalized autoregressive pretraining for language understanding, In: Conference on Neural Information Processing Systems, Vancouver, Canada, 2019.
[32] Y. Mouilleau, A. Champassith, CFD simulations of atmospheric gas dispersion using the Fire Dynamics Simulator (FDS), *J. Loss Prev. Process. Ind.* 22 (3) (2009) 316−323.
[33] T.S. Shen, Y.H. Huang, S.W. Chien, Using fire dynamic simulation (FDS) to reconstruct an arson fire scene, *Build. Environ.* 43 (6) (2008) 1036−1045.
[34] J. Wahlqvist, P. van Hees, Validation of FDS for large-scale well-confined mechanically ventilated fire scenarios with emphasis on predicting ventilation system behavior, *Fire Saf. J.* 62 (2013) 102−114.
[35] J.F. Chen, J.J. Liu, X.L. Tian, L. Zhang, H.H. Cheng, M.H. Zhong, Study on the effect of obstacles on smoke diffusion and airflow structure in subway stations, *Build. Environ.* 242 (2023) 110553.
[36] S. Raja Vinay, S.M. Tauseef, S. Varadharajan, Investigating the impact of oxygen concentration on fire dynamics using numerical simulation with FDS, *Process*, *Saf. Environ. Prot.* 178 (2023) 195−203.