ARTICLE IN PRESS

Engineering xxx (xxxx) xxx

Contents lists available at ScienceDirect

Engineering

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

Research
Applications of ChatGPT—Article

Integrating Smart Fire Forecast with LLM-Powered Emergency Response

Weikang Xie [a], Yuxin Zhang [a,*], Tong Lu [a], Xianjia Huang [b], Jihao Shi [a,*], Xinyan Huang [a,*], Fu Xiao [a], Asif Usmani [a]

[a] Department of Building Environment and Energy Engineering, The Hong Kong Polytechnic University, Hong Kong 999077, China
[b] Guangzhou Institute of Industrial Technology, Guangzhou 511458, China

A R T I C L E I N F O

Article history:
Received 11 May 2025
Revised 13 January 2026
Accepted 12 February 2026
Available online xxxx

Keywords:
Intelligent building
Smart firefighting
Lost data remediation
Deep learning
Large language model
Self-driven agent

A B S T R A C T

Existing data-driven fire forecast systems often exhibit limitations in real-world emergency response scenarios, particularly with respect to efficient data reuse and vulnerability of sensor networks. This study proposes a smart agent that integrates an artificial intelligence (AI)-driven fire situational awareness engine with a large language model (LLM) to realize the diverse demands of emergency response in complex fire scenarios. First, a fire-resilient deep learning model based on ConvLSTM is developed to reconstruct building temperature fields using limited inputs from a partially failed temperature sensor network. The proposed architecture constructs spatiotemporal correlations between missing and survived sensor data, enabling the transformation of discrete temperature measurements into a continuous two-dimensional (2D) temperature contour. Subsequently, a smart agent powered by a domain-specific LLM is designed to enhance human–AI interaction during fire emergency response. A self-driven framework capable of automatically executing LLM-generated programs is established to deliver real-time, user-specific information to multiple stakeholders. Experimental results demonstrate that, compared with generic LLM-based responses, the proposed agent augmented with fire situational awareness can generate customized operational recommendations through dynamic interactions with the ConvLSTM-based fire model. This hybrid agent improves situational awareness and safety during fire emergencies, improves the resilience of fire services systems, and advances the practical implementation of AI-driven smart firefighting.

1. Introduction

In recent years, the concept of smart cities, characterized by technological innovation and sustainable development, has attracted increasing attention [1]. Meanwhile, rapid urbanization has intensified the frequency and impact of disasters, posing significant challenges to sustainable urban development. Consequently, enhancing disaster resilience has become essential to improving a city's safety, livability, and sustainability [2]. Among various urban hazards, fire remains a significant and persistent risk owing to its high occurrence rate and severe consequences worldwide [3]. Furthermore, increasing population density has significantly increased the likelihood and complexity of urban fire incidents.

In hazardous fire scenarios, both trapped occupants and on-duty firefighters are exposed to highly unpredictable fire behavior, as tragically exemplified by the 2017 Grenfell Tower fire in London (Fig. 1(a)). In parallel, unchecked urban expansion into wildland areas has exposed massive communities to increasingly severe wildfires, a trend further exacerbated by climate change, as illustrated by the 2025 Los Angeles wildfire (Fig. 1(b)). Beyond the tragic loss of life, the economic consequences of fire-related incidents are considerable, with global fire-related costs estimated at approximately 1% of the global gross domestic product (GDP) annually [4]. Post-incident investigation of major fire disasters consistently reveals deficiencies in emergency response, usually stemming from suboptimal decision-making caused by insufficient situational information from the incident location [5]. For instance, two firefighters lost their lives while combating a mini-storage facility fire in Hong Kong (Fig. 1(c)) [6]. These catastrophic events underscore the limitations of traditional firefighting strategies, which frequently fail to support effective emergency

* Corresponding authors.
E-mail addresses: yx.zhang@polyu.edu.hk (Y. Zhang), jihao.shi@polyu.edu.hk (J. Shi), xy.huang@polyu.edu.hk (X. Huang).

https://doi.org/10.1016/j.eng.2026.02.023
2095-8099/

Please cite this article as: W. Xie, Y. Zhang, T. Lu et al., Integrating Smart Fire Forecast with LLM-Powered Emergency Response, Engineering, https://doi.org/10.1016/j.eng.2026.02.023

Fig. 1. (a) The 2017 London Grenfell Tower Fire, (b) the 2025 LA fire, and (c) the 2016 HK mini-storage fire.

decision-making owing to delayed information acquisition and inadequate situational awareness [7].

The effectiveness of emergency response critically depends on situational awareness, which refers to the accurate, real-time perception of environmental conditions through integrated hardware and software systems [8], such as the spatiotemporal evolution of temperature fields. Contemporary smart city infrastructures, supported by Internet of Things (IoT) networks, enable the generation and sharing of data for rapid sensing of the fire environment [9]. Meanwhile, artificial intelligence (AI) has been increasingly employed in urban disaster management owing to its advantages in accuracy and real-time processing capabilities [10]. Nevertheless, significant challenges remain in deploying existing AI-driven emergency response systems in realistic firefighting scenarios based on on-site fire information. First, disaster environments are inherently destructive, imposing stringent requirements on the robustness and integrity of AI-driven situational awareness systems. In addition, conventional single-purpose AI models typically produce outputs in fixed and predetermined formats, which limits their ability to support real-time re-processing or accommodate the diverse needs of different users during emergency response [11].

Currently, the emergence of large language models (LLMs) demonstrates their potential to serve as a foundational technology for generic smart firefighting systems capable of delivering real-time support. LLMs have significantly reduced barriers in human–computer interaction (HCI) [12] by enabling natural language understanding and generating context-aware responses to diverse user queries [13]. However, most existing domain-specific LLMs are developed through fine-tuning and retrieval-augmented generation (RAG) approaches that primarily rely solely on static, general-domain knowledge, such as laws and regulations [14]. The absence of dynamic, real-time fire scene information limits the ability of LLMs to generate actionable guidance involving specific objects and situational constraints. Consequently, retrieval mechanisms that enable real-time fire situational awareness are a crucial prerequisite for effective LLM-driven emergency response systems. Moreover, knowledge-optimized LLMs typically function as pre-defined query assistants and lack direct integration with executable local tools, which restricts their capacity to substantially enhance emergency response efficiency. To align firefighting engineering with contemporary technological advancements, there is an urgent demand for a novel emergency response framework that can deliver customized recommendations and automatically execute operational procedures tailored to specific fire scenarios.

This study proposes a self-driven smart agent for emergency response based on fire situational awareness, integrating AI-generated domain-specific knowledge with an LLM to support dynamic user requirements, including decision generation and mission execution. Resilient building fire information is obtained using a deep learning model (ConvLSTM-Fire), which predicts the spatiotemporal patterns of building fires even in the presence of single-sensor failures. Within an LLM-centered framework, the proposed agent can autonomously perform sophisticated tasks, such as emergency decision-making informed by real-time situational awareness. By bridging the interaction gap between AI and human users during fire emergencies, the proposed framework enhances the feasibility and reliability of intelligent firefighting systems in future smart buildings and smart cities.

2. Methodology

2.1. Framework of the hybrid fire-emergency agent

Although agent-based approaches have been explored in industrial management systems [15,16], conventional workflow-based approaches are insufficient to cope with dynamic emergency response scenarios. Key stakeholders at the fire scene, including trapped occupants, firefighters, and commanders, require timely and accurate information such as hazardous area identification, advanced fire behavior prediction, and evacuation routes planning. Limited response time makes it difficult for a general fire-emergency agent to address diverse and rapidly changing emergency demands.

Developments in computer technology have laid the foundation for the application of emergency response agents in disaster management. On the one hand, by integrating diverse pre-trained AI models or surrogate models, existing AI-driven systems provide multi-scale disaster situational awareness, ranging from compartments [17] and complex buildings [18] to citywide environments [19]. On the other hand, LLMs have significant potential to redefine smart emergency response [20]. With their strong contextual understanding, LLMs can serve as the core of agents designed to develop more efficient emergency response strategies [21]. Table 1 compares the characteristics of two AI techniques, pre-trained deep learning models and LLMs, in emergency response scenarios.

For pre-trained AI models, results close to real solutions for complex problems are obtained through multi-parameter fitting. Based on simulation and experimental methods, the dynamics of fire development, emergency response, and evacuation have been extensively modeled [22–24]. Using knowledge datasets, various deep learning models have been developed for specific fire emergency response tasks, such as backdraft forecast [25], risk analysis [26], and evacuation guideline [27]. However, deep learning models designed for specialized objectives often exhibit limitations in real-world applications, particularly in terms of invocation flexibility and data reusability. Compared with LLMs, the strict requirements on data formats and communication protocols make

Table 1
Suitability comparison of pre-trained models and large models for emergency response scenarios.

Performance	Pre-trained AI models	Large language models
Generalization capability	Low (specialized tasks)	**High** (comprehensive tasks)
Input requirement	High	**Low**
Output accuracy	**High**	Moderate (potential logic errors)
Interactivity	Moderate (fixed command)	**High** (natural language)
Deployment costs	**Low**	High
Knowledge update cost	High (retraining and deployment)	**Low** (context-based training)
Multi-device collaboration	Moderate (custom protocols)	**High** (command generation)

pre-trained AI models unsuitable for building user-friendly interaction mechanisms. Moreover, panic during fire incidents may lead to irrational behavior [28], preventing users from effectively operating such systems. As a result, smart agents relying solely on a single deep learning engine are often incapable of meeting complex human demands [29], especially in fire emergency scenarios.

LLMs demonstrate competitive interactivity, owing to natural language processing capabilities and extensive data memory. However, one of the key challenges in LLM-driven emergency response is that their outputs are often generic and weakly actionable [30,31]. Even when optimized using normative references such as codes and standards, their inherently scenario-agnostic nature can result in insufficient guidance or even hallucinatory responses [32]. Fire situational awareness provides essential real-time information support emergency decision-making. Therefore, it is necessary for generic LLMs to incorporate deep learning-based situational awareness models to establish smart agents suitable for building fire emergency response.

Fig. 2 presents the framework of a hybrid AI-driven fire-emergency agent that provides demanders with scenario-specific guidance within a limited response time through natural language interaction. Using wearable or handheld devices, the cloud-based LLM can immediately deliver foundational knowledge or general safety advice. When handling user queries, the agent rapidly fuses multi-source data, which is organized into structured data. Both the user query and the structured environmental data are then fed into the LLM to condition its reasoning process. The LLM output is constrained to generate actionable emergency decisions under an "evidence-based content generation" requirement. In this manner, the agent can invoke appropriate tools to mitigate hallucination and execution risks. System latency is further reduced by caching inference data and reusing contextual information, enabling timely and context-specific guidance.

This framework establishes a promising engineering paradigm for hybrid AI-driven smart firefighting, in which a domain-specific agent is developed for real-time emergency response. Enhanced by deep learning-enabled situational awareness, a well-established LLM demonstrates improved real-time decision-making capabilities for scenario-specific applications. Meanwhile, the self-driven mechanism enables the proposed agent to integrate multi-source downstream information, providing dynamic technical support for constructing a multimodal firefighting large model.

2.2. Fire prediction with data remediation

With field data sharing enabled by the Internet of Things (IoT), AI engines can predict building fires based on knowledge derived from experiments and simulations, forming an artificial intelligence of things (AIoT)-powered smart firefighting framework [18]. However, in urban fire emergencies, adverse conditions such as high temperatures, smoke, and structural damage can lead to sensor data loss [18,33,34]. Data-driven systems become vulnerable when even a single sensor in the IoT network is compromised, significantly reducing the reliability of AI applications in disaster scenarios. The inherent nonlinearity of fire dynamics and the non-random nature of missing data further limit traditional remediation methods based on statistical approaches [35]. For example, spatial interpolation methods such as Kriging rely on smooth spatial covariance, which is typically observed in large-scale continuous phenomena, such as climate systems. However, the assumption of stationarity is unsuitable for building fire scenarios, where the thermal properties of walls induce non-linear temperature variations across different compartments [36].

Advanced deep learning-based technologies have emerged and demonstrated powerful data remediation capabilities. Temporal dependency-based remediation mechanisms, represented by long short-term memory (LSTM) networks, have been developed for this purpose [37]. However, irreversible sensor failures during fire incidents can result in sustained data loss, leading to flat predictions as errors accumulate over time [38]. Another remediation approach uses generative adversarial networks (GANs) with context encoders to infer missing data from surrounding information [39]. This approach is challenging in building fire scenarios, as sensors are often sparsely distributed across different compartments. Once a sensor fails, it becomes difficult to obtain sufficient information for the affected compartment. Therefore, a promising strategy is to leverage historical data from remaining sensors to compensate for missing measurements.

Data remediation based on historical data from remaining sensors requires a hybrid deep learning framework capable of spatiotemporal information processing [40]. This study proposed a hybrid deep learning model, ConvLSTM-Fire, which integrates convolutional neural networks (CNN) [41] for spatial features extraction with an LSTM backbone. ConvLSTM-Fire focuses on real-time data remediation using an information matrix derived from residual sensors. Mathematically, given time series data $\boldsymbol{X}$ from IoT with missing values caused by sensor damage, ConvLSTM-Fire predicts the missing data $\boldsymbol{T}_{\text{miss}}$. The mathematical formulation is given as follows:

$$\boldsymbol{X} = \begin{pmatrix} T_1^1 & \cdots & T_1^t \\ \vdots & \ddots & \vdots \\ T_N^1 & \cdots & T_N^t \end{pmatrix}, \boldsymbol{T}_{\text{miss}} = T_n^t \tag{1}$$

where T represents temperature (℃). N denotes the number of sensors. n represents the index of the failed sensor, and t represents time (s). Missing sensor data are set to 0 to complete $\boldsymbol{X}$, which is then normalized as the model input.

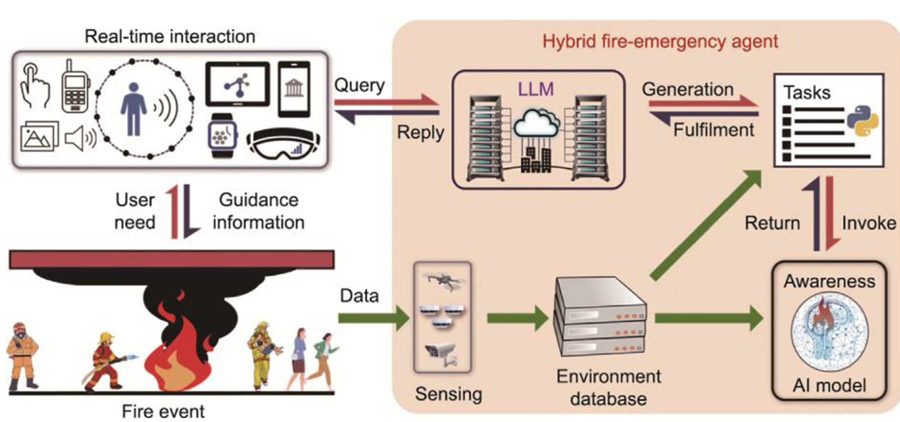

Fig. 2. Hybrid fire-emergency agent-driven emergency response in smart firefighting.

In the first step, the input data are convolved and pooled by the CNN to transform the 2D spatiotemporal information into a one-dimensional (1D) hidden representation Z. Subsequently, key temporal features H would be extracted from Z through LSTM, which is then fitted to the target $\boldsymbol{T}_{\text{miss}}$ for data remediation through the fully connected neural network (DNN). The information flow is shown as follows:

$$Z = \text{Conv}^{w_1}(\boldsymbol{X}) \tag{2}$$

$$H = \text{LSTM}^{w_2}(Z) \tag{3}$$

$$\boldsymbol{T}_{\text{miss}} = \text{DNN}^{w_3}(H) \tag{4}$$

where $w = [w_1, w_2, w_3]$ denotes the hyperparameters of the different neural network components, which are optimized during model training. The remediation missing value $\boldsymbol{T}_{\text{miss}}$ is combined with the remaining sensor measurements to form the temperature vector $\boldsymbol{T}^t$ at time t. Furthermore, a deconvolution neural network (DeConv) is applied to upscale the 1D temperature sequence $\boldsymbol{T}^t$ into a 2D temperature slice $\boldsymbol{T}_{\text{slide}}$. Further details can be found in Ref. [18].

$$\boldsymbol{T}_{\text{slide}} = \begin{pmatrix} T_{1,1} & \cdots & T_{1,D_2} \\ \vdots & \ddots & \vdots \\ T_{D_1,1} & \cdots & T_{D_1,D_2} \end{pmatrix} \tag{5}$$

where (D_1, D_2) represent a slice defined by any 2D within a three-dimensional (3D) space.

The proposed ConvLSTM-Fire model enables accurate reconstruction of data from missing sensors by leveraging historical temperature information from spatially related sensors. As a result, the sensing resilience of the IoT system in disaster environments is enhanced, ensuring the data integrity required for deep learning-based analysis. In addition, ConvLSTM-Fire predicts the 2D spatial distribution of temperature from 1D sensor data. Consequently, the resulting high-resilience fire situational awareness provides a reliable information basis for fire emergency response, supporting evacuation and firefighting rescue operations. Furthermore, the convenient invocation and functional redevelopment of deep learning models require further investigation to ensure the practical usability of pre-trained models such as ConvLSTM-Fire in real-world emergency response scenarios.

2.3. LLM-based emergency interaction

In recent years, the emergence of generative large models has created significant opportunities for industrial process intelligence [42]. For example, domain-specific LLMs can be developed by fine-tuning model parameters using dedicated databases with updated knowledge [43]. Currently, several studies have explored the application potential of optimized LLMs in emergency management [21,44,45]. Although LLM-connected chatbots provide a simple interaction interface, they struggle to execute complex tasks that require interaction with local data owing to sandbox constraints. In contrast, a smart fire-emergency agent integrates the LLM with redefined modules, such as ConvLSTM-Fire, enabling direct operation on local data with safety guardrails. Rapid deployment and flexible functional development further enhance the practical applicability of the proposed smart agent.

Moreover, pre-trained models that lack real-time data reprocessing are significantly limited in adapting to evolving user requirements. The emergence of LLMs has made automatic programming technically feasible, enabling the generation of executable code directly from natural language input from users. Representative LLMs, including ChatGPT [46], LLaMA [47], and DeepSeek [48], have demonstrated expert performance in generating real-time solutions across diverse tasks. The key mechanism underlying automatic programming lies in the pre-training and fine-tuning paradigm [49]. Unsupervised pre-training on massive datasets, followed by task-specific fine-tuning, allows models to learn complex patterns, including programming language syntax and structure [50]. The effectiveness of code-specific LLMs in Python code generation has been validated [51]. Yu et al. [52] conducted a comprehensive empirical evaluation of ChatGPT, a representative LLM-based software, focusing on its self-programming and self-verification capabilities. The results indicate that while ChatGPT is robustly programmable, it may still generate hallucinated outputs when handling complex problems.

Relying on manual execution of auto-generated code is impractical in emergency situations such as disasters; therefore, a self-driven agent is necessary. Frameworks for multi-agent system collaboration have been preliminarily developed to enhance LLM capabilities, enabling the automatic creation of workflows to accomplish complex tasks without requiring participants to possess programming skills [53,54]. However, existing LLM-driven approaches have not considered interaction with established deep learning models. When combined with well-trained deep learning models, an LLM-driven hybrid smart firefighting system can provide a unified framework to address various emergency response demands, including customized recommendations such as hazardous area identification. Accordingly, this work aims to develop a self-driven LLM-based fire emergency agent that enables efficient user interaction and delivers real-time responses to diverse emergency queries based on enhanced fire situational awareness.

3. Key elements of the hybrid self-driven fire-emergency agent

3.1. Benchmark numerical dataset

As a critical component of the proposed agent for handling task-specific operations, deep learning models are required to enhance fire situational awareness. Because the deep learning model in this study focuses on temperature slice inference under data loss conditions, conducting large-scale experiments to obtain spatial distributions is impractical. Instead, numerical datasets generated using well-validated computational fluid dynamics (CFD) simulations are employed, as they can provide sufficiently reliable data for deep learning models development [55]. In this study, fire dynamics simulator (FDS) version 6.7 is selected as the representative CFD software for building fire simulations [56]. The 3D numerical model is designed with reference to the OECD fire research program. As shown in Fig. 3(a), the building consists of four interconnected compartments. The left room and middle room share identical dimensions (length × width × high: 5 m × 6 m × 4 m), while the right room (atrium) has the same floor area but double the height (length × width × high: 5 m × 6 m × 8 m). These rooms are interconnected by a corridor with overall geometric dimensions of length × width × high: 15 m × 2.5 m × 5 m.

The project was organized in multiple experimental phases and has been widely validated, demonstrating that CFD can be used to reconstruct experimental data [55,57]. In this study, the PRS-SI-D1 experiment [58] is selected for CFD accuracy validation to ensure the reliability of the subsequent dataset. A pool fire is configured at the center of the middle room, fueled by hydrogenated tetra-propylene (TPH, $C_{12}H_{26}$), with an area of 0.4 m^2. The experimental setup includes a ventilation system, with air inlet and outlet openings located below the ceiling, each measuring 0.3 m × 0.6 m. Three thermocouple trees are installed at the centers of the northeast, southeast, and southwest corners of the room, and each tree consists of 18 thermocouples arranged with a vertical spacing of

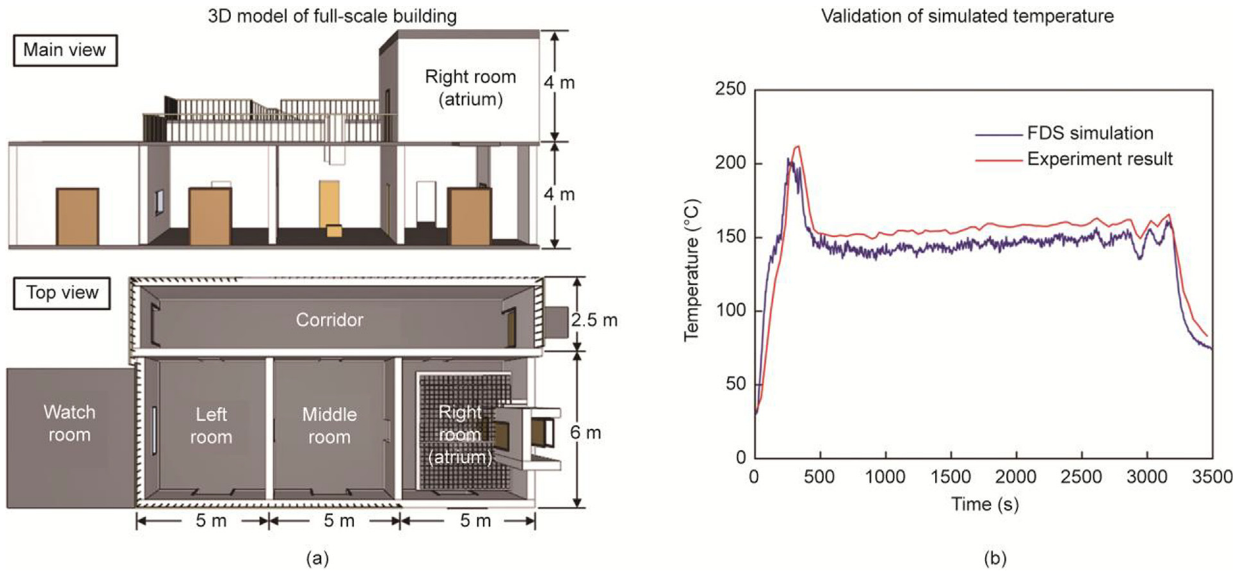

Fig. 3. 3D numerical model based on the OECD and validation using PRS-SI-D1.

0.25 m. Related studies have reported the mass loss rate as a function of time, from which the heat release rate (HRR) is extrapolated.

Fig. 3(b) compares the average temperatures obtained from the experiment and the simulation at the northeast thermocouple tree. The numerical simulation shows reliable results that align well with the experimental measurements. Initially, the temperature exhibits a clear increasing trend and reaches a peak at approximately 330 s. The temperature then begins to decrease and stabilizes at around 150 ℃ by approximately 480 s. After the experiment duration of about 3200 s, the temperature gradually decreases further. Overall, the simulation accurately reproduces the experimental phenomenon in terms of both trends and key temperature values. These results demonstrate that the numerical model robustly predicts the overall thermal evolution of the fire scenario, providing a reliable database for training the deep learning model.

With validated accuracy, CFD is employed to generate the benchmark numerical database for constructing the ConvLSTM-Fire model. A total of 60 fire scenarios is configured, including four ignition locations, three fire intensities, and five ventilation layouts based on door opening conditions. Each fire scenario lasts 2000 s, and temperature databases with spatiotemporal characteristics under different scenarios are generated, as shown in Fig. 4(a). The numerical results are pre-processed to construct the dataset $D = [\boldsymbol{X}, \boldsymbol{Y}]$ for model training. Fig. 4(b) presents temperature slices at a height of 3.8 m, incorporating the locations of six sensors. Historical temperature values recorded by the sensors are extracted as the inputs $\boldsymbol{X}$ for the ConvLSTM-Fire model. The corresponding 2D temperature slices are stored in array format as the output $\boldsymbol{Y}$ to supervise model training. With a sampling interval of 2 s, a total of 60 000 groups of the D dataset are generated. As shown in Fig. 4(c), dataset D is divided into two subsets: 80% for training and validation during model development, and the remaining 20% for testing the performance of the well-trained model.

3.2. ConvLSTM-Fire model development

Given the importance of situational awareness in fire emergency response, a building temperature inference approach that accounts for sensor vulnerability is required. This objective is specifically addressed through two steps: ① temperature field inversion based on discrete sensing data) and ② data remediation under sensor failure conditions. To achieve these goals, the ConvLSTM-Fire model is proposed to capture both the inference mechanism between sensor measurements and the building thermal environment, as well as the mapping relationships among sensors.

The architecture of ConvLSTM-Fire, summarized in Fig. 5, consists of four main components. Part 1 comprises temperature data collected from the IoT system, which serves as the input to the ConvLSTM-Fire model. The input data are formatted as 2D arrays with a shape of (6,15), representing temperature data of 6 sensors over 15 timesteps. Depending on specific requirements, the input data are processed to provide appropriate inputs for the 2nd and 3rd parts. Part 2 performs 2D temperature field $\boldsymbol{Y}^{81\times54}$ inversion

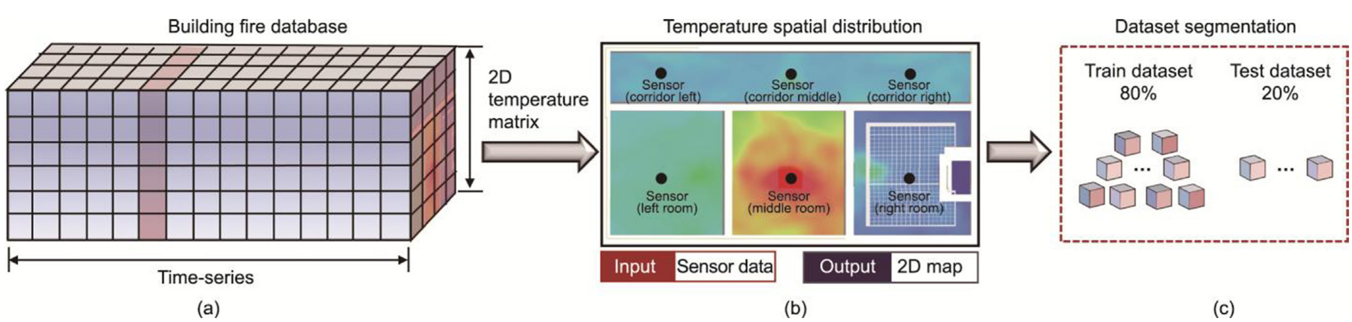

Fig. 4. Database development for ConvLSTM model.

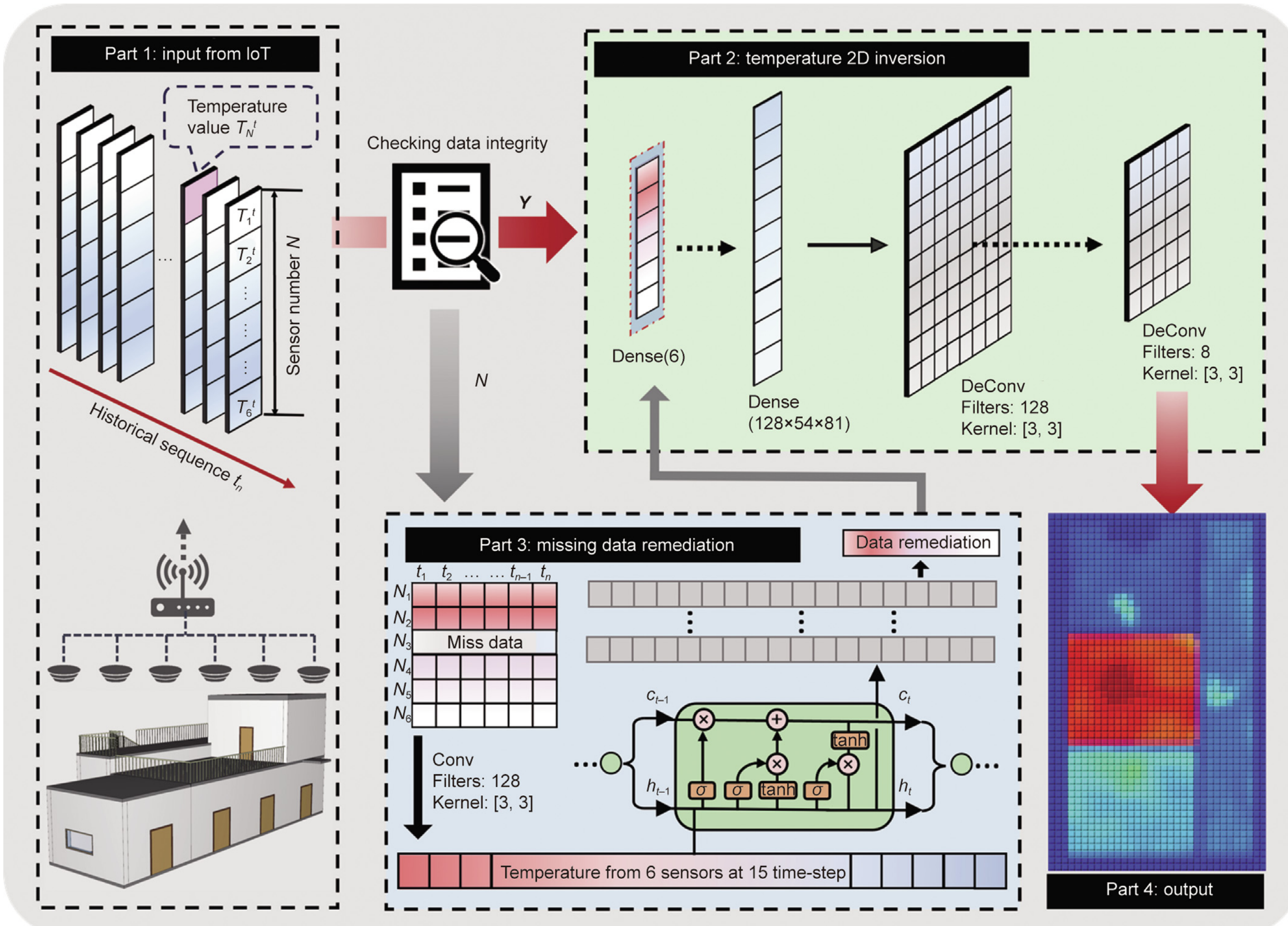

Fig. 5. Architecture of ConvLSTM-Fire model.

from sensor data $\boldsymbol{X}^6$ at time t. Through a DNN, the initial six sensor inputs are mapped to a hidden layer, which is reshaped into a 2D representation using successive DeConv layers. For damaged sensors, Part 3 performs data remediation using historical data from the remaining sensors. The input matrix $\boldsymbol{X}^{6\times15}$ is temperature measurements from six sensors during 15 time steps, in which missing or abnormal values are filled with 0. The CNN convolves this 2D input to extract features, which are reshaped into a (15, 6 × 128) representation. These features are then processed by an LSTM through two state parameters (c for cell state, h for hidden state) for temporal features extraction, and finally compressed by a DNN to estimate the temperature value of the missing sensor. Part 4 represents the output, namely, the visualization of the inferred 2D temperature distribution slide $\boldsymbol{Y}^{81\times54}$ presented to the user. The detailed configuration of Part 2 and Part 3 is provided in Table 2.

Table 2
Configuration of ConvLSTM-Fire.

Part 2 Temperature 2D inversion: DNN +DeConv	Part 3 Missing data remediation: Conv+LSTM
6 Input.↓, ReLU	(6, 15) Input.↓, ReLU
128 × 54 × 81 Dense.↓, ReLU	3 × 3 128 Conv2D ↓, ReLU
(128, 54, 81) Reshape ↓, ReLU	50 LSTM ↓, ReLU
3 × 3 128 DeConv2D ↓, ReLU	50 Dense ↓, ReLU
3 × 3 1 DeConv2D, None	1 Dense, None

Fig. 6(a) illustrates the training process of Part 2 using two key metrics: mean square error (MSE) in blue and the coefficient of determination R^2 in red. Dashed lines represent prediction performance on the training dataset D_{train}, while solid lines indicate performance on the test dataset D_{test}. During the initial training epochs, poor initial parameter fitting is observed, characterized by high MSE (> 0.20) and low R^2 (≈ 0). By epoch 20, the MSE of D_{train} decreases sharply to below 0.02. Although the MSE of D_{test} is preliminarily higher than that of D_{train}, it rapidly decreases to a comparable level. Meanwhile, the R^2 value of both D_{train} and D_{test} stabilizes at approximately 99% following a synchronized increase over similar training epochs. During subsequent epochs, both MSE curves remain low with minor fluctuations, and the R^2 curves remain high, indicating strong predictive performance for temperature field inversion based on sparse sensor data. Importantly, the strong performance on test dataset D_{test} indicates that overfitting does not occur during the training process.

A similar training process is observed in Fig. 6(b), which demonstrates rapid convergence to low error (MSE < 1 × 10^{-3}) and high explanatory power (R^2 > 97%). The close alignment of the curves for D_{train} and D_{test} highlights the robust generalization capability of the model in estimating missing sensor data. These results indicate that the proposed ConvLSTM-Fire model achieves high accuracy in inference from point measurements to spatial fields, while effectively reconstructing missing temperature values using historical sensor array data.

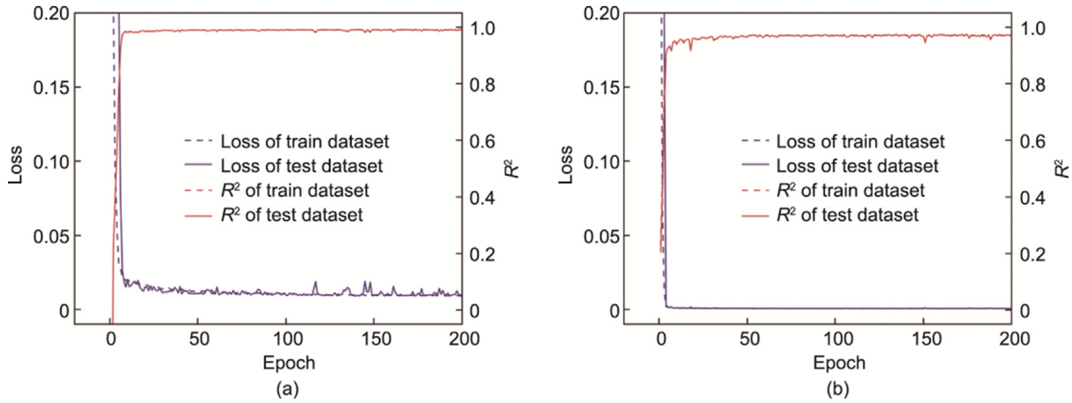

Fig. 6. Training process of ConvLSTM-Fire: (a) 2D temperature field inversion and (b) missing data remediation.

3.3. Fire-emergency agent based on hybrid AI technology

The hybrid fire-emergency agent is driven by both the well-trained ConvLSTM-Fire model and a pre-defined LLM. Fig. 7 demonstrates the specific configuration of the agent, which is designed to meet the emergency interaction demands of users. The LLM is benchmarked against ChatGPT-4o based on the Transformer architecture [59,60] and is fine-tuned with domain knowledge related to ConvLSTM-Fire, including model location, data semantics, and invocation parameters. Environmental data from sensing systems deployed within the building are synchronized with the IoT database, enabling ConvLSTM-Fire to generate real-time 2D temperature slices. When users require information support, the LLM invokes ConvLSTM-Fire to obtain up-to-date fire situational information and provide customized guidance. Unlike conventional LLMs that generate generic responses, this framework would consider the dynamic data thanks to deep learning models and sensor data, thereby delivering highly contextualized decision support during fire emergencies.

The practical effectiveness of the proposed agent arises from the synergy between a self-driven framework and LLM-based code generation. Because LLM-generated data-processing mechanisms cannot be manually executed within a limited emergency response time, a self-driven framework is integrated to perform tasks

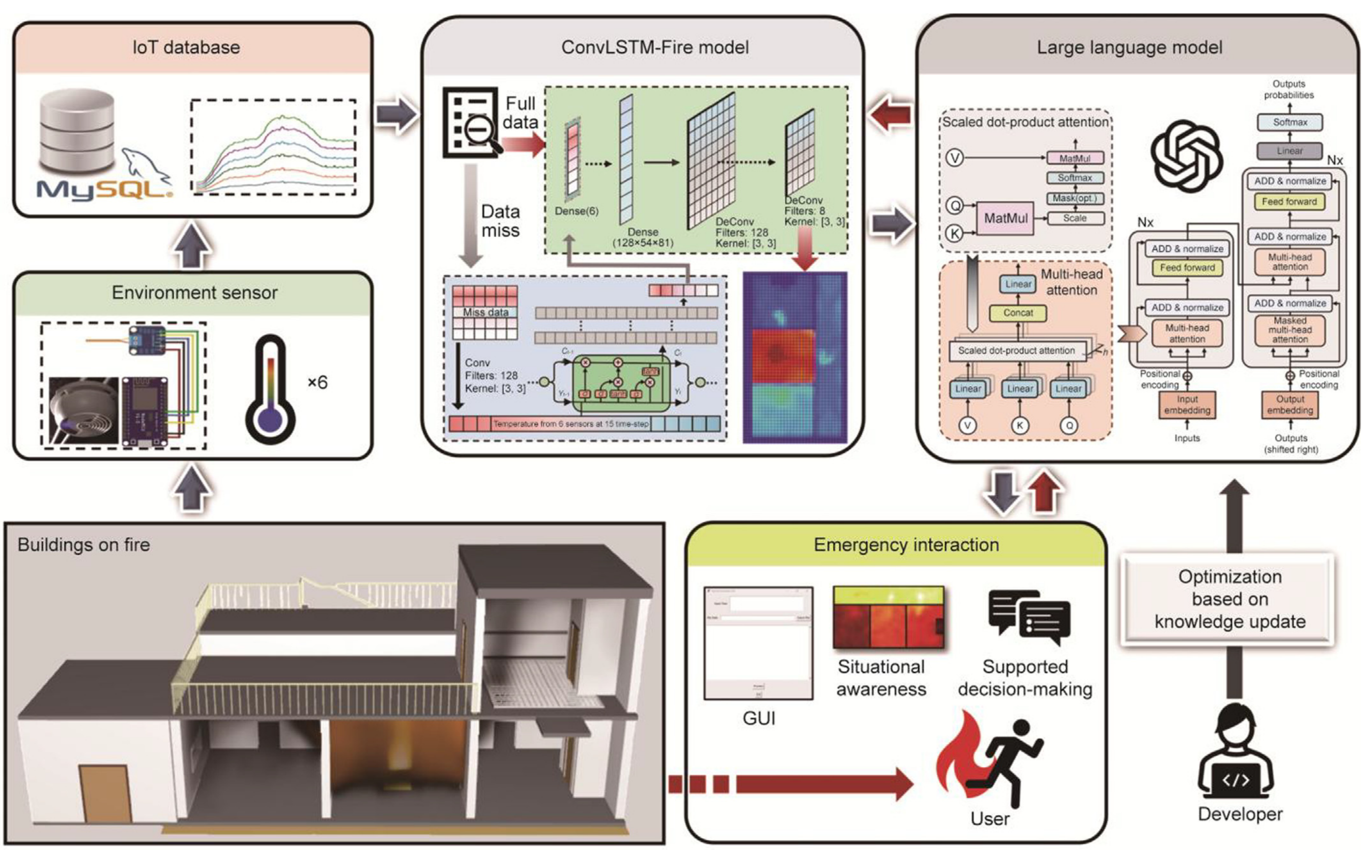

Fig. 7. Configuration of components within the hybrid fire-emergency agent. GUI: graphical user interface.

autonomously on behalf of users. Based on the text from LLM, this agent features an execution function through the invocation of tools such as Python.

Fig. 8 illustrates the architecture of the proposed self-driven agent, in which the LLM serves as the central coordinator. By integrating automated code generation with sequential task orchestration, the agent can dynamically adapt to user requirements and evolving on-site conditions. First, the LLM classifies user queries according to their dependence on scenario-specific data. General queries are addressed directly using the LLM's internal knowledge base. In contrast, scenario-dependent queries, such as those involving building temperature distributions, are decomposed into sequential tasks. For each task, the LLM automatically generates the corresponding code. To ensure safe and autonomous operation, the generated code is saved as an executable script and run within a restricted subprocess to protect core system resources. If runtime errors occur, the agent captures the error message, updates the code accordingly, and re-executes the process.

Throughout this workflow, the agent has direct access to deep learning models and sensor data. It can automatically load pretrained networks, such as the ConvLSTM-Fire model, to infer temperature distributions from sensor inputs. The resulting heat maps are stored locally in a database and encoded for further LLM analysis, enabling more sophisticated and context-specific follow-up actions. Under the self-driven architecture, these tasks are executed sequentially, and the results are delivered to on-site responders in an audio-visual, user-friendly manner via cloud services. Ultimately, the proposed hybrid fire-emergency agent establishes a closed-loop cycle in which the LLM integrates its natural language processing capabilities with real-time data and advanced deep learning inferences to provide actionable and customized recommendations to stakeholders.

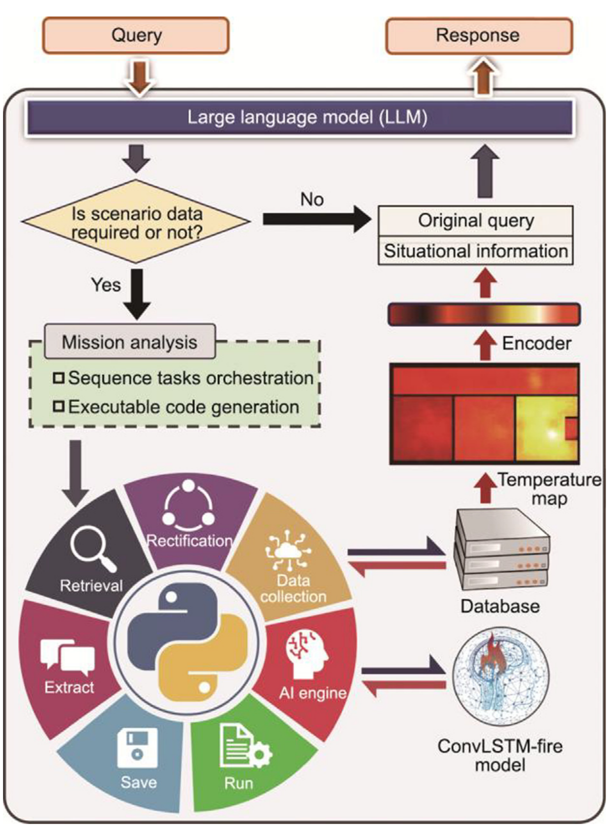

Fig. 8. LLM-based self-driven mechanism and interaction with well-trained deep learning models

4. Results and discussion

4.1. Fire situational awareness

The performance of the ConvLSTM-Fire model is demonstrated in terms of sensor data prediction and spatiotemporal inference of building-wide temperature distributions, based on numerical and experimental validation. The numerical scenario shown in Fig. 3 (a) is selected, in which the fire source is located in the middle room with a maximum heat release rate (HRR) of 2 MW·m^{-2}. To maintain continuous combustion, the remanding rooms are vented to the external atmosphere. Fig. 9 presents an accuracy validation of the ConvLSTM-Fire model for temperature time histories at six sensor locations. The sampling time period covers three different fire phases: ignition, growth, and full development. In each subplot, the predicted temperature at the sensor location is extracted from the 2D building temperature field generated by the ConvLSTM-Fire model. The predicted temperature aligns closely with the benchmark data in both trend and magnitude, although temperature fluctuations at the sensor (corridor middle) are predicted more conservatively.

This demonstrates that the AI-driven inversion effectively captures the spatial relationship between sensor measurements and the temperature distribution. Furthermore, when a given sensor is assumed to fail, the temperature of the missing sensor is inferred solely from the remaining five sensors. Fig. 9(c) shows that the remediation of the sensor (right room) initially deviates from the benchmark from 50 to 70 s but is subsequently corrected, highlighting the robustness of the proposed model. Overall, ConvLSTM-Fire exhibits strong predictive correction capability by leveraging historical data from multiple sensors. The remediated temperature curve closely aligns with the benchmark data while effectively reducing spurious fluctuations.

Fig. 10 compares the 2D spatiotemporal temperature evolution obtained from CFD simulations with that predicted by the ConvLSTM-Fire model (Video S1 in Appendix A for more details). The CFD results in the left column serve as the benchmark. The middle column illustrates the temperature field inverted by ConvLSTM-Fire using data from six sensors. The right column presents predictions assuming that the sensor (middle room) is compromised after the environmental condition exceeds its survival threshold. Temporally, the ConvLSTM-Fire model exhibits temperature evolution patterns similar to the CFD benchmark for both direct inversion and data remediation cases. A localized high-temperature region emerges in the Middle Room at approximately 60 s and expands rapidly until 120 s. Temperatures in the remaining rooms gradually increase and stabilize over time. Spatially, the model accurately captures key distribution characteristics, including distinct temperature differences among the four rooms and pronounced temperature gradients within the middle room. Localized hot spots near inter-room connections are also reproduced, indicating the invasion pathways of high-temperature gases. Overall, ConvLSTM-Fire faithfully reproduces the spatiotemporal evolution of building fire in a numerical scenario, providing reliable environmental awareness information for downstream LLM-based decision support.

To demonstrate the generalization capability and practical credibility of the proposed method, a full-scale building fire experiment was conducted at the Guangzhou Institute of Industrial Technology. Fig. 11(a) shows the experimental setup, whose structural configuration follows the PRS-SI test reference and is consistent with the ConvLSTM-Fire model. For the experimental scenario, six wireless sensors were installed according to the simulation case, and detailed information on the sensing system configuration can be found in [18]. The middle room was selected as the

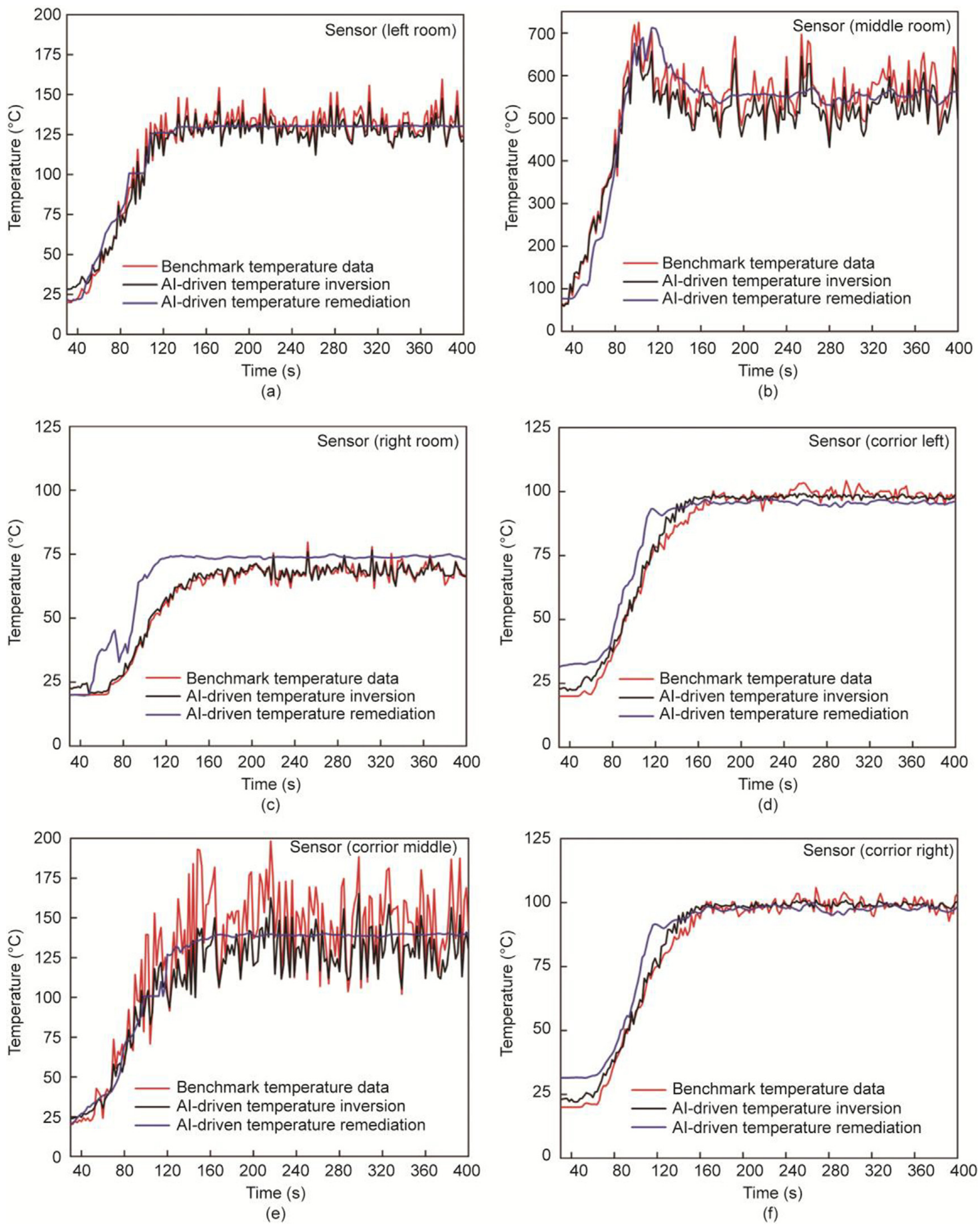

Fig. 9. Accuracy validation of ConvLSTM-Fire based on sensor data.

combustion chamber, with a gasoline pool (50 cm × 50 cm × 5 cm) placed at the center. Internal doors within the building were kept open to ensure gas circulation, while doors connecting to the outside environment were closed.

The recorded temperature–time histories are shown in Fig. 11(b). Compared with the simulated scenario, the experiment exhibits a complete combustion process, including a decay stage. For safety concerns, the fire source was constrained, resulting in a peak temperature of approximately 62 °C, which is more conservative than in the numerical case. In addition, the limited fuel supply was insufficient to sustain the peak temperature. These constraints introduce substantial differences from the numerical database, providing a suitable and challenging test case for validating the proposed method.

Fig. 12 illustrates the situational awareness of the experimental building fire inferred using the ConvLSTM-Fire model. Taking the sensor in the sensor (middle room) that exhibits the most pronounced temperature variation as reference, Fig. 12(a) compares the experimental measurements with the ConvLSTM-Fire outputs. The temperature values inferred through field inversion indicate a more dangerous prediction. The remediated data under sensor failure conditions provide a more conservative estimate with a modest delay in peak timing. Nevertheless, both AI-driven curves reproduce similar variation patterns, such as the rapid decline

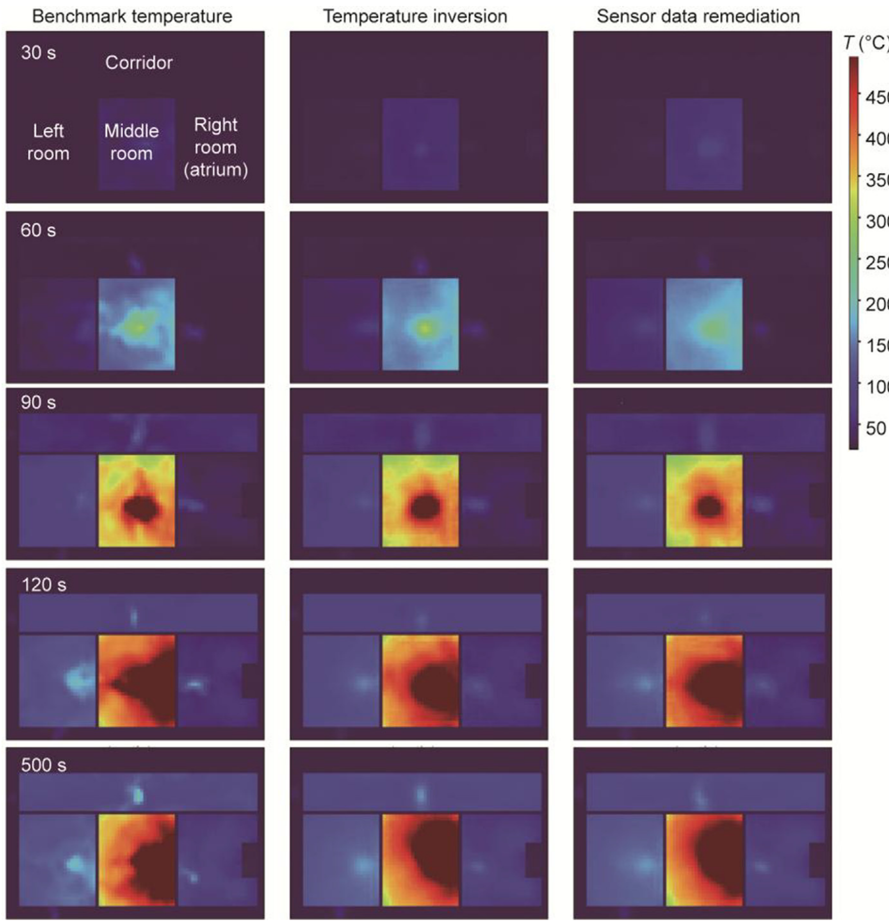

Fig. 10. Comparison of temperature fields between CFD and the ConvLSTM-Fire model. (a) Left column: benchmark data from CFD; (b) middle column: AI-driven prediction based on sensors; (c) right column: AI-driven prediction assuming failure of the sensor (middle room). Please see Video S1 for more details.

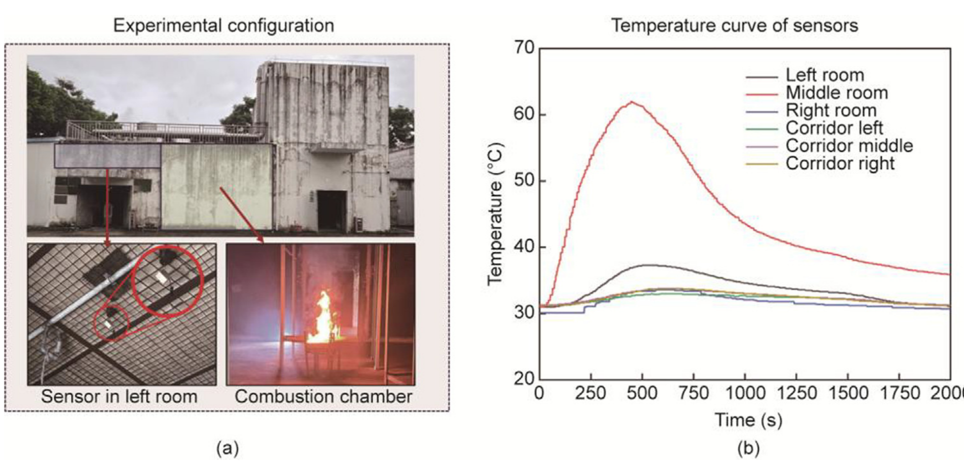

Fig. 11. Full-scale building fire (a) experiment and (b) historical data.

following the peak temperature. The consistency with experimental trends demonstrates the resiliency of ConvLSTM-Fire in fire information inference. Fig. 12(b) presents the 2D temperature field at a representative time (t = 450 s), highlighting potential thermal hazards within the building. As the fire data generated by the model has been validated for reliability, this temperature field can serve as scene input for the LLM to generate fire-aware and context-specific guidance.

4.2. Quantitative evaluation of LLM-driven output

To address the limitation that generic knowledge alone cannot provide LLMs with scene-specific understanding, the suitability of AI-driven fire data as domain knowledge for decision support is evaluated under experimental conditions.

Based on the representative large language model ChatGPT-4o, three decision modes are developed: "LLM-Generic," which

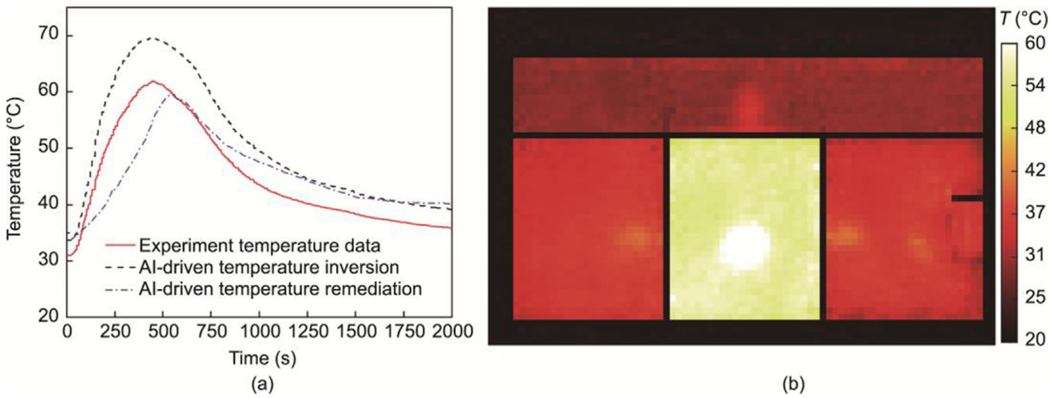

Fig. 12. Fire situational awareness for the experimental scenario driven by the ConvLSTM-Fire model.

responds to user queries without incorporating fire scene data; "LLM+6-Sensors," which augments queries with 6 sensor measurements from the IoT system as domain knowledge; and "LLM+2D-Temp", which uses the 2D temperature field generated by the ConvLSTM-Fire model as contextual input. To enable quantitative analysis and reduce output variability, a fixed system prompt is applied to standardize the response format. For each room, the output is constrained to the structured format {"name", "T_max_C", "T_fre_C", "risk", "ff_entry", "actions"}. Under this configuration, responses are generated for all three decision modes.

It is necessary to establish benchmark emergency response measures for evaluating LLM outputs. As no published references are available for this specific scenario, a room-level action plan is manually constructed based on an analysis of the temperature distribution. According to the *Code of Practice for Fire Safety in Buildings* [61] in Hong Kong, a temperature threshold of 60 ℃ is applied to classify risk and determine firefighter entry policies. The resulting action plan is then formatted by ChatGPT as the reference for metric calculation. The responses generated under the three different modes are presented in Table 3.

To quantitatively evaluate the quality of the three different decision modes, the relevant metrics are summarized in Table 4 and compared in Fig. 13. Bilingual evaluation understudy (BLEU) is adopted to measure the textual accuracy of LLM-generated responses, defined as follows:

$$P_n = \frac{m_n}{g_n} \tag{6-1}$$

$$\mathrm{BLEU} = \mathrm{BP} \times \exp\left(\sum_{n=1}^{N} W_n \log(P_n)\right) \tag{6-2}$$

where P_n denotes the precision at the n-gram level between the generated text and the reference texts; m_n denotes the number of matched n-grams; g_n denotes the total number of n-grams in the generated text. *BP* represents the brevity penalty, and $W_n = 1/N$. The upper bound of N is set to 4, meaning that up to 4-gram accuracy is considered. In addition, Recall-oriented Understudy for Gisting Evaluation (ROUGE) is employed to incorporate recall-based information, reflecting the coverage of the generated responses:

$$R_n = \frac{m_n}{r_n} \tag{7-1}$$

$$\mathrm{ROUGE_{F1}} = \frac{2 \times P_n \times R_n}{P_n + R_n} \tag{7-2}$$

where R_n denotes the recall rate at the n-gram level between the generated text and the reference text; r_n denotes the total number of n-grams in the reference text.

In addition, semantic similarity can be quantified by mapping text into vector representations. Specifically, texts are vectorized

Table 3
Comparison between the reference and three different decision modes.

Type	Content
Reference	Left Room: peak 40.8 °C, reference 38.4 °C, estimated incapacitation time ≈ 782 min; risk: low; entry: entry_allowed_with_caution; actions: routine_search, continuous_monitoring. Middle room: peak 72.0 °C, reference 60.1 °C, estimated incapacitation time ≈ 80.9 min; risk: moderate; entry: limited_entry_with_cooling; actions: doorway_cooling, short_entry_with_monitoring, ventilation_control. Right room: peak 41.3 °C, reference 38.3 °C, estimated incapacitation time ≈ 601.4 min; risk: low; entry: entry_allowed_with_caution; actions: routine_search, continuous_monitoring. Corridor: peak 35.9 °C, reference 32.3°C, estimated incapacitation time ≈ 997.5 min; risk: low; entry: entry_allowed_with_caution; actions: routine_search, continuous_monitoring. **Overall priority:** search rooms where entry is allowed with caution; cool down doorway when entry middle room and limit entry time; maintain continuous monitoring and ventilation control; re-evaluate if temperatures rise
LLM-Generic	Information is insufficient for room-level decisions. Adhere to fire safety SOPs: control ventilation, ensure occupant evacuation, and stage resources for cooling from safe positions. Only after conditions are verified as tenable should teams consider brief interior reconnaissance
LLM+6-Sensors	Left room: 36.9 °C; entry allowed with caution and continuous monitoring. Middle room: 62.0 °C; limited entry with cooling and ventilation control. Right room: 33.1 °C; entry allowed with caution and continuous monitoring. Corridor: 33.3 °C; entry allowed with caution and continuous monitoring. **Overall priority:** cool the hottest space first (if ≥ 60 °C), otherwise proceed with cautious search and monitoring
LLM+2D-Temp	Left room: peak 41 °C, reference 39 °C; conditions are generally allowed entry with caution for routine search and continuous monitoring. Middle room: temperatures more than 60 °C; employ doorway cooling and limit entry time with monitoring. Right room: peak 41.3 °C, reference 39 °C; conditions are generally allowed entry with caution for routine search and continuous monitoring. Corridor: peak 37 °C, reference 34 °C; conditions are generally allowed entry with caution for routine search and continuous monitoring. **Overall priority:** follow search path and search rooms with entry allowed while maintaining ventilation control; reassess if temperatures rise

Table 4
Quantitative evaluations of the responses generated by three different decision modes.

Test metrics	Item	LLM-Generic	LLM+6-Sensors	LLM+2D-Temp
BLEU SCORE (Uni-gram)	–	0.0091	0.2023	0.3147
BLEU SCORE (Bi-gram)	–	0.0000	0.1544	0.2431
Rouge1	Precision	0.2667	0.6484	0.5820
	Recall	0.0609	0.2995	0.3604
	F1	0.0992	0.4097	0.4451
Rouge2	Precision	0.0000	0.3778	0.3471
	Recall	0.0000	0.1735	0.2143
	F1	0.0000	0.2378	0.2650
RougeL	Precision	0.1556	0.4835	0.5082
	Recall	0.0355	0.2234	0.3147
	F1	0.0579	0.3056	0.3887
Cosine similarity	–	0.0444	0.2896	0.3226
Euclidean distance	–	1.3825	1.1920	1.1640

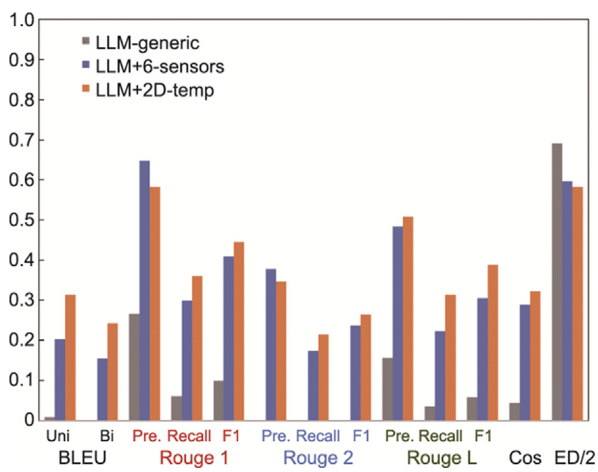

Fig. 13. Evaluation result histogram for three different decision modes (Uni means unigram, 1-gram; Bi means bigram, 2-gram).

using Bidirectional Encoder Representation from Transformers (BERT) embeddings, and the cosine similarity between the reference texts ($\boldsymbol{A}$) and the generated response ($\boldsymbol{B}$) is computed as:

$$\text{Similarity}(A,B) = \frac{\boldsymbol{A}\cdot\boldsymbol{B}}{\|\boldsymbol{A}\|\times\|\boldsymbol{B}\|} = \frac{\sum_{i=1}^{z}(\boldsymbol{A}_i\times\boldsymbol{B}_i)}{\sqrt{\sum_{i=1}^{z}\boldsymbol{A}_i^2}\times\sqrt{\sum_{i=1}^{z}\boldsymbol{B}_i^2}} \quad (8)$$

Furthermore, the Euclidean distance between the two vectors $\boldsymbol{A} = (x_1, x_2, \ldots, x_z)$ and $\boldsymbol{B} = (y_1, y_2, \ldots, y_z)$ can be employed to quantify semantic dissimilarity [62,63] defined as follows:

$$\text{Distance}(\boldsymbol{A},\boldsymbol{B}) = \|\boldsymbol{A}-\boldsymbol{B}\| = \sqrt{\sum_{i=1}^{z}(x_i-y_i)^2} \quad (9)$$

The LLM-Generic mode exhibits minimal lexical overlap and weak semantic similarity across all evaluation metrics. Its poor performance in BLEU_Bi and Rouge2 indicates a lack of effective scenario-specific action guidance, such as "doorway_cooling." For LLM+6-Sensors, the inclusion of IoT sensor data significantly improves inference performance at the room level by enabling customizing decisions, for example, "middle room: limited entry with cooling." However, the further improvement in BLEU and ROUGE scores for LLM+2D-Temp demonstrates higher textual accuracy and broader content coverage compared with LLM+6-Sensors. In addition, LLM+2D-Temp achieves the strongest semantic similarity, with the highest cosine similarity (Cos) and the smallest Euclidean distance (ED). Overall, the LLM+2D-Temp model, which incorporates a full 2D temperature map, achieves a better balance between precision (Pre.) and comprehensiveness for the specific fire scenario. As more scene-specific information is provided, the actionability of LLM responses improves substantially. When equipped with ConvLSTM-Fire, LLM+2D-Temp enables operationally meaningful guidance for emergency response in building fire accidents.

4.3. Fire-emergency response and decision support

A hybrid self-driven fire-emergency agent can address a wide range of emergency response needs, including basic user interactions as well as the invocation of well-trained deep learning models for functional redevelopment and emergency decision-making.

To improve human–computer interaction, the GUI for the proposed agent is developed, as shown in Fig. 14. This interface demonstrates general interaction based solely on the pre-embedded model knowledge. An initial understanding of the agent can be obtained through a query such as: "Hi, please describe the difference between you and the generic large language model? And what are your unique applications in smart firefighting?" Because information related to the deep learning models is pre-initialized, no additional data sources are required to generate a response. In replying to this query, the agent highlights key features, including domain expertise, practical implementation, and safety awareness. In addition, the agent demonstrates a comprehensive understanding of the functionality of the integrated pre-trained deep learning models. It summarizes its unique smart firefighting application when combined with the ConvLSTM-Fire model, such as temperature monitoring, heat mapping, predictive analysis, and safety alerts (Video S2 in Appendix A for more details). These capabilities enable users to more effectively utilize the model and support further functional development based on its strengths.

Fig. 15 provides an end-to-end demonstration of deep learning model invocation and subsequent processing by the self-driven agent, exemplified through three representative use cases. Fig. 15(a) shows how users can directly invoke the prediction function of the ConvLSTM-Fire model using natural language commands. The query is "If the data from 6 temperature sensors at this point is (80, 100, 150, 60, 78, 130), give the temperature plan of the building." The LLM interprets the request and generates a five-step action sequence: ① normalize input data, ② load the model, ③ use the model to prediction, ④ de-normalize the output, and ⑤ visualize the results and generate a runnable program script. The script is then automatically executed within the self-driven framework to produce the building-wide temperature distribution.

Fig. 15(b) illustrates a functional redevelopment scenario in which the user requests both the mean and maximum temperature values to quantify overall building fire conditions. Using the generated 2D temperature array, the agent computes an average temperature of 70.15 °C and a maximum temperature of 186.14 °C,

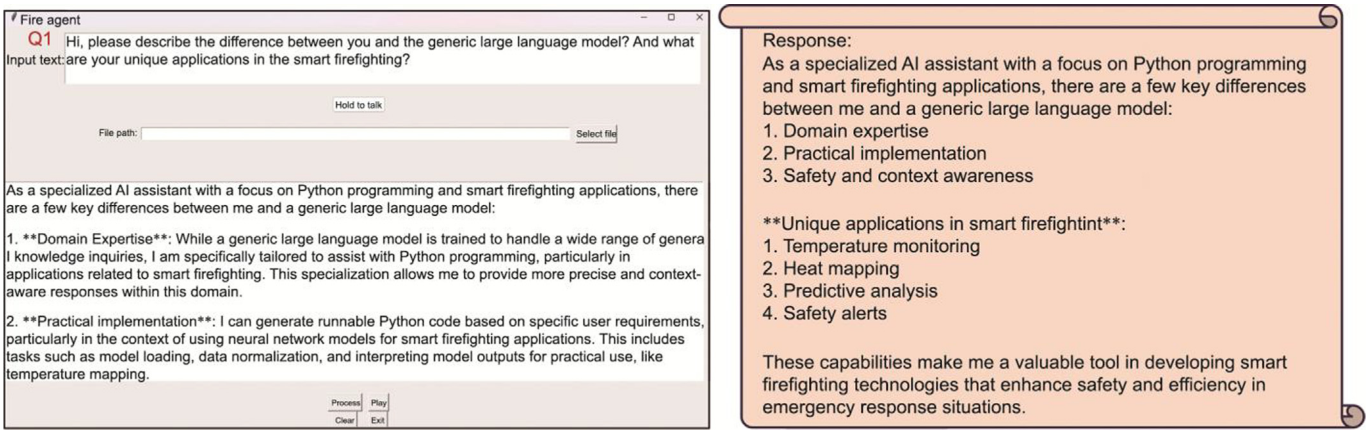

Fig. 14. GUI-based generic interactions (Video S2 for more details).

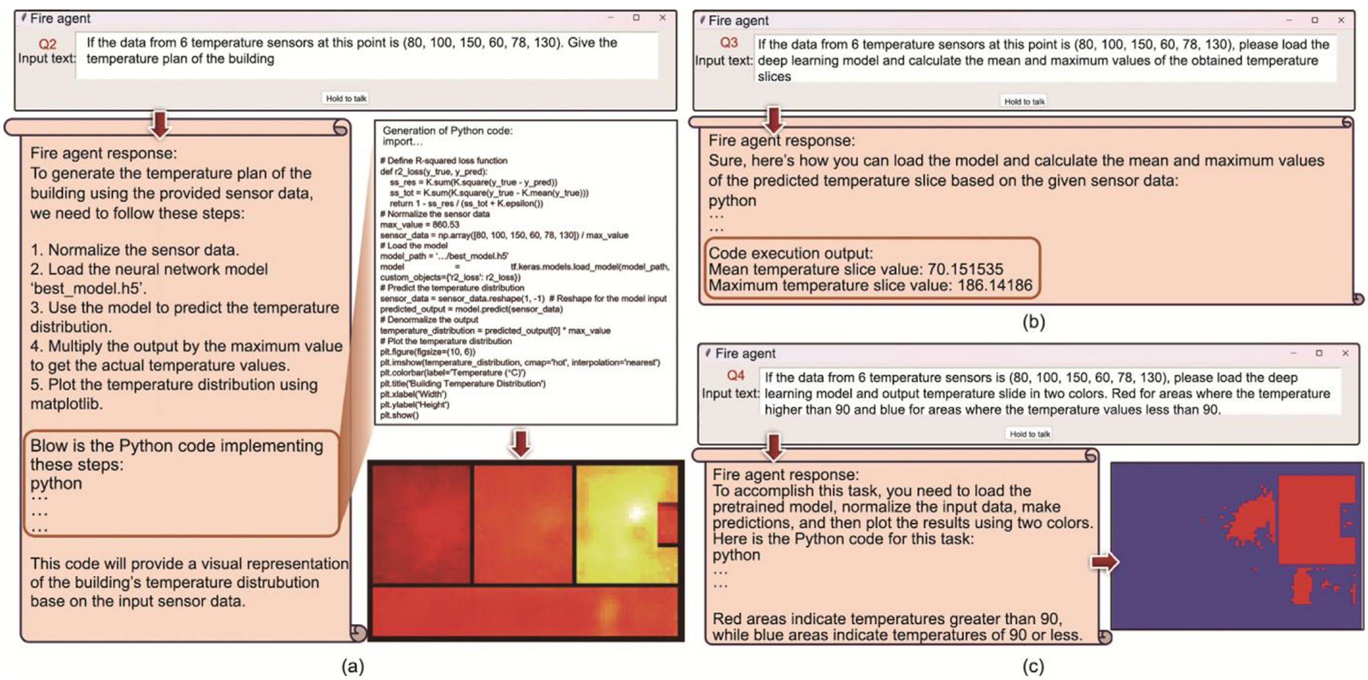

Fig. 15. Agent-driven ConvLSTM-Fire model invocation and functional re-development (Video S3 for details).

with the result returned directly in the chat interface. Fig. 15(c) demonstrates another redevelopment case focused on hazardous area identification. By designing a temperature threshold of 90 °C, the agent automatically classifies the temperature field into hazardous and non-hazardous zones and presents the results in a colorized visualization. This approach simplifies complex spatial distributions into actionable visual information, thereby supporting decision-making and enabling users to focus on evacuation strategy development based on their proximity to hazardous zones (Video S3 in Appendix A for details).

Fig. 16 illustrates emergency decisions supported by fire situational awareness and audio-based interaction. In this configuration, visual outputs generated by the ConvLSTM-Fire model are stored locally and retrieved through a prescribed path. The visual information is encoded and parsed for communication with the LLM, enabling the agent to analyze and interpret AI-generated prediction results. As shown in the response to Q5, the agent demonstrates advanced image comprehension capabilities, as it not only recognizes the color bar and associated text but also correctly interprets color variations as representations of the building temperature distribution. In addition, the agent exhibits spatial awareness by identifying darker regions corresponding to walls, thereby segmenting the building layout into four distinct rooms. This capacity to interpret visual cues confers a comprehensive understanding of the environment, supporting more nuanced reasoning and informed emergency decision-making.

In response to a follow-up inquiry regarding room entry feasibility, the smart fire-emergency agent provides accurate guidance based on the uploaded temperature distribution and predefined temperature damage guidelines. For the top–right room, which exhibits hazardous conditions, the agent clearly advises against entry. For the other two rooms with lower temperatures, the agent supports entry while emphasizing the need for continuous situational awareness. Rather than relying solely on precise numerical outputs from the ConvLSTM-Fire model, the agent derives a preliminary risk hierarchy from the visual temperature map and user instructions. This demonstrates its ability to integrate information from multiple data sources and generate actionable insights in real

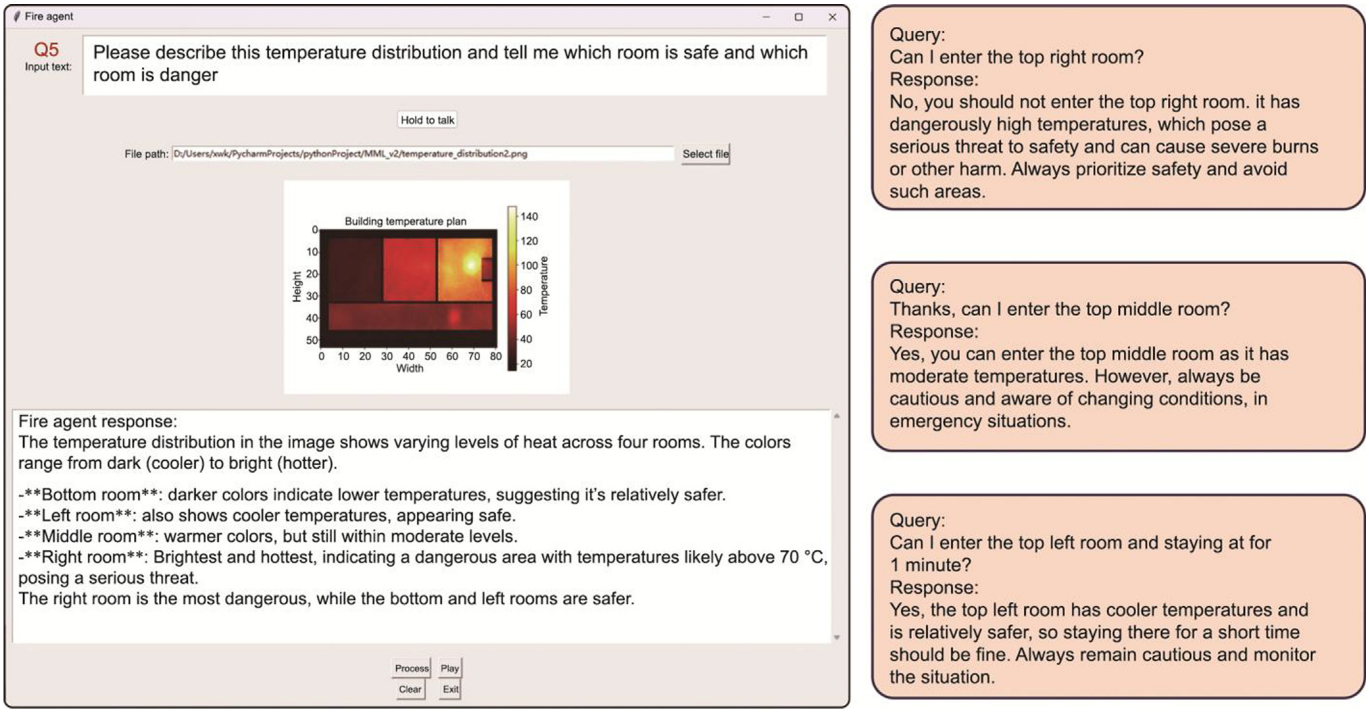

Fig. 16. Emergency decision supported by fire situational awareness and audio interaction (Video S4 for more details).

time, which is particularly valuable in emergency scenarios requiring rapid decision-making. In addition, the use of voice input and text-to-speech output reduces the cognitive and attentional demands on operators. As a result, the agent avoids imposing excessive decision-making burden and shows promise as an effective emergency decision-support aid (Video S4 in Appendix A for details).

Coupled with automated interaction mechanisms involving deep learning models and IoT devices, the proposed agent significantly expands its problem-solving capabilities, particularly in tracking evolving fire patterns. Such situational dynamics-based information support has the potential to help humans overcome cognitive limitations, thereby providing more scientifically grounded guidance for emergency response. Overall, the hybrid self-driven fire-emergency agent effectively addresses emergency response demands during building fires, making the practical implementation of smart firefighting technology technically feasible.

5. Conclusions

To overcome the practicality challenges of smart firefighting systems, this study proposed a self-driven smart agent framework that enables dynamic, situational-awareness emergency response for diverse firefighting requirements in building fires. As the foundation for real-time situational awareness, the deep learning model ConvLSTM-Fire was developed to perform spatiotemporal inference of temperature fields, even under extreme conditions where critical sensor data may be compromised. Based on discrete sensor arrays, ConvLSTM-Fire reconstructs 2D building temperature slices with over 97% accuracy by intelligently compensating for sensor data loss through historical data analysis.

Building on the dynamic characteristics of the fire scenes, a domain-specific LLM was developed to overcome the limitations of generic knowledge optimization in guiding scenario-specific fire response. Through interaction between the LLM and specialized deep learning models, such as ConvLSTM-Fire, the proposed agent enables customized decision-making strategies based on building fire information. Using natural language queries, the agent automatically orchestrates temperature prediction and decision-making processes, thereby delivering actionable guidance tailored to the unique requirements of different stakeholders.

Moreover, this work equips the agent with a novel self-driven framework for functional redevelopment. By establishing operational mechanisms that execute LLM-generated information within the local environment, the agent can autonomously reason and execute actions in emergency scenarios, including model invocation and data processing. This framework provides a generic implementation foundation for smart firefighting systems, enabling agents to respond to diverse user requirements in real time. Supported by the GUI with audio interaction capabilities, the proposed system reduces constraints related to time, expertise, and platform dependence, fostering end-to-end interaction between users and smart firefighting systems.

Overall, the proposed hybrid self-driven fire-emergency agent substantially advances the practical deployment of smart firefighting technologies. By combining robust data analytics with intuitive human–computer interactions, the agent establishes a foundation for more adaptive fire management systems that align with the evolving demands of modern smart city environments.

To address the limitations of this study, future work will further investigate the ConvLSTM-Fire model through architectural optimization and robustness testing under multi-sensor failure conditions. The enhanced model is expected to provide higher-fidelity environmental awareness information for downstream large language models. In addition, survival thresholds for different sensor types under building fire conditions can be identified to inform the initiation of data remediation mechanisms. Furthermore, improving the real-time response speed of the model remains a priority to enhance overall system resilience. Through optimized deployment strategies and query handling, the negative effects of network latency can be further mitigated.

CRediT authorship contribution statement

Weikang Xie: Writing – original draft. **Yuxin Zhang:** Writing – review & editing. **Tong Lu:** Formal analysis. **Xianjia Huang:** Resources. **Jihao Shi:** Formal analysis. **Xinyan Huang:** Writing – review & editing, Methodology, Conceptualization. **Fu Xiao:** Writing – review & editing. **Asif Usmani:** Funding acquisition.

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.

Acknowledgment

This research is funded by the Hong Kong Research Grants Council under the Theme-based Research Scheme (T22-505/19-N), the National Natural Science Foundation of China (52108480), the MTR Research Fund (PTU-23005), and the Hong Kong Polytechnic University (PolyU; P0045772).

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.eng.2026.02.023.

References

[1] Shi J, Pan Z, Jiang L, Zhai X. An ontology-based methodology to establish city information model of digital twin city by merging BIM, GIS and IoT. Adv Eng Inform 2023;57:102114.
[2] Alshamaila Y, Papagiannidis S, Alsawalqah H, Aljarah I. Effective use of smart cities in crisis cases: a systematic review of the literature. Int J Disaster Risk Reduct 2023;85:103521.
[3] Han L, Potter S, Beckett G, Pringle G, Welch S, Koo SH, et al. FireGrid: an e-infrastructure for next-generation emergency response support. J Parallel Distrib Comput 2010;70(11):1128–41.
[4] McNamee M, Meacham B, van Hees P, Bisby L, Chow WK, Coppalle A, et al. IAFSS agenda 2030 for a fire safe world. Fire Saf J 2019;110:102889.
[5] Rezaeifam S, Ergen E, Günaydın HM. Fire emergency response systems information requirements' data model for situational awareness of responders: a goal-directed task analysis. J Build Eng 2023;63:105531.
[6] Leung H. Deaths of 2 Hong Kong firefighters in 2016 storage unit blaze ruled as a result of misadventure [Internet]. Hong Kong: Hong Kong Free Press; 2022 May 11 [cited 2026 Feb 11]. Available from: https://hongkongfp.com/2022/05/11/deaths-of-2-hong-kong-firefighters-in-2016-storage-unit-blaze-ruled-as-a-result-of-misadventure/.
[7] Zeng Y, Zheng Z, Zhang T, Huang X, Lu X. AI-powered fire engineering design and smoke flow analysis for complex-shaped buildings. J Comput Des Eng 2024;24(11):359–73.
[8] Li N, Yang Z, Ghahramani A, Becerik-Gerber B, Soibelman L. Situational awareness for supporting building fire emergency response: information needs, information sources, and implementation requirements. Fire Saf J 2014;63:17–28.
[9] Zhang Y, Zhang X, Huang X. Design a safe firefighting time (SFT) for major fire disaster emergency response. Int J Disaster Risk Reduct 2023;88:103606.
[10] Lin SS, Shen SL, Zhou A, Chen XS. Smart techniques promoting sustainability in construction engineering and management. Engineering 2025;45:262–82.
[11] Zheng J, Meister M. The unbearable slowness of being: why do we live at 10 bits/s? Neuron 2025;113(2):192–204.
[12] Brown T, Mann B, Ryder N, Subbiah M, Kaplan JD, Dhariwal P, et al. Language models are few-shot learners. In: Larochelle H, Ranzato M, Hadsell R, Balcan MF, Lin H, editors. Advances in neural information processing systems. Red Hook: Curran Associates, Inc.; 2020. p. 1877–901.
[13] Naveed H, Khan AU, Qiu S, Saqib M, Anwar S, Usman M, et al. A comprehensive overview of large language models. ACM Trans Intell Syst Technol 2023;16(5):106.
[14] Oktari RS, Munadi K, Idroes R, Sofyan H. Knowledge management practices in disaster management: systematic review. Int J Disaster Risk Reduct 2020;51:101881.
[15] Leitão P, Karnouskos S, Ribeiro L, Lee J, Strasser T, Colombo AW. Smart agents in industrial cyber–physical systems. Proc IEEE 2016;104(5):1086–101.
[16] O'Brien PD, Wiegand ME. Agent based process management: applying intelligent agents to workflow. Knowl Eng Rev 1998;13(2):161–74.
[17] Zhang T, Wang Z, Zeng Y, Wu X, Huang X, Xiao F. Building artificial-intelligence digital fire (AID-Fire) system: a real-scale demonstration. J Build Eng 2022;62:105363.
[18] Xie W, Zeng Y, Zhang X, Wong HY, Zhang T, Wang Z, et al. AIoT-powered building digital twin for smart firefighting and super real-time fire forecast. Adv Eng Inform 2025;65:103117.
[19] De Iuliis M, Battegazzorre E, Domaneschi M, Cimellaro GP, Bottino AG. Large scale simulation of pedestrian seismic evacuation including panic behavior. Sustain Cities Soc 2023;94:104527.
[20] Jiang Y. The applications of large language models in emergency management. In: Proceedings of the 2024 IEEE 6th advanced information management, communicates, electronic and automation control conference (IMCEC), 2024 May 24–26, Chongqing, China. New York City: IEEE; 2024. p. 199–202.
[21] Chen M, Tao Z, Tang W, Qin T, Yang R, Zhu C. Enhancing emergency decision-making with knowledge graphs and large language models. Int J Disaster Risk Reduct 2024;113:104804.
[22] Lorusso P, De Iuliis M, Marasco S, Domaneschi M, Cimellaro GP, Villa V. Fire emergency evacuation from a school building using an evolutionary virtual reality platform. Buildings 2022;12(2):223.
[23] Xie W, Zhang X, Shi J, Huang X, Chang Y, Usmani AS, et al. Real-time spatiotemporal forecast of natural gas jet fire from offshore platform by using deep probability learning. Ocean Eng 2024;294:116658.
[24] Zhang Y, Yan Z, Zhu H, Tang P. Physics-based model and data dual-driven approaches for predictive evacuation. Dev Built Environ 2023;16:100269.
[25] Zhang X, Zhang T, Ding Y, Huang X. Internet of things and digital twin in fire safety management. In: Intelligent building fire safety and smart firefighting. Berlin: Springer; 2024. p. 335–61.
[26] Zhang Y, Geng P, Sivaparthipan CB, Muthu BA. Big data and artificial intelligence based early risk warning system of fire hazard for smart cities. Sustain Energy Technol Assess 2021;45:100986.
[27] Ding Y, Chen X, Zhang Y, Huang X. Smart building evacuation by tracking multi-camera network and explainable re-identification model. Eng Appl Artif Intell 2025;148:110394.
[28] Cimellaro GP, Mahin S, Domaneschi M. Integrating a human behavior model within an agent-based approach for blasting evacuation. Comput Aided Civ Infrastruct Eng 2019;34(1):3–20.
[29] Rivkin D, Hogan F, Feriani A, Konar A, Sigal A, Liu X, et al. AIoT smart home via autonomous LLM agents. IEEE Internet Things J 2025;12(3):2458–72.
[30] Barman KG, Wood N, Pawlowski P. Beyond transparency and explainability: on the need for adequate and contextualized user guidelines for LLM use. Ethics Inf Technol 2024;26(3):47.
[31] Bedi S, Liu Y, Orr-Ewing L, Dash D, Koyejo S, Callahan A, et al. A systematic review of testing and evaluation of healthcare applications of large language models (LLMs). MedRxiv:2024.04.15.24305869; 2024.
[32] Wu F, Shen T, Bäck T, Chen J, Huang G, Jin Y, et al. Knowledge-empowered, collaborative, and co-evolving AI models: the post-LLM roadmap. Engineering 2025;44:87–100.
[33] Kim S, Park S, Shin J, Choi IR, Choi S. Deep-learning-based data loss reconstruction for spatiotemporal temperature in piloti structures: enhancing applicability with limited datasets. Fire Saf J 2023;140:103887.
[34] Sturm P, Fößleitner P, Fruhwirt D, Heindl SF, Heger O, Galler R, et al. Dataset of fire tests with lithium-ion battery electric vehicles in road tunnels. Data Brief 2022;46:108839.
[35] Rubin DB. Inference and missing data. Biometrika 1976;63(3):581–92.
[36] Cao X, Wu K, Geng X, Guan Q. Field detection of indoor fire threat situation based on LSTM-Kriging network. J Build Eng 2024;84:108686.
[37] Hochreiter S, Schmidhuber J. Long short-term memory. Neural Comput 1997;9(8):1735–80.
[38] Ma J, Cheng JCP, Jiang F, Chen W, Wang M, Zhai C. A bi-directional missing data imputation scheme based on LSTM and transfer learning for building energy data. Energy Build 2020;216:109941.
[39] Zeng Y, Li Y, Du P, Huang X. Smart fire detection analysis in complex building floorplans powered by GAN. J Build Eng 2023;79:107858.
[40] Adhikari D, Jiang W, Zhan J, He Z, Rawat DB, Aickelin U, et al. A comprehensive survey on imputation of missing data in Internet of Things. ACM Comput Surv 2022;55(7):133.
[41] Li Z, Liu F, Yang W, Peng S, Zhou J. A survey of convolutional neural networks: analysis, applications, and prospects. IEEE Trans Neural Netw Learn Syst 2022;33(12):6999–7019.
[42] Ren L, Wang H, Wang Y, Huang K, Wang L, Li B. Foundation models for the process industry: challenges and opportunities. Engineering 2025;52:53–9.
[43] Ge Y, Hua W, Mei K, Ji JC, Tan J, Xu S, et al. OpenAGI: when LLM meets domain experts. In: Oh A, Naumann T, Globerson A, Saenko K, Hardt M, Levine S, editors. Advances in neural information processing systems. Red Hook: Curran Associates, Inc.; 2023. p. 5539–68.
[44] Luo G, Weng L, Li Y, Sun Y, Hong Y, Wu Y, et al. FireExpert: fire event identification and assessment leveraging cross-domain knowledge and large language model. IEEE Trans Mobile Comput 2025;24(6):1–18.
[45] Hostetter H, Naser MZ, Huang X, Gales J. The role of large language models (AI chatbots) in fire engineering: an examination of technical questions against domain knowledge. Nat Hazards Res 2024;4(4):669–88.
[46] Liu J, Xia CS, Wang Y, Zhang L. Is your code generated by ChatGPT really correct? Rigorous evaluation of large language models for code generation. In: Oh A, Naumann T, Globerson A, Saenko K, Hardt M, Levine S, editors. Advances in neural information processing systems. Red Hook: Curran Associates, Inc.; 2023. p. 21558–72.

[47] Rozière B, Gehring J, Gloeckle F, Sootla S, Gat I, Tan E, et al. Code Llama: open foundation models for code. arXiv:2308.12950; 2024.
[48] DeepSeek-AI, Zhu Q, Guo D, Shao Z, Yang D, Wang P, et al. DeepSeek-Coder-V2: breaking the barrier of closed-source models in code intelligence. arXiv:2406.11931; 2024.
[49] Allamanis M, Barr ET, Devanbu P, Sutton C. A survey of machine learning for big code and naturalness. ACM Comput Surv 2018;51(4):81.
[50] Radford A. Improving language understanding by generative pre-training. Homology Homotopy Appl 2018;9:399–438.
[51] Chen M, Tworek J, Jun H, Yuan Q, Ponde H, Pinto DO, et al. Evaluating large language models trained on code. arXiv:2107.03374; 2021.
[52] Yu X, Liu L, Hu X, Keung JW, Liu J, Xia X. Fight fire with fire: how much can we trust ChatGPT on source code-related tasks? IEEE Trans Softw Eng 2024;50(12):3435–53.
[53] Talebirad Y, Nadiri A. Multi-agent collaboration: harnessing the power of intelligent LLM agents. arXiv:2306.03314; 2023.
[54] Otal HT, Stern E, Canbaz MA. LLM-assisted crisis management: building advanced LLM platforms for effective emergency response and public collaboration. In: Proceedings of the 2024 IEEE conference on artificial intelligence (CAI), 2024 Jun 25–27, Singapore City, Singapore. New York City: IEEE; 2024. p. 851–9.
[55] Lee J, Kim B, Lee S, Shin WG. Validation of the fire dynamics simulator (FDS) model for fire scenarios with two liquid pool fires in multiple compartments. Fire Saf J 2023;141:103892.
[56] McGrattan K, Hostikka S, McDermott R, Floyd J, Weinschenk C, Overholt K. Fire dynamics simulator user's guide. 6th ed. Gaithersburg: National Institute of Standards and Technology (NIST) Special Publication; 2015.
[57] Wahlqvist J, van Hees P. Validation of FDS for large-scale well-confined mechanically ventilated fire scenarios with emphasis on predicting ventilation system behavior. Fire Saf J 2013;62:102–14.
[58] Audouin L, Chandra L, Consalvi JL, Gay L, Gorza E, Hohm V, et al. Quantifying differences between computational results and measurements in the case of a large-scale well-confined fire scenario. Nucl Eng Des 2011;241(1):18–31.
[59] Wang H, Li J, Wu H, Hovy E, Sun Y. Pre-trained language models and their applications. Engineering 2023;25:51–65.
[60] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, et al. Attention is all you need. In: Guyon I, Von LU, Bengio S, Wallach H, Fergus R, Vishwanathan S, editors. Advances in neural information processing systems. Red Hook: Curran Associates, Inc.; 2017. p. 6000–10.
[61] Buildings Department. Code of practice for fire safety in buildings 2011. 2024 ed. Chinese standard. Hong Kong: Buildings Department; 2024.
[62] Hu T, Zhou XH. Unveiling LLM evaluation focused on metrics: challenges and solutions. ArXiv:2404.09135; 2024.
[63] Lee J, Ahn S, Kim D, Kim D. Performance comparison of retrieval-augmented generation and fine-tuned large language models for construction safety management knowledge retrieval. Autom Construct 2024;168:105846.