IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 19, 2021, accepted February 16, 2021, date of publication February 18, 2021, date of current version February 26, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3060338

Advances Toward the Next Generation Fire Detection: Deep LSTM Variational Autoencoder for Improved Sensitivity and Reliability

ZHAOYI XU, YANJIE GUO, AND JOSEPH HOMER SALEH, (Senior Member, IEEE)
School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA

Corresponding author: Zhaoyi Xu (zxu328@gatech.edu)

This work was supported in part by the Space Technology Research Institute grant from NASA's Space Technology Research Grant Program.

ABSTRACT Fire detection is a critical component of a building safety monitoring system and remains an important research area with weighty practical relevance. Significant advances have occurred in recent years in building automation, and the operation of buildings has become more complex and requires ever more effective monitoring systems. In this work, we develop a novel fire detection method using deep Long-Short Term Memory (LSTM) neural networks and variational autoencoder (VAE) to meet these increasingly stringent requirements and outperform existing fire detection methods. To evaluate the effectiveness of our method, we develop high-fidelity simulations, and we use datasets from real-world fire and non-fire experiments provided by NIST. We compare and discuss the performance of our proposed fire detection with alternative methods, including the standard LSTM, cumulative sum control chart (CUSUM), exponentially weighted moving average (EWMA), and two currently used fixed-temperature heat detectors. The results using the simulation-based and the real-world experiments are complementary, and they indicate that the LSTM-VAE robustly outperforms the other detection methods with, for example, statistically significant shorter alarm time lags, no missed detection, and no false alarms. The results also identify shortcomings of other detection methods and indicate a clear ranking among them (LSTM-VAE$\succ$*EWMA* $\succ$ *LSTM*$\succ$*CUSUM*).

INDEX TERMS Fire detection, machine learning, anomaly detection, long short-term memory, variational autoencoder.

I. INTRODUCTION

We first provide the context within which our work on fire detection is situated. This includes some background on fire detection and a brief discussion of machine learning for anomaly detection. We then delineate our scope and objectives and provide a roadmap for the remainder of this work.

A. BACKGROUND ON FIRE DETECTION

Significant advances have occurred in recent years in building automation and information systems. The operation of buildings has become more complex, and several of its functions require increasingly more effective and reliable monitoring of the environment within buildings (their internal state). Fire detection, as a critical component of a building safety monitoring system, uses fire signatures such as smoke, heat, CO_2, or radiation, to identify early signs of fire and trigger alarms. There can be significant costs associated with missed fire detection in terms of loss of life and property damage when signs of fire are not detected early enough to neutralize or contain the unfolding accident at an early stage. There can also be significant costs associated with false alarms if, for example, sprinklers are triggered unnecessarily and water damage to the building occurs. Advanced fire detections use statistical models and optimization methods to improve the detection accuracy and enhance the understanding of fire event development [1]–[5].

Effective anomaly detection in general, and fire detection in particular remains a critical research area with substantial practical relevance [6]. Its focus is to improve, among other things, the sensitivity of the detection scheme and reduce its false alarm rate and missed detection. The sensitivity of fire

The associate editor coordinating the review of this manuscript and approving it for publication was Hiu Yung Wong.

detection stands for the ability to detect early signs of fire. The reduction of missed detection is related to sensitivity. For example, a highly sensitive detection system is capable of detecting early, small signatures of fire. Its missed detection will be low. However, its drawback is that these small detected signatures might be ambiguous and non-fire related, and as a result, the false alarm rate will be high. In short, a tradeoff is generally understood to mediate between these performance metrics of a fire detection system, its sensitivity on the one hand, and its false alarm rate (the complement of specificity) on the other hand.[1]

Fire detection methods can be classified into two broad categories based on the alarm triggering mechanism. The first category is memoryless threshold-based detection [7]–[11]. In this category, the fire alarm decision is made based on the comparison between the present sensor signal and a predefined threshold value above which the alarm is triggered. Only the present sensor output is accounted for in this decision. For example, fixed temperature heat detectors [7]–[9] belong to this category. They use materials with different melting points to set different temperature thresholds to achieve different sensitivity. This first category is subsumed under the broader heading of point anomaly detection [12]. The second category is history-based fire detection. In this category, the fire alarm is triggered based on past and present sensor output, the information contained in or extracted from the time-series data, not just the sensor's present output as in the previous, memoryless threshold category. One popular history-based method is the cumulative sum control chart (CUSUM) for fire detection [13], [14]. The CUSUM detector calculates a partial sum of the abnormal sensor signal and triggers the fire alarm when the sum exceeds a given threshold. In this case, there is memory of past sensor outputs in the alarm triggering decision. The challenge for this category of fire detection methods is to probe the dynamics of the sensor's output and extract meaningful features that are reliably predictive of fire occurrence. This second category is subsumed under the broader heading of contextual anomaly detection [12]. Conceptually, this second category of fire detection methods can be viewed as seeking to ''accumulate evidence'' over time before making a decision, whereas point anomaly detection methods operate with a single observation, e.g., ''exhibit A'', as the primary and only evidence in support of the decision to trigger the fire alarm. The present work belongs to the second category of methods of contextual anomaly detection: our objective is to leverage state-of-the-art machine learning tools to improve both the sensitivity and reliability of fire detection without compromising the false alarm rate. We expand on this shortly.

B. MACHINE LEARNING FOR ANOMALY DETECTION

There is a broader context within which our work on fire detection is situated. It is related to advances in machine learning (ML) in general, and unsupervised learning in particular for reliability and safety applications. We recently provided a review of this broad analytical landscape, and we include here a short excerpt for the convenience of the reader. More details can be found in [16]. Recent applications of Machine learning models in reliability engineering include methodology development, system diagnostic, remaining useful life estimation and prognostic health management [17]–[21]. Unsupervised learning consists in examining datasets with only input variables or features, and no labels or response variable. Its general objective is to explore the feature space and find patterns in the dataset. Two major sub-categories or tasks of unsupervised learning are based on the nature of the patterns sought: clustering and anomaly detection. Clustering consists in dividing the observations into clusters that share some similarities in the feature space. Anomaly detection consists in identifying unexpected observations in a dataset. The term anomaly in this ML contest is used in a broader sense than how it is understood in reliability and safety contexts. Within this ML context, anomaly detection refers to ''the problem of finding patterns in data that do not conform to expected normal behavior'' [12]. Anomaly detection algorithms have found applications in many domains because they produce critical information that can be acted upon and prompt meaningful intervention. For example, anomaly detection is used in cyber-security and intrusion detection [22], in banking and insurance fraud detection, in a host of medical applications [23], and increasingly in reliability and safety applications, which is where our work on fire detection fits in. Anomaly detection is particularly well-suited for and used in early fault detection of equipment and structures. It is related to sensor data, and for industrial machinery and equipment, the data typically comes in streaming fashion. Early detection of anomalies is essential in some contexts to prevent further damage and preempt catastrophic failures. The literature includes applications of anomaly detection in support of prognostic and health management (PHM) for different systems, for example, aircraft flight data recorders [24], industrial gas turbines [25], spacecraft operation and health monitoring [26], [27], and induction motors with a focus on ball-bearing faults [28]. Applications of anomaly detection algorithms for structural damage detection also abound, a discussion of which can be found in [12].

C. SCOPE AND OBJECTIVE

In this work, we consider fire as an anomaly in the monitored environment, and we leverage advanced ML anomaly detection algorithms, Long Short-Term Memory (LSTM), and variational autoencoder (VAE), to improve the sensitivity and reliability of fire detection. We develop a novel architecture and the supporting analytics for the next generation fire detection system. To evaluate the effectiveness of our method, we develop computational experiments with high-fidelity large eddy simulation (LES) data, and we also use real-world fire and non-fire datasets [29]. We compare our proposed

[1]There are several other detection performance metrics, generally included in the ''confusion matrix''. Details can be found in [15]; they are not relevant for our introductory purpose.

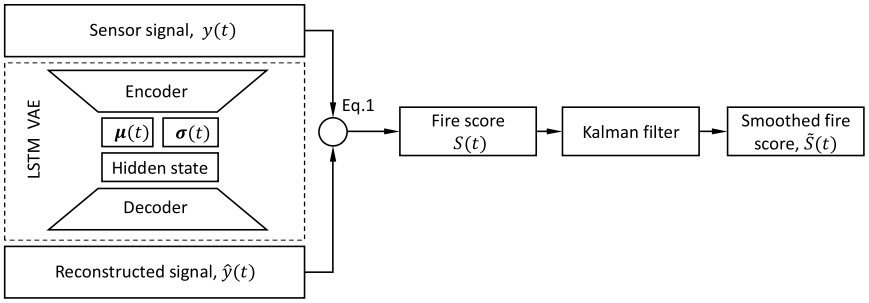

FIGURE 1. Architecture of the proposed LSTM-VAE fire detection.

fire detection with alternative methods, including standard LSTM detection [27], CUSUM fire detection [13], [14], exponentially weighted moving average (EWMA) anomaly detection [30], and fixed-temperature heat detection [7]–[9]. Our pool of alternative methods includes both point anomaly and contextual anomaly detections for a fair comparison. We also develop a two-sample t-test to statistically compare the performance of our method against the alternative fire detections in the simulation-based computational experiments. We restrict the present work to heat detectors in order to focus and expand on the method proposed and demonstrate its effectiveness. In a follow-up work, we extend this method to a host of other sensors, including smoke detectors, CO2 and CO sensors, and we examine sensor data fusion to further improve the performance of this fire detection system.

The main contribution of this work is the development and validation of a novel LSTM with VAE fire detection method, which provides robust and significant advantages over other fire detection approaches in terms of improved sensitivity and reduced false alarm rate.

The remainder of the article is organized as follows. Our LSTM-VAE fire detection framework along with its technical details are provided in Section II. The simulation-based computational experiments and the real-world fire (and non-fire) datasets used to train and validate the LSTM-VAE fire detection are discussed in Section III. The results are provided and examined in Section IV. Section V concludes this work.

II. LSTM-VAE FIRE DETECTION METHOD

In this section, we first provide a high-level overview of the LSTM-VAE fire detection architecture. We then discuss its technical details, including the LSTM neural network, the variational autoencoder, and the Kalman filter.

A. OVERVIEW OF LSTM-VAE FIRE DETECTION METHOD

We begin with the LSTM-VAE fire detection architecture shown in Fig. 1.

The following discussion is a cursory overview of the fire anomaly detection process. The technical details are provided in the next subsections. The LSTM-VAE fire detection consists of (1) a variational autoencoder with deep LSTM networks for both the encoder and decoder, and (2) a denoising Kalman filter. We use the LSTM-VAE neural network to encode the input signal. The encoder learns the most salient features or properties of the input signal. Next, the decoder rebuilds the input signal based on the previously learned lower dimensional representation of the signal. It might appear otiose to copy an input signal to a neural network and reproduce it as its output. This is the essence of an autoencoder, except the significant benefit obtained is in the features learned, and in our architecture, in the difference between the input signal and the output. More specifically, we compare the rebuild signal, $\hat{y}(t)$, with the sensor signal, $y(t)$, and we calculate a corresponding fire score $S(t)$ based on their difference, which can indicate a possible anomaly. We then use the Kalman filter to denoise $S(t)$ and produce a smoothed fire score, $\tilde{S}(t)$. The fire detection decision is based on this smoothed fire score.

The encoder and decoder networks are trained using the normal, non-fire operating conditions (details in the next subsections). The objective of the training is to have the LSTM-VAE "learn" the dynamics of the sensed quantity in non-anomalous conditions, and subsequently to reconstruct the signal based on these learned non-fire dynamics.

Once trained, the LSTM-VAE is used to supervise the sensor signal.[2] We compare the reconstructed signal $\hat{y}(t)$, the output of the decoder, with the online sensor signal $y(t)$, the input to the encoder, and we derive a fire score $S(t)$ as expressed in Eq. 1:

$$S(t) = \frac{y(t) - \hat{y}(t)}{\sigma_{train}} \tag{1}$$

σ_{train} is the root mean square error (RMSE) of the training sequence. This training RMSE is calculated using the training sensor signal and the LSTM-VAE reconstructed signal, and it reflects the noise level in the training performance of the network. The general idea of this fire detection mechanism is as follows: under non-anomalous conditions, the LSTM-VAE, having learned these nominal environmental dynamics from the times series data, can accurately reconstruct the sensor signal (subject to the noise reflected in the RMSE), and

[2]The sensor signals are imported into the LSTM-VAE encoder to calculate the encoded mean, $\boldsymbol{\mu}(t)$ and standard deviation, $\boldsymbol{\sigma}(t)$ of the hidden states (layer). These hidden states are random variables sampled from a Gaussian distribution with the encoded mean and standard deviation. The sampled hidden layer is then decoded to rebuild an estimation of the input signal, $\hat{y}(t)$.

the corresponding fire score is small. The situation changes when a fire anomaly occurs: the environmental dynamics as captured by the sensor having shifted, the signal reconstruction by the LSTM-VAE, $\hat{y}(t)$, will increasingly diverge at every time step from the input signal $y(t)$. The signal reconstruction increasingly deviates from the sensor signal because the former $\hat{y}(t)$ is based on the learned environmental dynamics under the nominal conditions that trained LSTM-VAE, whereas the latter $y(t)$ evolve based on different, anomalous (fire) dynamics. This provides a richer basis for anomaly detection with temporal depth compared with the memoryless threshold-based detection discussed previously. **The growing difference between the y(t)and $\hat{y}$(t)accumulates evidence that an anomaly has occurred and increases the fire score; when this increase becomes large enough and clearly distinguishable from the noise, the fire alarm is triggered.**

We also include a Kalman filter in our detection architecture as shown in Fig. 1 to produce a smoothed fire score $\tilde{S}(t)$ based on which the fire detection decision is made. The sensor noise distribution may have small probability tail events with large values, and this can cause large variations in the fire score, which in turn can result in false alarms. The Kalman filter is appended to mitigate this problem. The details of this entire process are discussed next, and the testing, validation, and benchmarking of this detection method against others are provided in the next sections.

B. LONG SHORT-TERM MEMORY (LSTM) NEURAL NETWORK

LSTM is one of the most effective type of recurrent neural network (RNN) for dealing with sequential data and capturing dependencies over multiple time scales [31]. It has been successful in many applications, including speech recognition and machine translation [32]. LSTM can learn with high accuracy the environmental dynamics of the sensed quantity under nominal conditions, and consequently, it is sensitive to shifts in these dynamics for detecting anomalies and triggering the fire alarm [27]. We provide here a brief introduction to LSTM. The reader is referred to Ref. [31] for more details.

The purpose of LSTM is to deal with the exploding and vanishing gradient problem in the traditional RNN models [33], and it can capture as noted previously time dependencies over multiple time scales including from long past data. A standard LSTM network is shown in Fig. 2 with a hidden state, $h(t)$, a cell state, $C(t)$, and four gates: forget, input, cell, and output gates.

The forget gate, $f(t)$, determines what the LSTM should forget from the previous cell state, $C(t-1)$, as shown in Eq. 2. The input gate, $i(t)$, determines what value should be updated to the cell state $C(t)$, as shown in Eq. 3. The cell gate, $\tilde{C}(t)$, uses current state input and the previous hidden state to generate a candidate for the new cell state, as shown in Eq. 4. The output gate, $o(t)$, determines what should the output be and updates to hidden state, $h(t)$, as shown in Eq. 5. The updated hidden and cell states as Eq. 6 and 7 are used to model the state of the current system and propagate to the next

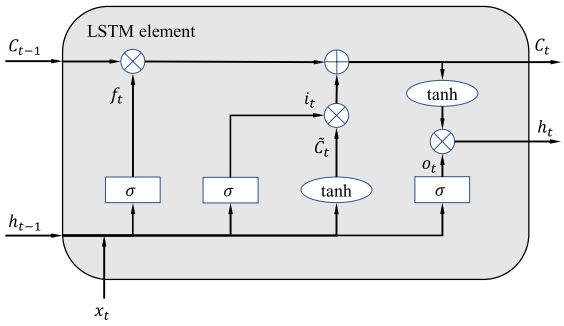

FIGURE 2. The cell structure of a single neuron within the LSTM.

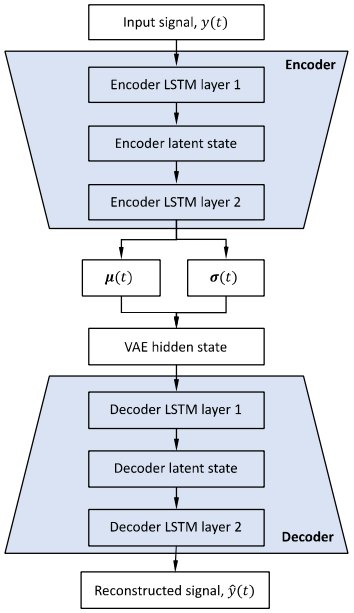

FIGURE 3. Two-layer deep structure of the encoder and decoder in the VAE.

cell for the prediction of system future performance. Here, a two layers deep LSTM network is used for the encoder, and another two-layer deep network for the decoder in the VAE as shown in Fig. 3. Their parameters are trained with the Adam optimizer [34]. Depth in autoencoder networks has been experimentally shown to provide significantly better performance than shallow autoencoders [32].

In the VAE structure shown in Fig. 3, the input of the encoder is the input signal (y) and the outputs are mean value $\boldsymbol{\mu}(t)$ and the standard deviation $\boldsymbol{\sigma}(t)$ of the hidden Gaussian distribution. The input of the two-LSTM-layer deep decoder is the sampled VAE hidden state, and the output is the reconstructed signal ($\hat{y}$). For both the encoder and decoder, the first LSTM layer maps the input to a latent state, and this latent state serves as the input to the second LSTM layer for the output [35]. The details of the VAE structure and process are introduced in the next subsection.

$$f(t) = \sigma(W_f \cdot [h(t-1),x(t)] + b_f) \quad (2)$$

$$i(t) = \sigma(W_i \cdot [h(t-1),x(t)] + b_i) \quad (3)$$

$$\tilde{C}(t) = \tanh(W_C \cdot [h(t-1),x(t)] + b_C) \quad (4)$$

$$o(t) = \sigma(W_o \cdot [h(t-1), x(t)] + b_o) \quad (5)$$
$$C(t) = f_t \times C(t-1) + i_t \times \tilde{C}(t) \quad (6)$$
$$h(t) = o(t) \times \tanh(C(t)) \quad (7)$$

C. VARIATOINAL AUTOENCODER

The Variational autoencoder (VAE) is an unsupervised learning method used to infer potentially complex distributions of the input layer. The VAE is an advanced ML anomaly detection method that is highly robust to signal noise. As noted previously, the use of LSTM can make the detection method more sensitive, however, it can also increase the false alarm rate [27]. We leverage this VAE to reduce the likelihood of false alarms. We examine shortly how the VAE can effectively deal with these issues in our computational experiments. The difference between the VAE input and the reconstructed signal is used to calculate the fire score (Eq. 1). Here, we provide a brief introduction to VAE. The reader is referred to Ref. [36] for more details.

Similar to a standard autoencoder architecture, the VAE consists of an encoder and a decoder. First, the input layer of the sensor signal is encoded as random variables in the hidden state by the encoder. The coded posterior distribution is a diagonal Gaussian distribution as shown in Eq. 8, and the encoder maps the input $\mathbf{x}(t)$ into mean $\boldsymbol{\mu}(t)$ and standard deviation $\boldsymbol{\sigma}(t)$ values of the hidden distribution, as shown in Eq. 9. Second, the hidden state $\mathbf{z}(t)$ is sampled from the previously coded hidden distribution. Third, the decoder reconstructs the signal $\mathbf{x}'(t)$ based on the sampled hidden state, as shown in Eq. 10. Finally, the VAE loss function is calculated as shown in Eq. 11. The first part of VAE loss, $\left(\mathbf{x}(t)-\mathbf{x}'(t)\right)^2$, is sum square error (SSE) of the reconstruction $\mathbf{x}'(t)$ compared with the input layer $\mathbf{x}(t)$. The second part, $\frac{1}{2}(-\log\left(\boldsymbol{\sigma}(t)^2\right) + \boldsymbol{\sigma}(t)^2 + \boldsymbol{\mu}(t)^2 - 1)$, is Kullback-Leibler (KL) divergence between $p(\mathbf{z}(t) \mid \mathbf{x}(t))$ and standard normal distribution. The VAE is trained to provide an accurate reconstruction of the original signal and ensure the hidden state distribution is more likely to be Gaussian. The training data for VAE in our computational experiments are discussed in Section III.

$$p(\mathbf{z}(t) \mid \mathbf{x}(t)) = N(\mathbf{z}(t) | \boldsymbol{\mu}(t), \boldsymbol{\sigma}(t)^2) \quad (8)$$
$$[\boldsymbol{\mu}(t), \boldsymbol{\sigma}(t)] = encoder(\mathbf{x}(t)) \quad (9)$$
$$\mathbf{x}'(t) = decoder(\mathbf{z}(t)) \quad (10)$$
$$\mathcal{L}_{VAE}(\mathbf{x}(t)) = \left(\mathbf{x}(t)-\mathbf{x}'(t)\right)^2 + \frac{1}{2}(-\log\left(\boldsymbol{\sigma}(t)^2\right) + \boldsymbol{\sigma}(t)^2 + \boldsymbol{\mu}(t)^2 - 1) \quad (11)$$

D. KALMAN FILTER

Kalman filtering, also known as linear quadratic estimation (LQE), is an algorithm that uses a series of measurements observed over time and corrupted with statistical noise to estimate the future trend of the temporal signal. The Kalman filter has many applications in guidance and navigation [37], for example, and control theory [38]. We use the Kalman filter to denoise the fire score in our fire detection method with the objective to lower a potential false alarm rate without compromising the improved sensitivity obtained by the LSTM-VAE. Here, we provide a brief introduction to the Kalman filter. The reader is referred to Ref. [39] for more details.

The Kalman Filter is commonly used for system state estimation and prediction with disturbance and measurement noise, as shown in Eq. 12, where x is the state variable, u is the control input, w is the state disturbance, y is the measurement, and v is the measurement noise. A, B, C are the system matrices for the state transition, control, and measurement. The noise is assumed belonging to a centered Gaussian distribution with fixed covariance, R, and Q in Eq. 13.

$$x(t+1) = Ax(t) + Bu(t) + w(t)$$
$$y(t) = Cx(t) + v(t) E(v) = E(w) = 0 \quad (12)$$
$$R = E\left(vv^T\right) Q = E\left(ww^T\right) \quad (13)$$

The parameters of the model are tuned to minimize the error of the posterior state estimation $e = \hat{x} - x$, where $\hat{x}$ denotes the estimated state, and its covariance $P = E(ee^T)$ given the measurements. The posterior state estimation consists of five equations, Eq. 14-18, which can be divided into two parts, the prior prediction, and the correction update. The prediction process propagates the current estimation to the next time step, as shown in Eq. 14 and Eq. 15, where the $\hat{x}'$ and P' denote prior measurement predictions of the state variable and error covariance, respectively.

$$\hat{x}'(t+1) = A\hat{x}(t) + Bu(t) \quad (14)$$
$$P'(t+1) = AP(t)A^T + Q \quad (15)$$

After taking the next measurement, the correction process updates the prior prediction, as shown in Eq. 16 and Eq. 17, where K_t is the Kalman gain shown in Eq. 18:

$$\hat{x}(t) = \hat{x}'(t) + K(t)\left(y(t) - C\hat{x}'(t)\right) \quad (16)$$
$$P(t) = (I - K(t) C) P'(t) (I - K(t) C)^T + K(t) RK(t)^T \quad (17)$$
$$K(t) = P'(t-1)C^T \left(CP'(t-1)C^T + R\right)^{-1} \quad (18)$$

The parameters of the denoising Kalman filter include $(A, B, C, R, Q, x_0', P_0')$, and they are automatically tuned by the expectation-maximization method [40]. We denote the output of the Kalman filter as the smoothed fire score (SFS), based on which the fire alarm is triggered, as discussed in the next subsection.

E. ALARM CRITERIA FOR THE SMOOTHED FIRE SCORE

The alarm criteria are set as thresholds on the smoothed fire score (SFS). Assuming that under normal operating conditions the SFS is normally distributed (diagnostic of this assumption is provided in Appendix A), different thresholds can be set based on this (z-like) score and the corresponding confidence interval. This threshold can be continuously varied, and the corresponding sensitivity and false alarm rate noted. In section VI, we first illustrate the detection results

TABLE 1. Illustrative fire alarm detection threshold values on the smoothed fire score.

Label	Threshold value on the smoothed fire score $\bar{S}(t)$	Confidence interval
lv_1	1.65	90%
lv_2	1.96	95%
lv_3	2.58	99%

of the LSTM-VAE for three commonly used thresholds, with 90%, 95%, and 99% confidence intervals, as shown in Table 1. We then provide more general results by varying the detection threshold.

III. EXPERIMENTS FOR EVALUATING AND BENCHMARKING THE PERFORMANCE OF THE LSTM-VAE AGAINST OTHER FIRE DETECTION METHODS

To evaluate the effectiveness of our LSTM-VAE fire detection method and benchmark its performance against other methods, we adopt two different approaches:

1) Simulation-based experiments: The first approach consists of simulation-based computational experiments with different fire and non-fire scenarios. We leverage the Fire Dynamic Simulation (FDS) computational fluid dynamic (CFD) software to this end. This is a free software developed by National Institute of Standards and Technology (NIST). FDS numerically solves a form of the Navier-Stokes equations appropriate for low speed (Mach number less than 0.3) thermally-driven flow with an emphasis on smoke and heat transports from the fire. FDS has been extensively validated [41]. We design different scenarios and run multiple simulations with FDS under different conditions. We let the LSTM-VAE and other methods detect the fire, or react to the non-fire scenarios, and we calculate high-level statistics for the different fire detection methods to benchmark their performance.
2) Real-world fire and non-fire datasets: The second approach is meant to further test and evaluate the performance of the LSTM-VAE and other methods, and build more confidence in the simulation results. To this effect, we use real-world fire and non-fire datasets provided by the National Institute of NIST [29]. We test the LSTM-VAE and other detection methods on all 69 real datasets in [29], 27 fire and 42 nuisance but non-fire experiments, and we calculate alarm time lag, missed detection rates, false alarms rates, and F1 scores for all the detection methods. To clarify what these experiments are, for example, one real fire dataset used is that of a flaming chair experiment (SDC02) NIST conducted. And one non-fire dataset is another experiment NIST conducted by heating *polyurethane (PU)* foam.

We compare our LSTM-VAE (*lv*) with several competing methods: (1) the standard LSTM (*ls*) anomaly detection [27]; (2) the CUSUM (*cu*) fire detection [13], [14], the EWMA (*ew*) anomaly detection [30], and two currently used fixed temperature detectors with thresholds of 47°C and 58°C each (ft_1 and ft_2) [7]–[9].

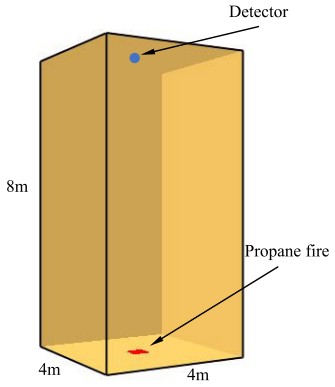

FIGURE 4. The basic fire scenario: propane fire inside the adiabatic room.

In this section, we introduce the simulation-based experiments and two of the real-world datasets. We then discuss the performance metrics used to assess and benchmark their performance. We also develop a relative sensitivity test to statistically compare the performance of the LSTM-VAE against alternative fire detection methods in the simulation-based experiments.

A. SIMULATOIN-BASED COMPUTATIONAL EXPERIMENTS

1) TEST SCENARIOS

As noted previously, we leverage the Fire Dynamic Simulation (FDS) for the computational experiments. We assess the performance of the detection methods in four different scenarios. The first basic fire scenario is a propane fire at the center of the bottom of an adiabatic room as shown in Fig. 4. The temperature sensor is located at the center of the ceiling. The gravity is along the negative z direction. The initial and normal operating condition temperature inside the room is 20°C. The propane fire is simulated with the reaction as Eq. 19 [42], which releases heat, CO_2 and soot ($C_{0.9}H_{0.1}$) as fire signatures, and the fire ignition occurs at $t = 0s$.

$$\begin{aligned} C_3H_8 + 4.81308\,(O_2 + 3.7619N_2) \\ \rightarrow\ 18.10631N_2 + 2.81813CO_2 + 3.98990H_2O \\ +0.20208C_{0.9}H_{0.1} \end{aligned} \tag{19}$$

In addition to this basic fire scenario, we devise three additional scenarios to test and compare the performance of the different detection methods. It is sometimes the case that in real-world situations, an obstacle might exist between the fire source and the detection sensor, which in effect delays mass and heat transports, thus delaying the fire alarm. To simulate this situation, we develop a fire-obstacle scenario by adding a $2m \times 2m \times 2m$ adiabatic obstacle located at the center of the room, as shown in Fig. 5 (a). Furthermore, there can be various disturbances in real-world applications, which might cause false alarms. To simulate such possibilities, we develop two additional scenarios with human presence in the room but no fire to test for false alarm rates. The third scenario consists of human presence modeled as a constant

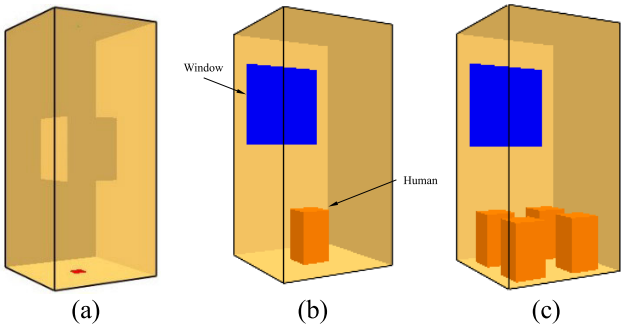

FIGURE 5. Additional scenarios: (a) fire-obstacle scenario; (b) single human presence, light perturbation but no fire scenario; (c) multiple human presence, medium perturbation but no fire scenario.

temperature (37.5°C) volumetric heat source with dimension $1m \times 1m \times 1.8m$ at the bottom center of the room as shown in Fig. 5 (b). We add a ventilation effect in the human presence scenario with a $3m \times 3m$20°C constant temperature window. The fourth scenario is an expansion of the third one, and it consists in placing four heat sources as proxies for humans in the adiabatic "room" as shown in Fig. 5 (c).

To recap, we consider four scenarios in our simulation-based computational experiments to test and compare the performance of the different detection methods. They are: the basic fire scenario; the obstacle fire scenario; a light perturbation but no fire scenario; and a medium perturbation but no fire scenario. The initial temperature is 20°C and the temperature sensor is located at the top center of the room ceiling.

2) FIRE DYNAMIC SIMULATION

We simulate these four scenarios with FDS version 6.7.2 on a Windows 10 platform. The simulations are carried out over $100s$ period with time step $dt = 0.01s$. The simulation results are provided in Fig. 6.

Note for example that in the basic and obstacle fire scenarios, the temperature measurements keep increasing within the 100s period. Note also that the obstacle causes a delay in temperature sensor measurement compared with the basic fire scenario. The temperature disturbances in the non-fire scenarios are rather weak but still captured by the sensors. These results will be fed to the detection methods to assess their alarm time lag and accuracy as discussed shortly.

3) EXPERIMENTS WITH SENSOR SIGNAL NOISE

In real-world applications, the sensor measurements have noise caused by the noisy environment and/or the sensor inherent uncertainty. To model these noise effects in our simulation-based experiments, we computationally add sensor signal noise (ϵ) with standard Gaussian distribution and variance m, as shown in Eq. 20. We examine the effect of different noise levels of 0.4, 1.0,and 2.0°C to test the robustness of our method to sensor noise.

$$\begin{aligned} e &\sim N(0, 1) \\ \epsilon &= m \times e \end{aligned} \tag{20}$$

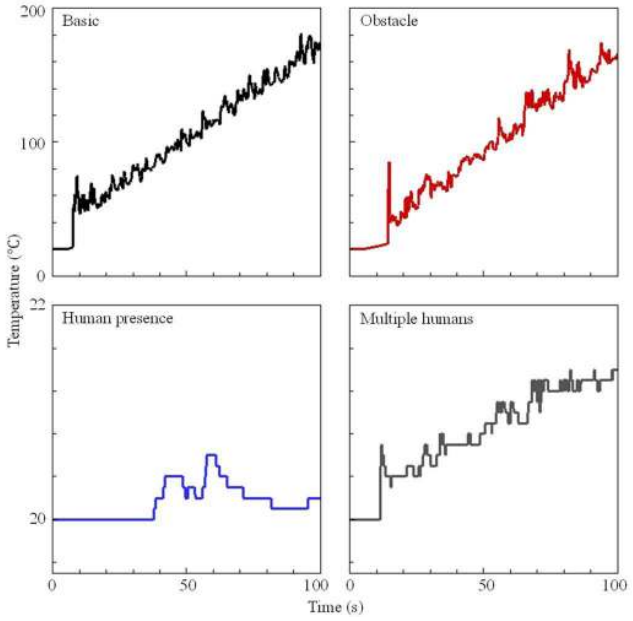

FIGURE 6. FDS fire event simulation of the temperature sensor measurements.

We use the 300 samples before fire ignition ($-3s \leq t \leq 0s$) with this signal noise to train both the LSTM-VAE and LSTM anomaly detection and for the EWMA threshold [30].

To evaluate high-level statistics of the performance of each detection method, we run 100 independent simulations for each scenario with different random numbers for the sensor noise and model initialization. This is further examined in subsection III.D where the relative efficiency t-test is discussed.

B. REAL-WORLD DATASETS

As noted previously, we use in addition to the simulation-based experiments 69 real-world datasets provided by the National Institute of Standards and Technology (NIST) [29] of fire and non-fire experiments. We used these datasets to further evaluate and compare the performance of the different detection methods. We briefly discuss here two of these datasets, and the details of the remaining ones can be found in [29].

1) NIST FLAMING CHAIR FIRE DATASET

We use the temperature sensor signal from the NIST flaming chair fire experiment (SDC02) [29]. The temperature signal is provided in Fig. 7.

The data consists of 515 temperature observations sampled at $0.5Hz$ starting from $t = -646s$ to $t = 382s$. The fire is ignited at $t = 0$s. The time sequence consists of three periods based on the temperature signal: (1) the nominal, non-fire phase, (2) the fire ramp-up phase, and (3) the fire exhaustion phase. In the nominal phase ($-646s \leq t < 0s$), the temperature is around 20°C with sensor signal noise before the fire ignition. During the fire ramp-up phase ($0s \leq t < 206$) the temperature signal

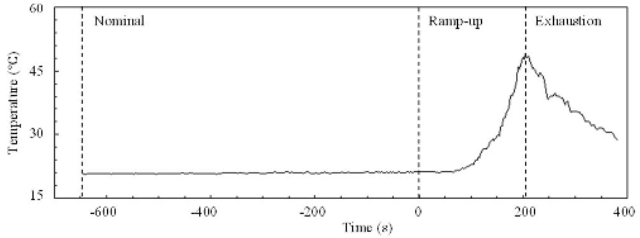

FIGURE 7. Temperature sensor signal of NIST flaming chair experiment.

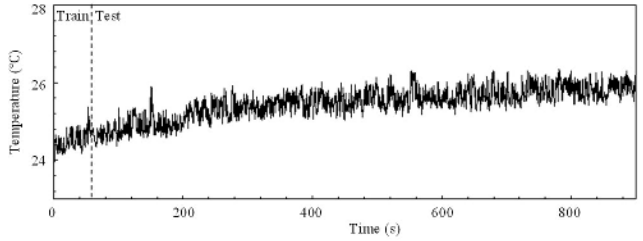

FIGURE 8. Temperature sensor signal from the NIST nuisance experiment (MHN42).

increases and reaches a peak temperature of 51.2°C at $t = 206s$. During the exhaustion phase ($206s \leq t \leq 382s$), the temperature decreases because of the exhaustion of fuel. We use the first 300 samples in the nominal phase to train both the LSTM-VAE and LSTM anomaly detection and to calculate/set the EWMA threshold [30]. The trained LSTM-VAE and other detection methods are then used to supervise the fire event after ignition ($t = 0s$) and to evaluate their performance.

2) NIST NON-FIRE NUISANCE DATASET: HEATED POLYURETHANE (PU) FOAM

In addition to the real fire dataset, we use another dataset (MHN42) from a NIST experiment [29] to evaluate the possibility of false alarms by the different fire detection methods. This was termed a nuisance experiment and the real temperature signal recorded, which shows clear noise as seen in Fig. 8.

The nuisance dataset consists of 4500 data points from $t = 0s$ to $t = 900s$ with a sampling frequency of $5Hz$. The main trend of the temperature increase is the result of a heated *polyurethane* foam in this experiment. We use the first 300 samples to train both the LSTM-VAE and LSTM anomaly detection and to calculate/set the EWMA threshold [30]. The trained ML models are used to supervise the temperature signal and to evaluate the possibility of false alarms.

C. PERFORMANCE METRICS

We propose different performance metrics to benchmark the sensitivity, missed detection rate, and false alarm rate of the fire detection methods in our computational experiments. First, we use the fire alarm time lag, as shown in Eq. 21, to evaluate the sensitivity of fire detection.

$$\Delta t = t_{alarm} - t_{fire} \tag{21}$$

t_{alarm} is the time when the alarm is triggered and t_{fire} the time the fire was ignited. In the simulation-based computational experiments, we examine the mean value and standard deviation of the time lag to assess the overall sensitivity and stability of the fire detection in our 100 independent simulations. In the real-world data experiments, we use this alarm time lag to benchmark and compare the sensitivity of the different fire detection methods. In addition to the alarm time lag, we measure the missed detection rate, the false alarm rate, and the F1 score. We also calculate a criterion used in pattern recognition, namely accuracy (ACC), to further evaluate the type I (alarm without fire) and type II (fire without an alarm) errors. Accuracy (ACC) is defined as the ratio of correct detection (true positive or true negative) to the total number of decisions as shown in Eq. 22:

$$ACC = \frac{truepositive + truenegative}{alldecisions} \tag{22}$$

A true positive occurs when the fire detection triggers the alarm and an actual fire is present, a false positive when the alarm is triggered but no fire is present (type I). Similarly, a true negative is when no alarm is triggered and indeed no fire is present. A false negative (type II) is a missed detection when the alarm is triggered but a fire is present.

D. RELATIVE FIRE DETECTION SENSITIVITY TEST

In the simulation-based computational experiments, we independently conduct 100 simulations for all four scenarios and develop a relative sensitivity test to statistically compare the sensitivity of the LSTM-VAE with the other fire detection methods based on the student's t-distribution. The student's t-distribution is a family of continuous probability distributions that estimate the mean of a Gaussian distribution in situations where the sample size is small and distribution variance is unknown. Given n observations from the distribution, the student's t-distribution with the degree of freedom $\nu = n - 1$ is defined as the distribution of the sample mean relative to the true mean of the distribution divided by the sample standard deviation. Furthermore, the two-sample student's t-distribution and test are used for assessing the statistical significance of the difference between the two means. Here, we apply a one-sided two-sample t-test [43] to determine if our LSTM-VAE method is more sensitive than an alternative fire detection denoted by A. We define the mean of LSTM-VAE fire alarm time lag as μ_L and the mean of method A as μ_A. The null hypothesis can be stated as follows: the mean of LSTM-VAE (μ_L) is identical to the mean of the detection method A (μ_A), H_0:$\mu_A = \mu_L$. The one-sided alternative hypothesis therefore states that the mean of the alarm time lags of A is larger than that of LSTM-VAE, H_a:$\mu_A > \mu_L$, and consequently, the fire detection alarm time lag of LSTM-VAE is smaller and the method is more sensitive. This statistical test will allow us either to (1) reject the null hypothesis and assert that LSTM-VAE provides statistically significant smaller alarm time lag than the alternative fire detection method; or (2) fail to reject the

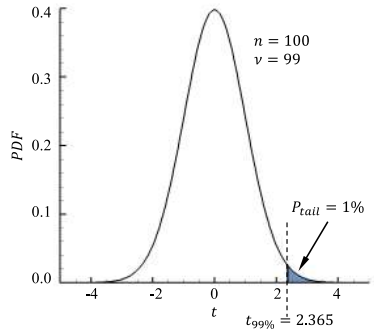

FIGURE 9. One-sided t-test with $\nu = 99$ and $t_{99\%}$.

null hypothesis that LSTM-VAE performs equivalently to the alternative method. The one-sided t-test is defined as follows:

Relative efficiency t-test

$$H_0: \mu_A = \mu_L$$
$$H_a: \mu_A > \mu_L$$
$$\text{Test statistic: } t = \frac{\bar{Y}_A - \bar{Y}_L}{s_p\sqrt{1/N_A + 1/N_L}}$$

with significance level α , rejecting the null hypothesis if $t > t_{1-\alpha}$.

$\bar{Y}_L$, $\bar{Y}_A$ are the calculated mean time lag of LSTM-VAE and the alternative fire detection method (Eq. 23), N_L, $N_A = 100$ are the number of computational experiments for Y_L, Y_A respectively, and s_p is calculated by Eq. 24. The calculated standard deviation s_L, s_A are defined by Eq. 25. Here, we use $n = 100$ iterations in the Monte-Carlo simulation with a degree of freedom $\nu = 99$. We use significance level $\alpha = 1\%$, to reject the null hypothesis, this corresponds to a critical t value $t_{99\%} \approx 2.365$ shown in Fig. 9.

$$\bar{Y} = \frac{1}{N}\sum_{i=1}^{N} Y_i \tag{23}$$

$$s_p = \sqrt{\frac{(N_L - 1)\, s_L^2 + (N_A - 1)\, s_A^2}{N_L + N_A - 2}} \tag{24}$$

$$s = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}\left(Y_i - \bar{Y}\right)^2} \tag{25}$$

IV. RESULTS AND DISCUSSION

In this section, we first present and discuss the results of the simulation-based computational experiments. We then do the same for the two real-world datasets, the flaming chair and heated PU foam. Finally, we provide the results for the entire 69 datasets, and we conclude with a synthesis of the comparative performance analysis of our LSTM-VAE fire detection against other methods. Our objective is to evaluate and benchmark the performance of different detection methods.

A. FIRE DETECTION PERFORMANCE IN THE SIMULATION-BASED COMPUTATIONAL EXPERIMENTS

We first present the results of the different fire detection methods in the simulation-based computational experiments. The mean and standard deviation of alarm time lag (Δt) and the accuracy in the basic fire scenario with different sensor signal noise levels are provided in Fig. 10. For the fire detection method on the x-axis, lv_i stands for the LSTM-VAE with different confidence intervals shown in Table 1, *ls* for the standard LSTM anomaly detection, *cu* for the CUSUM fire detection; *ew* for the EWMA anomaly detection, and ft_1 and ft_2 for the fixed temperature heat detectors of 47°C, 58°C respectively. The mean value of Δt stands for the overall alarm time lag and the sensitivity of the corresponding method. The standard deviation of Δt reflects the stability or consistency of the fire detection method.

First, we note the perfect accuracy (1.0) of all fire detection methods with different noise levels. This indicates that all methods are capable of capturing this basic fire event (no missed detection), and as a result no differentiation can be made based on this metric in this case. The interesting results appear when we examine the top panels in Fig. 10. The mean alarm time lag is the shortest for the LSTM-VAE compared with the other five detection methods for all noise levels. This represents an earlier fire detection on average for our proposed method compared with the others. Increasing sensor noise shrinks this advantage but maintains the edge of the LSTM-VAE. Next, comparing the performance of the LSTM-VAE with different confidence intervals (lv_1, lv_2, and lv_3), we note that the standard deviation of the alarm time lag decreases with increasing the confidence interval. A lower confidence interval, say, with lv_1, allows a faster fire detection but the detection time is more variable than, say, with lv_3. Conversely, with a higher confidence interval, say lv_3, detection takes a bit longer, but the detection time is more robust with less variability than with, say, lv_1. This reflects one aspect of the tradeoff noted previously in fire detection threshold setting, and it provides flexibility to adjust the alarm criteria according to different end-user requirements and preferences. Among the other detection methods, we note that the standard LSTM and EWMA outperform, on average, the other detection methods. But the LSTM exhibits a noticeable standard deviation (e.g., variability of the detection time) when the noise level is small. The CUSUM also exhibits significantly high variability when noise level is high. This suggests that this method is not robust to increasing sensor noise.

Next, we consider the fire-obstacle scenario. The mean and standard deviation of the alarm time lag and the accuracy of the different fire detection methods are provided in Fig. 11.

The results are similar to the previous basic fire event with one notable difference. For example, the accuracy level is again 1.0 for all fire detection methods with different noise levels, which indicates that all methods are capable of detecting the fire despite the obstacle. The trends in the

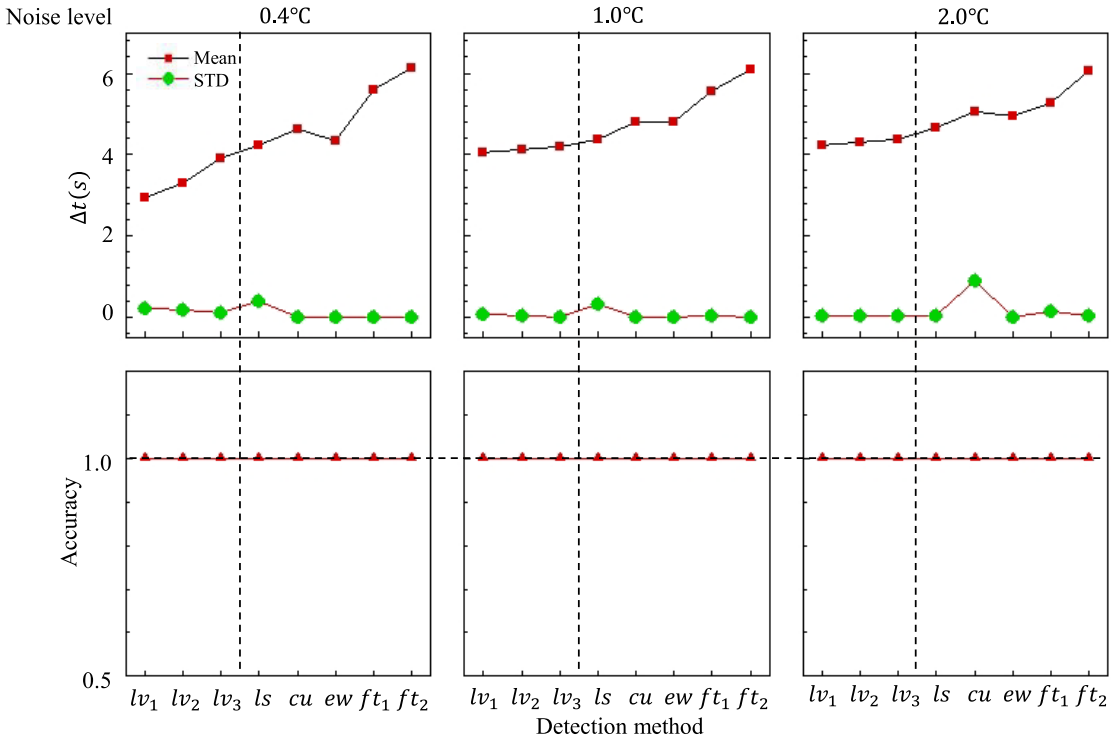

FIGURE 10. Mean and standard deviation (STD) of the alarm time lag and accuracy of the different fire detection methods in the basic fire scenario.

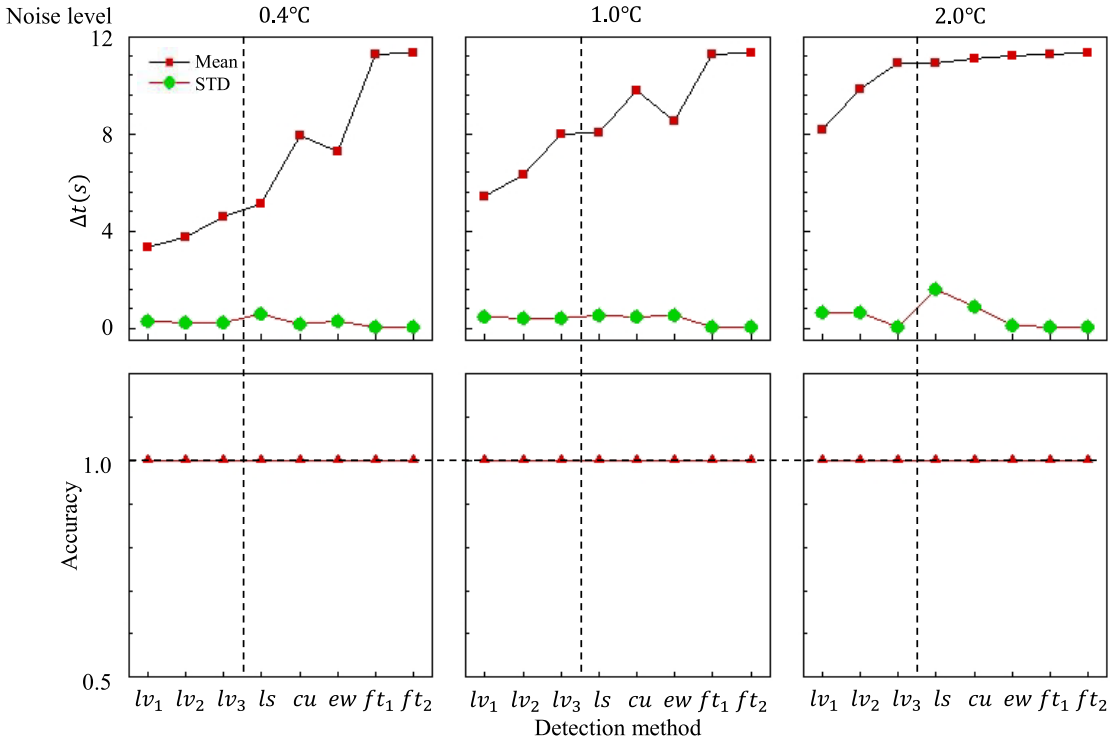

FIGURE 11. Mean and standard deviation (STD) of the alarm time lag and accuracy of different fire detection methods in fire-obstacle scenario.

mean and standard deviation of Δt are similar to the basic fire scenario. However, as shown in Fig. 6, the sensor detects a temperature increase with a time delay compared with the basic fire scenario due to the presence of the obstacle. This translates into an increase in the detection time shown in Fig. 11 (when compared with Fig. 10). Interestingly, this

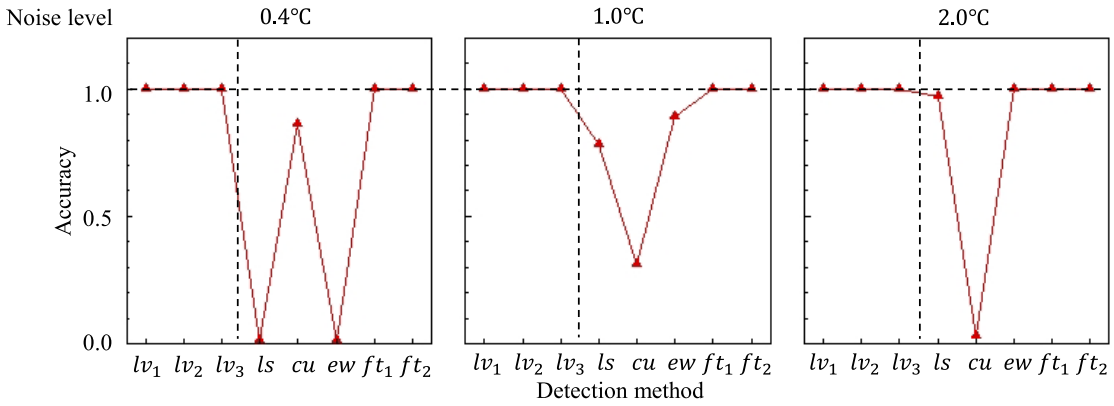

FIGURE 12. Accuracy of various fire detection methods with different sensor noise levels in the multiple human presence scenario.

upward shift in the detection time due to the obstacle is smallest for the LSTM-VAE and largest for the CUSUM and fixed temperature detectors. This suggests an advantage for LSTM-VAE in situations where heat transport and temperature development are delayed to reach the sensors, and the fire is harder to detect. Similarly, with the real-world datasets the fire event is weaker than the propane fire considered here, and the sensitivity differential of the LSTM-VAE will be more pronounced, as will be seen shortly.

In the human presence scenario (no fire), we test for type I errors (alarm with no fire) and measure the accuracy index. Here, accuracy reflects the complement of the false alarm rate since no fires or true positives exist (Eq. 22). The accuracy results of the multiple human presence scenario are provided in Fig. 12. The results of the single human presence scenario are similar but less pronounced, and consequently they are not provided here but are included in Appendix B.

First, we note that there are no false alarms with the LSTM-VAE and the fixed temperature heat detectors, and that these detection methods are robust to varying sensor noise. The false alarm rates of the other detection methods are significant at different noise levels. Interestingly, the accuracy of the LSTM and EWMA improves with increasing sensor noise, whereas that of the CUSUM method degrades significantly. This finding will be further confirmed with the results from the real-world non-fire datasets discussed shortly. The perfect accuracy of the LSTM-VAE at all noise levels in this scenario, and the improvement in the LSTM accuracy with increasing noise levels is due to the way the fire score is calculated. Recall that the fire score (Eq. 1) is normalized by the training RMSE σ_{train}, which can effectively reflect the sensor signal noise level. As the noise level and σ_{train} increase, for a given level of discrepancy between the sensor signal and the reconstructed signal $(y - \hat{y})$ due to the perturbation in this scenario, the magnitude of the fire score decreases. This leads to the improvement in accuracy of the LSTM as noise level increases. Finally, we note in Fig. 12 the roughly similar accuracy performance of the EWMA detection method with the LSTM: poor accuracy at low noise levels, and improvement as noise levels increase. The results real-world datasets will further confirm and add more nuances to this finding.

B. RELATIVE SENSITIVITY RESULTS WITH THE SIMULATION-BASED COMPUTATIONAL EXPERIMENTS

We now examine the results of the one-sided t-test to statistically compare the detection time of the LSTM-VAE with the other methods. We use the basic fire and obstacle fire scenarios for this comparison, and we include here for brevity only the lv_2 (the 95% confidence interval setting for the LSTM-VAE). The results are provided in Table 2. The t-test results for lv_1 and lv_3 are provided in Appendix C, and they are similar to those in Table 2. We use a significance level $\alpha = 1\%$, which corresponds to a critical t value for our 100 computational experiments of $t_{99\%} \approx 2.365$.

TABLE 2. t-values of the relative sensitivity test comparing lv_2 with other methods in basic and obstacle fire scenarios (critical value, $t_{99\%} \approx 2.365$).

	Two detection methods comparison for different noise levels	Noise level		
		0.4°C	1.0°C	2.0°C
Basic fire scenario	lv_2 versus ls	10.92	3.89	2.97
	lv_2 versus cu	13.37	29.32	4.40
	lv_2 versus ew	13.10	4.68	2.89
	lv_2 versus ft_1	52.16	56.21	23.56
	lv_2 versus ft_2	64.47	77.96	43.18
Obstacle-fire	lv_2 versus ls	15.91	13.60	3.05
	lv_2 versus cu	36.73	15.40	3.93
	lv_2 versus ew	15.25	30.16	7.66
	lv_2 versus ft_1	159.99	79.13	18.46
	lv_2 versus ft_2	160.84	79.78	18.98

The results in Table 2 demonstrate that the LSTM-VAE exhibits statistically significant performance improvement over the other detection methods in these two fire scenarios in terms of reduction of the detection time lag. This finding is robust to sensor noise level, and it is generally the case that the improvement is more pronounced in the obstacle scenario than in the basic fire scenario. Table 2 complements the previous results in Fig. 10 and 11, and instead of comparing mean values of detection time lag, we obtain with this relative sensitivity t-test more robust and statistically grounded results to reject the null hypothesis in favor of the alternative, namely

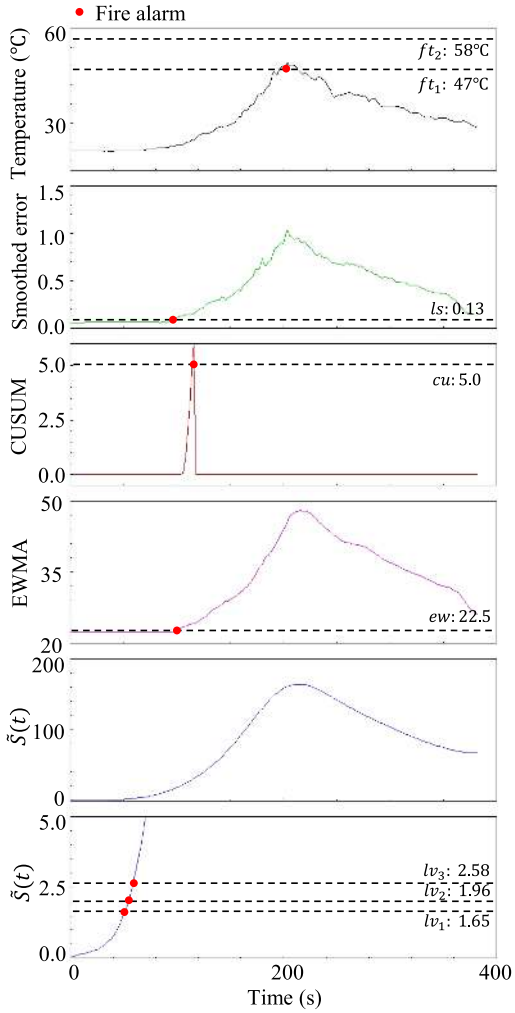

FIGURE 13. Temperature measurement, LSTM smoothed error, CUSUM, EWMA, and smoothed fire score along with the fire alarm triggering point (red circle) of different detection methods in the NIST flaming chair dataset.

that the mean alarm time lag of LSTM-VAE is shorter than that of the alternative fire detection methods.

C. REAL-WORLD DATASETS RESULTS: FLAMING CHAIR AND HEATED PU FOAM

We now present the results for two of the NIST experiments to illustrate and uncover particular details of the performance of different fire detection methods.

For the NIST flaming chair experiment, different quantities are provided in Fig. 13, these include the raw sensor temperature evolution, the standard LSTM smoothed error, the CUSUM, EWMA, and smoothed fire scores. We label the alarm triggering points of different fire detections with red circles on the figure. The alarm time lag of each method is provided in Fig. 14.

First, we observe that the temperature signal (top panel in Fig. 13) remains below 58°C, and as a result the fixed temperature heat detector ft_2 cannot detect this fire and will not trigger the alarm. The fixed temperature heat detector ft_1 with a detection threshold of 47°C triggers the fire alarm

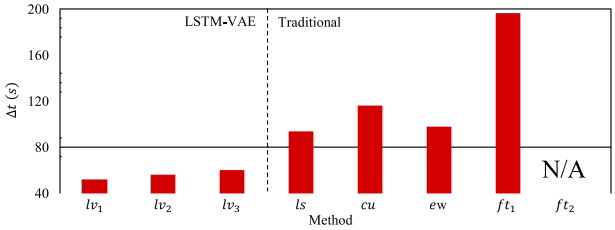

FIGURE 14. Alarm time lag results of different fire detection methods with the NIST flaming chair dataset.

at $\Delta t = 196s$, more than three minutes after the fire started. Second, we note roughly the same performance for the standard LSTM and the EWMA detection methods, with a $\Delta t \approx 95s$, more than twice faster than the detection with the fixed temperature heat detector ft_1. Compared with the LSTM and EWMA, the CUSUM fire detection slightly underperforms with a longer detection time lag $\Delta t = 116s$. The CUSUM triggers the fire alarm at a point when the temperature starts a sharp increase. Finally, we note that the LSTM-VAE outperforms the other detection methods by significantly shorter detection time lag. This is a meaningful achievement if the corresponding false alarm rate is not compromised. We will examine this aspect shortly. A closer look at the smoothed fire score (fifth panel from the top in Fig. 13) is informative. We observe first that the smoothed fire score development is consistent with the temperature increase. Because of the RMSE error (σ_{train}) effect in Eq. 1 and Kalman filter denoising, the fire score curve is smoother than the raw temperature development, and it is significantly more amplified than the quantities shown in the other panels (notice the range on the y-axis of $\tilde{S}$). This, in effect, is the consequence of the LSTM-VAE mechanism, to amplify the main trend in a signal when it differs from the nominal training results and to remain insensitive to noise in the data. This mechanism is responsible for the improvement in detection sensitivity and the shorter detection time lag. However, it comes with a risk of increasing the false alarm rate in the nuisance, no-fire cases. We will examine this possibility next.

For the NIST heated PU foam experiment, different quantities are shown in Fig. 15: these include the raw sensor temperature evolution, the standard LSTM smoothed error, the CUSUM, EWMA, and smoothed fire scores. We label the false alarm with red crosses on the figure.

The previous underperformance of the fixed temperature heat detectors in the previous case of a real fire becomes their advantage in this non-fire event experiment: we note in the top panel of Fig. 15 that the temperature signal in this experiment remains smaller than the threshold values of the fixed temperature heat detectors (47°C and 58°C), and as a result no false alarm is triggered.

The most salient result in Fig. 15 is the fact that the standard LSTM, the CUSUM, and the EWMA methods trigger false alarms in this experiment. In the case of the LSTM, the sensor noise is widely amplified, and when combined with the environmental disturbance (heated PU foam), a false alarm is triggered in this non-fire experiment. In the case of CUSUM

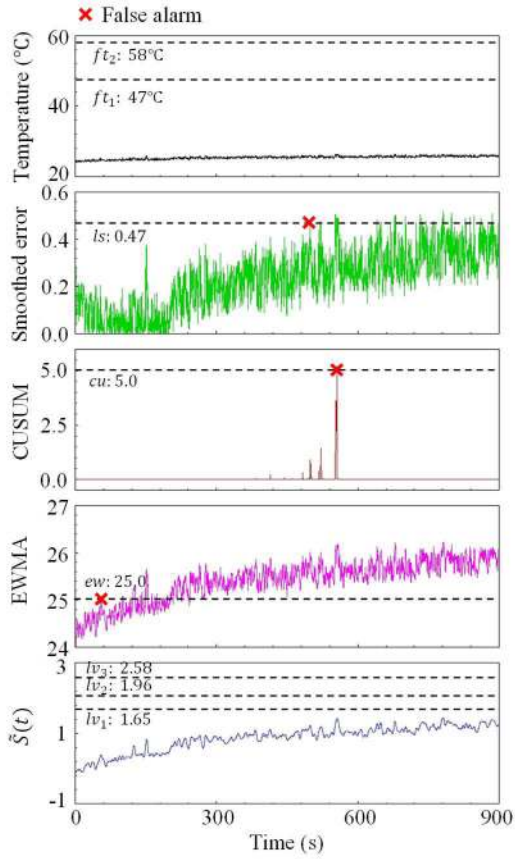

FIGURE 15. Temperature, LSTM smoothed error, CUSUM, EWMA, and smoothed fire score development and false alarm point of different fire detection methods in the NIST heated PU foam experiment.

detection, a related phenomenon occurs: in summing up the abnormal temperature measurements, and adding them on top of those the upward drift in the temperature due to heating the PU foam, the detection threshold is breached and the CUSUM also triggers a false alarm. The previously noted sensitivity of the EWMA fire detection becomes its drawback in this on-fire experiment: it is the fastest to trigger a false alarm. Finally, we note for LSTM-VAE that the smoothed fire score is less noisy than the original temperature signal (the joint effect of the VAE and the Kalman filter). And although this fire score increases with the heating of the PU foam, it remains asymptotically below the 90% threshold lv_1, and as a result no false alarm is triggered.

The LSTM-VAE fire detection method appears to combine high sensitivity and good robustness against false alarms in these two real-world experiments and the previous simulation-based computational experiments. While this is encouraging, significantly more testing and results are needed to develop further confidence in these findings. We present these results next.

D. THE COMPETE REAL-WORLD NIST DATASETS RESULTS

Beyond the results obtained using the two datasets discussed previously, we ran the detection methods on all the 69 real-world datasets provided by NIST (details in Ref. [29]). These include 27 real-fire datasets and 42 nuisance (non-fire) datasets. The fire datasets enable us to assess and benchmark the missed detection rates of the methods considered here, and the nuisance datasets the false alarm rates. Among the fire datasets, NIST conducted experiments with different types of fire: flaming (13 experiments), smoldering (11 experiments), and cooking oil fires (3 experiments). We provide next the collective results for the mean alarm time lag, the missed detection rates, the false alarm rates, and the F-1 scores [44] for all the detection methods. The results on each individual dataset are provided in Appendix D.

There are several important results in Table 3. The most salient are the following:

1) The LSTM-VAE is the only detection method that achieves a perfect F1 score in all 69 experiments (no missed detection and no false alarms). This is the case with all detection thresholds, lv_1, lv_2, and lv_3. The only performance difference between these three settings is the average detection time increases, as expected, from 231s with lv_1 to 371s with lv_3;
2) The smoldering fires are the most challenging to detect. They have the highest missed detections (except for ft_2), and when not missed, they require the longest time to be detected compared with other types of fire;
3) The EWMA is the second-best detection method with an F1 score of 64%, markedly below that of the LSTM-VAE. Interestingly, the EWMA has no missed detection, which reflects an excellent sensitivity, however it has a high false alarm rate with 31 of the 42 nuisance experiments incorrectly triggering the fire alarm (74%);
4) In contrast with the EWMA, both fixed temperature heat detectors have no false alarm rates. However, they have abysmal missed detection rates of 67% and 74% for ft_1 and ft_2 respectively, which can be more dangerous than the shortcoming of the EWMA method;
5) The LSTM method has a 100% false alarm rate, which for all intents and purposes disqualifies it from realistically being considered as a viable fire detection method;
6) The CUSUM method has 19% missed detection and 90% false alarm rate. The former is still dangerous, albeit not as much as the ft_i, and the latter is a significant nuisance. Its F1 score places it in the lowest performing tier of detection methods.

We present next the final performance analysis of the different methods that have a threshold setting for their detection, namely the LSTM-VAE (the threshold setting for the smoothed fire score and the associated confidence interval shown in Table 1), the standard LSTM (Appendix E), the EWMA (Appendix E), and the CUSUM (Appendix E). We vary their threshold and assess their diagnostic ability in terms of missed detection and false alarm rates over the 69 NIST real-world datasets. The results are provided in Fig. 16.

TABLE 3. Mean value of the alarm time lag, missed detection rate, and false alarm rate for all NIST experiments (27 fire and 42 nuisance datasets: the former consists of 13 flaming fires, 11 smoldering fires, and 3 cooking oil fires).

Detection method	Mean value of alarm time lag, seconds (27 datasets)	Missed detection rate, % (27 datasets)	False alarm rate, % (42 datasets)	F1 score
lv_1	Overall: 231.5 Flaming fires: 77.5 Smoldering fires: 414.6 Cooking oil fires: 227.0	Overall: 0% Flaming fires: 0% Smoldering fires: 0% Cooking oil fires: 0%	0%	**100%**
lv_2	Overall: 285.4 Flaming fires: 88.1 Smoldering fires: 530.8 Cooking oil fires: 240.3	Overall: 0% Flaming fires: 0% Smoldering fires: 0% Cooking oil fires: 0%	0%	**100%**
lv_3	Overall: 371.3 Flaming fires: 102.6 Smoldering fires: 720.7 Cooking oil fires: 254.3	Overall: 0% Flaming fires: 0% Smoldering fires: 0% Cooking oil fires 0%	0%	**100%**
ls	Overall: 487.5 Flaming fires: 134.4 Smoldering fires: 936.1 Cooking oil fires: 373.0	Overall: 0% Flaming fires: 0% Smoldering fires: 0% Cooking oil fires 0%	100%	**56%**
cu	Overall: 1350.2* Flaming fires: 502.5* Smoldering fires: 2688.6* Cooking oil fires: 955.0	Overall: 19% Flaming fires: 15% Smoldering fires: 27% Cooking oil fires: 0%	90%	**51%**
ew	Overall: 536.7 Flaming fires: 157.8 Smoldering fires: 1010.1 Cooking oil fires: 443.0	Overall: 0% Flaming fires: 0% Smoldering fires: 0% Cooking oil fires: 0%	74%	**64%**
ft_1	Overall: 2233.1* Flaming fires: 237.0* Smoldering fires: 5423.3* Cooking oil fires: 1440.0*	Overall: 67% Flaming fires: 69% Smoldering fires: 73% Cooking oil fires: 33%	0%	**50%**
ft_2	Overall: 2658.3* Flaming fires: 266.0* Smoldering fires: 5436.7* Cooking oil fires: 1500.0*	Overall: 74% Flaming fires: 77% Smoldering fires: 73% Cooking oil fires: 67%	0%	**41%**

* Missed detection occurred, which can be conceived as $\Delta t \to \infty$. The mean value of the alarm time lag is calculated using only the true positives.

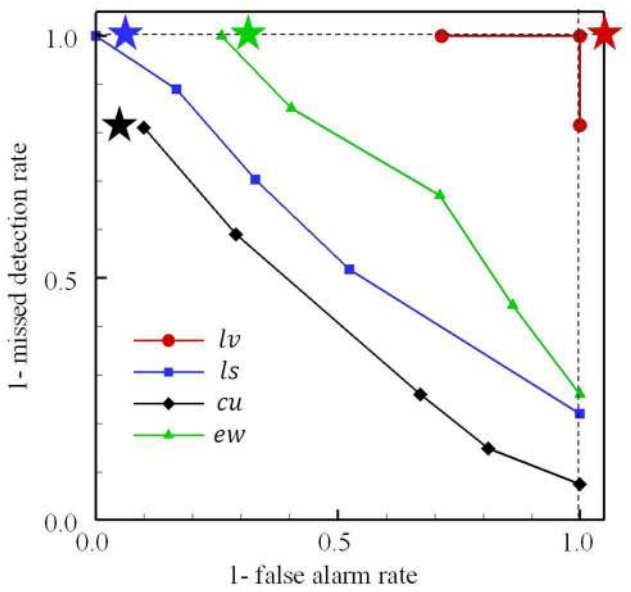

FIGURE 16. Receiver operating characteristic (ROC) curves for different fire detection methods over the 69 NIST real-world datasets. The starred points are the results in Table 3.

The results in Fig. 16 confirm and provide more nuance to the findings in Table 3. They show the Pareto dominance of the LSTM-VAE fire detection compared with the other methods. Similarly, the results show that the EWMA method outperforms the standard LSTM, and that the CUSUM is consistently the worst-in-class fire detection method, and consequently it can be discarded from further consideration as a viable approach to fire detection.

A word of caution is in order: these results, it should be kept in mind, are restricted to the 69 NIST experiments, and are therefore not to be considered general under all possible fire and nuisance circumstances. Nevertheless, the weight of evidence from both the real-world experiments and the simulation-based computational experiments strongly suggests the LSTM-VAE has a robust and significant performance advantage over other approaches to fire detection.

V. CONCLUSION

In this work, we developed a novel fire detection method using deep Long-Short Term Memory (LSTM) neural networks and variational autoencoder (VAE). Our objective was to improve on the performance of other existing detection methods. To evaluate the effectiveness of our method, we developed a set of computational experiments with high-fidelity LES data, and we used datasets from real-world fires (flaming, smoldering, and cooking oil fires) and non-fire experiments. We evaluated and compared the performance of our proposed fire detection with alternative

methods, namely the standard LSTM, CUSUM, exponentially weighted moving average (EWMA), and two currently used fixed-temperature heat detectors with thresholds of 47°C and 58°C each.

The results using both the simulation-based computations and the real-world fire and non-fire experiments are complementary, and they indicate that the LSTM-VAE robustly outperforms the other detection methods on all performance metrics with statistically significant shorter alarm time lags, no missed detection, and no false alarms. The results also identify shortcomings of other detection methods and indicate a clear ranking among them. For example, the CUSUM detection performance degrades with increasing sensor noise, and it has consistently the highest false alarm rate for any level of sensitivity of all other methods in the NIST real-world experiments. The fixed temperature heat detectors, in contrast, have no false alarm rates, but they have poor detection rates with the real fire experiments, which is a more dangerous limitation than the nuisance of false alarms. The standard LSTM is Pareto-dominated by the EWMA, and the latter is largely outperformed by the LSTM-VAE.

This work should be considered in light of its limitations. First, the results reported here are based on computational simulations and 69 real-world experiments. Although significant, the experimental basis for this work does not lend the results any claim to generalizability. Second, the advantages identified in the computational results may be less pronounced if non-propane fires were used (in the sensitivity test) or different perturbations considered (in the false alarm test). Third, the training of the methods with the NIST non-fire experiments was carried out on the first few seconds of the datasets provided. Had other conditions and sensor data been provided for the training (prior to the onset of the nuisance experiment), it may be the case that some false alarms would have been triggered.

Despite these limitations, which affected all methods equally, the weight of evidence indicates that the proposed LSTM-VAE has robust and significant advantages over other approaches to fire detection. In a follow-up work, we extend our method beyond heat detectors to accommodate a host of other sensors, including smoke detectors, CO2 and CO sensors, and we examine sensor data fusion to further improve the performance of this fire detection approach. Beyond the context in which it was used here, in future work, we propose to further generalize our method as an advanced anomaly detection tool for a broad range of applications (with sequential multi-sensor data), including air quality monitoring and other environmental concerns.

APPENDIX

A. VAE RECONSTRUCTION ERROR GAUSSIAN DISTRIBUTION ASSESSMENT

Here, we examine the standard Gaussian distribution using a Kolmogorov-Smirenov normality test (K-S test) [45] and normal probability plot [46] for the fire score of LSTM-VAE in the normal condition. We use the training data of LSTM-VAE in the simulation-based computational experiments as discussed in Section III.A. For this training data, we have 300 temperature measurements in the normal condition without fire with initial temperature at 20°C and sensor noise. Here, we examine the noise level of 0.4, 1.0, and 2.0°C.

For the K-S test, the null hypothesis is the observed data comes from the normal distribution, and we use the p-value to examine the probability of accepting this null hypothesis. The p-value of different noise level condition is listed in Table 4.

TABLE 4. Illustrative fire alarm detection threshold values on the smoothed fire score.

Noise level	0.4°C	1.0°C	2.0°C
p-value	0.964	0.953	0.965

Here, we use a p-value threshold of 0.95 with 95% probability to accept the null hypothesis that the data is under standard Gaussian distribution. The p-value of all these three conditions with different noise levels are larger than 0.95 and the fire score passes the normality test. Then, we use the normal probability plot to examine the normality hypothesis visually, and the normal probability plots of the fire score in these three noise conditions are shown in Fig. 17.

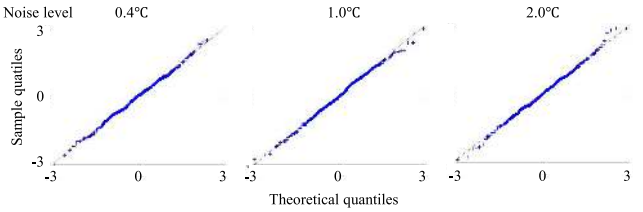

FIGURE 17. The normal probability plots of reconstruction errors in different noise level conditions.

From the normal plot, the fire score probability curve is approximating linear and this proves the normality assumption of the reconstruction error in the normal condition as well.

B. HUMAN PRESENCE SCENARIO SIMULATION-BASED COMPUTATIONAL EXPERIMENTS RESULTS

In this appendix, we present and discuss the result for the human presence scenario simulation-based computational experiments. In the human presence scenario (no fire), we test for type I errors (alarm with no fire) and measure the accuracy index. The accuracy results of the human presence scenario are provided in Fig. 18.

As discussed in Section VI, the results for human scenario is similar to that of the multiple human presence scenario. First, our LSTM-VAE and fixed temperature heat detectors are robust to this human environmental disturbance and achieve excellent accuracy. Second, the LSTM, CUSUM, EWMA fire detections have false alarm errors in this scenario. The LSTM and EWMA have false alarms when the sensor noise is small, say, 0.4°C. The CUSUM fire detection have significant false alarm rate for large sensor noise level, say, 2.0°C.

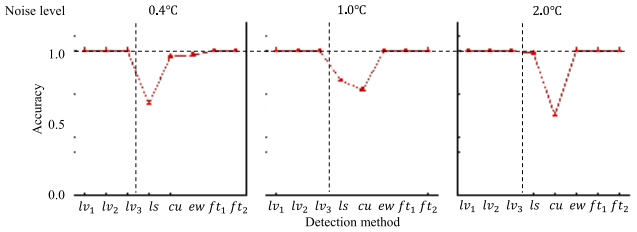

FIGURE 18. Accuracy of various fire detection methods with different sensor noise levels in human presence scenario.

C. THE T-TEST RESULTS OF lv_1 AND lv_3 IN BASIC AND OBSTACLE SCENARIOS IN SIMULATION-BASED COMPUTATIONAL EXPERIMENTS

In this appendix, we present the t-test results of lv_1 and lv_2 in basic and obstacle scenarios compared with LSTM, CUSUM, and EWMA fire detections. These results are shown in Table 5 for lv_1 and Table 6 for lv_3. These results are similar to the t-test result for lv_2 in Table 2. The LSTM-VAE exhibits statistically significant performance improvement over the other detection methods in these two fire scenarios in terms of shorter detection time lag for most of the cases.

TABLE 5. t-values of the relative sensitivity test comparing lv_1 with other methods in basic and obstacle fire scenarios.

	Two detection methods comparison for different noise levels	Noise level		
		0.4°C	1.0°C	2.0°C
Basic fire scenario	lv_1 versus ls	11.75	4.21	4.02
	lv_1 versus cu	13.26	23.12	4.65
	lv_1 versus ew	13.01	4.60	4.00
	lv_1 versus ft_1	56.00	46.10	24.55
	lv_1 versus ft_2	62.48	81.62	41.60
Obstacle-fire	lv_1 versus ls	16.05	14.91	8.63
	lv_1 versus cu	28.77	34.46	9.40
	lv_1 versus ew	15.00	19.31	12.61
	lv_1 versus ft_1	158.81	86.39	38.44
	lv_1 versus ft_2	159.61	87.00	38.95

D. NIST REAL-WORLD TEST RESULTS FOR FIRE AND NUISANCE EXPERIMENTS

In this appendix, we examine the fire detection sensitivity and false alarm errors of our LSTM-VAE against alternative methods by the NIST real-world experiments, including 27 fire test and 42 no fire experiments with household nuisance environmental disturbances. The details of the NIST experiments setup can be found in Ref. [29]. To be noted the 27 fire datasets can be categorized into three categories: (1) flaming fire, (2) smoldering fire, and (3) cooking oil fire. The results of this appendix are summarized in Table 3 in Section VI.

We use the 27 datasets of experiments with fire events to benchmark the sensitivity of our LSTM-VAE, and the alarm time lag results are listed in Table 7 for different methods for various experiments.

Then, we use 42 nuisance experiments to examine the false alarm errors of the fire detection methods, and the false alarm times for different fire detection methods in various NIST nuisance experiments are listed in Table 8.

TABLE 6. t-values of the relative sensitivity test comparing lv_3 with other methods in basic and obstacle fire scenarios.

	Two detection methods comparison for different noise levels	Noise level		
		0.4°C	1.0°C	2.0°C
Basic fire scenario	lv_3 versus ls	8.48	4.98	3.68
	lv_3 versus cu	42.21	32.78	4.34
	lv_3 versus ew	39.25	46.80	3.78
	lv_3 versus ft_1	47.35	57.92	21.82
	lv_3 versus ft_2	62.48	81.62	41.60
Obstacle-fire	lv_3 versus ls	9.50	1.32	0.36
	lv_3 versus cu	65.31	3.78	1.91
	lv_3 versus ew	46.79	3.18	3.48
	lv_3 versus ft_1	139.69	52.74	24.40
	lv_3 versus ft_2	140.52	53.39	26.87

TABLE 7. The alarm time lag (s) results of different methods for various NIST real-word experiments with fire events.

Data	Fire type	lv_1	lv_2	lv_3	ls	cu	ew	ft_1	ft_2
SCD01	Smoldering	467	482	522	1010	N/A	1580	N/A	N/A
SCD02	Flaming	52	56	60	94	116	98	196	N/A
SCD03	Smoldering	246	261	351	606	N/A	611	N/A	N/A
SCD04	Smoldering	698	1490	2710	3450	3520	3500	N/A	N/A
SCD05	Flaming	77	87	92	137	152	142	N/A	N/A
SCD06	Smoldering	278	318	368	718	6050	713	N/A	N/A
SCD07	Flaming	95	103	107	143	155	147	199	211
SCD08	Smoldering	229	249	261	293	377	333	3800	3810
SCD09	Flaming	29	73	93	101	111	263	N/A	N/A
SCD10	Flaming	32	36	38	78	132	98	198	212
SCD11	Smoldering	129	155	175	259	429	249	4360	4380
SCD12	Cooking oil fire	27	35	55	101	791	127	1460	1500
SCD13	Cooking oil fire	58	72	78	108	874	122	1420	N/A
SCD14	Flaming	82	86	90	162	4060	186	N/A	N/A
SCD15	Flaming	230	257	295	305	237	319	355	375
SCD30	Smoldering	41	55	57	59	173	63	N/A	N/A
SCD31	Smoldering	767	989	1010	1010	5760	1030	8110	8120
SCD32	Flaming	54	56	60	98	140	100	N/A	N/A
SCD33	Flaming	54	68	74	112	136	132	N/A	N/A
SCD34	Smoldering	1370	1480	2070	2250	3920	2190	N/A	N/A
SCD35	Flaming	36	42	46	82	138	112	N/A	N/A
SCD36	Flaming	110	116	194	202	N/A	196	N/A	N/A
SCD37	Smoldering	120	126	128	130	N/A	138	N/A	N/A
SCD38	Flaming	111	115	123	135	151	147	N/A	N/A
SCD39	Flaming	46	50	62	98	N/A	112	N/A	N/A
SCD40	Smoldering	216	234	276	512	1280	704	N/A	N/A
SCD41	Cooking oil fire	596	614	630	910	1200	1080	N/A	N/A

* N/A in Table D.I refers to missed detection error

E. CUSUM FIRE DETECTION AND THRESHOLD CALCULATION FOR STANDARD LSTM AND EWMA

We introduce the CUSUM fire detection, which serves as an alternative comparison of our LSTM-VAE. The CUSUM is a sequential analysis technique developed by Page [47]. It is a statistical method used for monitoring the change point detection, which aims at identifying times when the probability distribution of a time series changes. Here, we provide a brief introduction of CUSUM and, the reader is referred to Ref. [47] for more details.

In the CUSUM, the stopping rule first sums up the inputs ($x(t)$), here is the temperature signal, to get a test statistic ($g(t)$), which is then thresholded. The updating process of CUSUMJ is shown in Eq. E. 1, where t_a is the output alarm time, and v, h are the hyperparameters of CUSUM. $v$is the drift hyperparameter to prevent the false alarm caused by the noise in the sensor signal measurement, and h is the alarm threshold hyperparameter to trigger the alarm when $g(t) > h$. The input $x(t)$ is the temperature sequence centralized by the mean of normal operating temperature (T_o) that $x\,(t) = T\,(t) - T_o$, where $T(t)$ is the temperature sequence obtained by NIST data or simulation in our computational experiment. This T_o is the calculated mean value of the ML model training

TABLE 8. The false alarm time results (s) for different fire detection methods for various NIST experiments with nuisance environmental disturbances.

Data	lv_1	lv_2	lv_3	ls	cu	ew	ft_1	ft_2
MHN06	N/A	N/A	N/A	12	34	44	N/A	N/A
MHN07	N/A	N/A	N/A	13	43	26	N/A	N/A
MHN08	N/A	N/A	N/A	10	95	23	N/A	N/A
MHN09	N/A	N/A	N/A	75	58	231	N/A	N/A
MHN10	N/A	N/A	N/A	575	160	740	N/A	N/A
MHN11	N/A	N/A	N/A	208	329	230	N/A	N/A
MHN12	N/A	N/A	N/A	30	117	37	N/A	N/A
MHN13	N/A	N/A	N/A	115	115	117	N/A	N/A
MHN14	N/A	N/A	N/A	98	120	138	N/A	N/A
MHN15	N/A	N/A	N/A	218	335	N/A	N/A	N/A
MHN16	N/A	N/A	N/A	28	30	N/A	N/A	N/A
MHN17	N/A	N/A	N/A	53	51	122	N/A	N/A
MHN18	N/A	N/A	N/A	9	10	9	N/A	N/A
MHN19	N/A	N/A	N/A	105	138	312	N/A	N/A
MHN20	N/A	N/A	N/A	23	27	N/A	N/A	N/A
MHN21	N/A	N/A	N/A	38	40	102	N/A	N/A
MHN22	N/A	N/A	N/A	380	391	N/A	N/A	N/A
MHN23	N/A	N/A	N/A	58	44	163	N/A	N/A
MHN24	N/A	N/A	N/A	125	103	N/A	N/A	N/A
MHN25	N/A	N/A	N/A	68	40	N/A	N/A	N/A
MHN26	N/A	N/A	N/A	583	627	641	N/A	N/A
MHN27	N/A	N/A	N/A	108	266	N/A	N/A	N/A
MHN28	N/A	N/A	N/A	135	141	189	N/A	N/A
MHN29	N/A	N/A	N/A	23	25	141	N/A	N/A
MHN30	N/A	N/A	N/A	108	318	38	N/A	N/A
MHN31	N/A	N/A	N/A	8	10	N/A	N/A	N/A
MHN32	N/A	N/A	N/A	15	13	20	N/A	N/A
MHN33	N/A	N/A	N/A	8	363	13	N/A	N/A
MHN34	N/A	N/A	N/A	70	185	N/A	N/A	N/A
MHN35	N/A	N/A	N/A	98	327	142	N/A	N/A
MHN36	N/A	N/A	N/A	185	226	164	N/A	N/A
MHN37	N/A	N/A	N/A	18	224	25	N/A	N/A
MHN38	N/A	N/A	N/A	368	N/A	N/A	N/A	N/A
MHN39	N/A	N/A	N/A	80	125	60	N/A	N/A
MHN40	N/A	N/A	N/A	38	95	47	N/A	N/A
MHN41	N/A	N/A	N/A	258	N/A	434	N/A	N/A
MHN42	N/A	N/A	N/A	548	506	60	N/A	N/A
MHN43	N/A	N/A	N/A	506	548	49	N/A	N/A
MHN44	N/A	N/A	N/A	1180	N/A	1090	N/A	N/A
MHN45	N/A	N/A	N/A	6020	N/A	N/A	N/A	N/A
MHN46	N/A	N/A	N/A	138	1050	143	N/A	N/A
MHN47	N/A	N/A	N/A	15	193	38	N/A	N/A

* N/A in Table D.II refers to no false alarm error and correctly detect the no fire condition

set.

$$\begin{cases} g(t) = \max(g(t-1) + x(t) - v, 0) \\ t_a = t, when g(t) > h \end{cases} \quad \text{(E.1)}$$

We conduct a grid-search method to tune the hyperparameters of v, h that $0 < v < 20$ with step size 0.5 and $0 < h < 20$ with step size 1.0. The optimal hyperparameters for our problem set up are $v = 1.5$ and $h = 5$ that if we increase them, it will significantly damage the CUSUM sensitivity to fire events and if we decrease them, it will extensively increase the false alarm rate in human presence scenario of simulation-based computational experiments. Also, we tested different methods belonging to the CUSUM category, including CUSUM LS and CUSUM RLS [48], and method of Eq. E.1. provides the best performance for our fire detection problem.

The technical details of the standard LSTM and EWMA can be found in [27] and [30], respectively. Here, we present the method to determine the fire alarm threshold value in our computational experiments for these two methods.

For LSTM anomaly detection, in our simulation-based computational experiments, we use $z = 5.0$ for the smoothed error threshold selection since the authors suggested $2 < z < 10$ in their work [29] and it provides the best performance. If we increase z, the fire detection sensitivity for LSTM anomaly detection is degraded significantly. Contrarily, if we reduce z, the false alarm rate in the human presence and multiple human scenarios decreases. In the real data-based NIST experiments, we vary the threshold for LSTM for a ROC curve in these 69 datasets.

For EWMA fire detection, we use the suggested value $N = 3$ [30] for the threshold calculation in our simulation-based computational experiments. In the NIST real=world dataset test, we vary this N value for different threshold calculations for the ROC curve in Fig. 16.

REFERENCES

[1] X. Huang and L. Du, "Fire detection and recognition optimization based on virtual reality video image," *IEEE Access*, vol. 8, pp. 77951–77961, 2020.
[2] G. Xu, Q. Zhang, D. Liu, G. Lin, J. Wang, and Y. Zhang, "Adversarial adaptation from synthesis to reality in fast detector for smoke detection," *IEEE Access*, vol. 7, pp. 29471–29483, 2019.
[3] Y. Xie, J. Zhu, Y. Cao, Y. Zhang, D. Feng, Y. Zhang, and M. Chen, "Efficient video fire detection exploiting motion-flicker-based dynamic features and deep static features," *IEEE Access*, vol. 8, pp. 81904–81917, 2020.
[4] K. Muhammad, J. Ahmad, I. Mehmood, S. Rho, and S. W. Baik, "Convolutional neural networks based fire detection in surveillance videos," *IEEE Access*, vol. 6, pp. 18174–18183, 2018.
[5] G. Roque and V. S. Padilla, "LPWAN based IoT surveillance system for outdoor fire detection," *IEEE Access*, vol. 8, pp. 114900–114909, 2020.
[6] Z. Liu, "Review of recent developments in fire detection technologies," *J. Fire Protection Eng.*, vol. 13, no. 2, pp. 129–151, May 2003.
[7] Engineers, N. (n.d.). *Heat Detector Design*. Accessed: Jun. 23, 2020. [Online]. Available: https://www.ny-engineers.com/system-components/heat-detector-design
[8] *Automatic Fire Detection: Opal Heat detector, fixed 78C. (n.d.)*. Accessed: Jun. 23, 2020. [Online]. Available: https://www.notifierfiresystems.co.uk/product.asp?id=574
[9] S. Nam, "Predicting response times of fixed-temperature, rate-of-rise, and rate-compensated heat detectors by utilizing thermal response time index," *Fire Saf. J.*, vol. 41, no. 8, pp. 616–627, Nov. 2006.
[10] J. Geiman and D. Gottuk, "Alarm thresholds for smoke detector modeling," *Fire Saf. Sci.*, vol. 7, pp. 197–208, 2003.
[11] M. A. Jackson and I. Robins, "Gas sensing for fire detection: Measurements of CO, CO_2, H_2, O_2, and smoke density in European standard fire tests," *Fire Saf. J.*, vol. 22, no. 2, pp. 181–205, Jan. 1994.
[12] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Comput. Surv.*, vol. 41, no. 3, pp. 1–58, Jul. 2009.
[13] E. Zervas, A. Mpimpoudis, C. Anagnostopoulos, O. Sekkas, and S. Hadjiefthymiades, "Multisensor data fusion for fire detection," *Inf. Fusion*, vol. 12, no. 3, pp. 150–159, Jul. 2011.
[14] O. Sekkas, S. Hadjiefthymiades, and E. Zervas, "A multi-level data fusion approach for early fire detection," in *Proc. Int. Conf. Intell. Netw. Collaborative Syst.*, Nov. 2010, pp. 479–483.
[15] A. Tharwat, "Classification assessment methods," *Appl. Comput. Inform.*, to be published, doi: 10.1016/j.aci.2018.08.003.
[16] Z. Xu and J. Homer Saleh, "Machine learning for reliability engineering and safety applications: Review of current status and future opportunities," 2020, *arXiv:2008.08221*. [Online]. Available: http://arxiv.org/abs/2008.08221
[17] Y. Shi, Z. Lu, R. He, Y. Zhou, and S. Chen, "A novel learning function based on Kriging for reliability analysis," *Rel. Eng. Syst. Saf.*, vol. 198, Jun. 2020, Art. no. 106857.
[18] J. Zhou, L.-Y. Zheng, Y. Wang, and C. Gogu, "A multistage deep transfer learning method for machinery fault diagnostics across diverse working conditions and devices," *IEEE Access*, vol. 8, pp. 80879–80898, 2020.
[19] B. Chinomona, C. Chung, L.-K. Chang, W.-C. Su, and M.-C. Tsai, "Long short-term memory approach to estimate battery remaining useful life using partial data," *IEEE Access*, vol. 8, pp. 165419–165431, 2020.
[20] L. Ren, L. Zhao, S. Hong, S. Zhao, H. Wang, and L. Zhang, "Remaining useful life prediction for lithium-ion battery: A deep learning approach," *IEEE Access*, vol. 6, pp. 50587–50598, 2018.
[21] Z. Gao, C. Ma, J. Zhang, and W. Xu, "Enhanced online sequential parallel extreme learning machine and its application in remaining useful life prediction of integrated modular avionics," *IEEE Access*, vol. 7, pp. 183479–183488, 2019.

[22] C.-F. Tsai, Y.-F. Hsu, C.-Y. Lin, and W.-Y. Lin, "Intrusion detection by machine learning: A review," *Expert Syst. Appl.*, vol. 36, no. 10, pp. 11994–12000, 2009.
[23] A. L. Beam and I. S. Kohane, "Big data and machine learning in health care," *Jama*, vol. 319, no. 13, pp. 1317–1318, Apr. 2018.
[24] K. K. Reddy, S. Sarkar, V. Venugopalan, and M. Giering, "Anomaly detection and fault disambiguation in large flight data: A multi-modal deep auto-encoder approach," in *Proc. Annu. Conf. Prognostics Health Manage. Soc.*, 2016, pp. 1–8.
[25] W. Yan and L. Yu, "On accurate and reliable anomaly detection for gas turbine combustors: A deep learning approach," 2019, *arXiv:1908.09238*. [Online]. Available: http://arxiv.org/abs/1908.09238
[26] S. Fuertes, G. Picart, J.-Y. Tourneret, L. Chaari, A. Ferrari, and C. Richard, "Improving spacecraft health monitoring with automatic anomaly detection techniques," in *Proc. SpaceOps Conf.*, May 2016, p. 2430.
[27] K. Hundman, V. Constantinou, C. Laporte, I. Colwell, and T. Soderstrom, "Detecting spacecraft anomalies using LSTMs and nonparametric dynamic thresholding," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2018, pp. 387–395.
[28] T. Ince, S. Kiranyaz, L. Eren, M. Askar, and M. Gabbouj, "Real-time motor fault detection by 1-D convolutional neural networks," *IEEE Trans. Ind. Electron.*, vol. 63, no. 11, pp. 7067–7075, Nov. 2016.
[29] R. W. Bukowski, R. D. Peacock, J. D. Averill, T. G. Cleary, N. P. Bryner, and P. A. Reneke, "Performance of home smoke alarms, analysis of the response of several available technologies in residential fire settings," NIST, Gaithersburg, MD, USA, Tech. Rep. 1455, 2003.
[30] Z.-G. Zhou and P. Tang, "Improving time series anomaly detection based on exponentially weighted moving average (EWMA) of season-trend model residuals," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Jul. 2016, pp. 3414–3417.
[31] F. A. Gers, N. N. Schraudolph, and J. Schmidhuber, "Learning precise timing with LSTM recurrent networks," *J. Mach. Learn. Res.*, vol. 3, no. 1, pp. 115–143, 2003.
[32] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep Learning*, no. 2. Cambridge, MA, USA: MIT Press, 2016.
[33] R. Pascanu, T. Mikolov, and Y. Bengio, "Understanding the exploding gradient problem," vol. 2, 2012, p. 417, *arXiv:1211.5063*. [Online]. Available: https://arxiv.org/abs/1211.5063
[34] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: http://arxiv.org/abs/1412.6980
[35] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, and L. Antiga, "Pytorch: An imperative style, high-performance deep learning library," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 8026–8037.
[36] C. Doersch, "Tutorial on variational autoencoders," 2016, *arXiv:1606.05908*. [Online]. Available: http://arxiv.org/abs/1606.05908
[37] F. Aghili and C.-Y. Su, "Robust relative navigation by integration of ICP and adaptive Kalman filter using laser scanner and IMU," *IEEE/ASME Trans. Mechatronics*, vol. 21, no. 4, pp. 2015–2026, Aug. 2016.
[38] T. Moore and D. Stouch, "A generalized extended Kalman filter implementation for the robot operating system," in *Intelligent Autonomous Systems 13*. Basel, Switzerland: Springer, 2016, pp. 335–348.
[39] N. Thacker and A. Lacey, "Tutorial: The Kalman filter," Imag. Sci. Biomed. Eng. Division, Med. School, Univ. Manchester, Manchester, U.K., Tech. Rep. 1996-002, 1998, p. 61.
[40] T. K. Moon, "The expectation-maximization algorithm," *IEEE Signal Process. Mag.*, vol. 13, no. 6, pp. 47–60, Nov. 1996.
[41] K. McGrattan, S. Hostikka, R. McDermott, J. Floyd, C. Weinschenk, and K. Overholt, "Fire dynamics simulator user's guide," *NIST Special Publication*, vol. 1019, no. 6, pp. 1–339, 2013.
[42] M. J. Hurley, D. T. Gottuk, J. R. Hall, Jr., K. Harada, E. D. Kuligowski, M. Puchovsky, J. M. Watts, Jr., and C. J. WIECZOREK, *SFPE Handbook of Fire Protection Engineering*. New York, NY, USA: Springer, 2015.
[43] E. L. Lehmann and J. P. Romano, *Testing Statistical Hypotheses*. New York, NY, USA: Springer, 2006.
[44] Y. Sasaki. (2007). *The Truth of the F-Measure*. Accessed: Feb. 18, 2021. [Online]. Available: https://www.cs.odu.edu/~mukka/cs795sum10dm/Lecturenotes/Day3/F-measure-YS-26Oct07.pdf
[45] F. J. Massey, "The kolmogorov-smirnov test for goodness of fit," *J. Amer. Stat. Assoc.*, vol. 46, no. 253, pp. 68–78, Mar. 1951.
[46] J. Chambers, W. Cleveland, B. Kleiner, and P. Tukey, "Accessing distributional assumptions," in *Graphical Methods for Data Analysis*. Boca Raton, FL, USA: CRC Press, 1983, ch. 6, pp. 191–238.
[47] O. A. Grigg, V. T. Farewell, and D. J. Spiegelhalter, "Use of risk-adjusted CUSUM and RSPRTcharts for monitoring in medical contexts," *Stat. Methods Med. Res.*, vol. 12, no. 2, pp. 147–170, Apr. 2003.
[48] F. Gustafsson and F. Gustafsson, *Adaptive Filtering and Change Detection*. Hoboken, NJ, USA: Wiley, 2000.

ZHAOYI XU received the B.S. degree in aerospace engineering from Northwestern Polytechnical University, China, in 2015, and the M.S. degree in aerospace engineering from the Georgia Institute of Technology. He is currently pursuing the Ph.D. degree with the Guggenheim School of Aerospace Engineering, Georgia Institute of Technology. He is also proficient in computational fluid dynamics (CFD). His work on supercritical airfoil optimization, using the genetic algorithm (GA) with ML surrogate was recently published in the AIAA Journal. His research interests include machine learning (ML) in engineering applications, with an emphasis on ML for reliability and system safety, as well as design optimization. He received an Outstanding Undergraduate Thesis Award based on his work on transonic unsteady flow from Northwestern Polytechnical University.

YANJIE GUO received the B.S. degree in aeronautical engineering from the Rensselaer Polytechnic Institute. He is currently pursuing the Ph.D. degree in aerospace engineering with the Georgia Institute of Technology. He worked on the attitude determination and control system for a CubeSat mission for his M.S. degree. His current focus is on active sensing systems that exploit available actuators to improve system state estimation.

JOSEPH HOMER SALEH (Senior Member, IEEE) received the master's degree from Harvard and the Ph.D. degree from MIT, in 2002. He is currently an Associate Professor with the School of Aerospace Engineering, Georgia Institute of Technology. Prior to joining the Georgia Institute of Technology, he served as the Executive Director of the Ford-MIT research partnership. He is the author two books and 78 journal articles. His research interests include three themes, such as spacecraft reliability and multi-state analysis, analytical systems engineering, with a focus on its temporal dimension, and multidisciplinary accident causation and system safety. He is an Associate Fellow of AIAA. He has received several awards from Georgia Tech, including the Outstanding Faculty Award, the Lockheed Dean's Excellence in Teaching Award, and the Class of 1940 Roanne Beard Outstanding Teaching Award. He served as an Associate Editor for *Reliability Engineering and System Safety* and the IEEE TRANSACTIONS ON AEROSPACE AND ELECTRONIC SYSTEMS.