IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 19 May 2022, accepted 1 July 2022, date of publication 14 July 2022, date of current version 20 July 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3190852

RESEARCH ARTICLE

Uncertainty Assessment-Based Active Learning for Reliable Fire Detection Systems

YOUNG-JIN KIM [1] AND WON-TAE KIM [2]

[1]Department of Computer Science Engineering, Korea University of Technology and Education, Cheonan-si 31253, Republic of Korea
[2]Future Convergence Engineering Major, Department of Computer Science and Engineering, Korea University of Technology and Education, Cheonan-si 31253, Republic of Korea

Corresponding author: Won-Tae Kim (wtkim@koreatech.ac.kr)

This work was supported in part by the Institute for Information and Communication Technology Promotion (IITP) under Grant 2019-0-01347, and in part by the Postdoctoral Program of KOREATECH.

ABSTRACT Deep learning technologies, due to their advanced pattern extraction and recognition of high-dimensional data, have been widely adopted into multisensor-based fire detection systems. Since deep learning approaches can generate erroneous predictions due to incomplete training datasets, a retraining process over unseen observations is needed. However, storing a large amount of data from continuous multisensor streams and labeling them to create a retraining dataset are costly and time-consuming. In this paper, we propose an active learning framework based on an informative experience memory that is populated with meaningful retraining data by assessing the uncertainty of the data. In the proposed framework, the deep learning model predicts fire occurrence and estimates model uncertainty by taking advantage of a Bayesian neural network using Monte Carlo dropout. By storing only higher uncertain data points into the fixed-size informative experience memory and querying them to the system managers, the storage and labeling costs are minimized while improving performance. To evaluate our active learning framework with different neural network structures, we develop three Bayesian neural networks based on conventional classification networks, including the feedforward neural network, fully convolutional network, and long short-term memory. We further investigate various uncertainty assessment scoring methods for classification tasks such as entropy, BALD, variation ratios, and mean STD. Experiments on a real dataset show that the Bayesian FCN using the BALD assessment method has the highest performance gain with an F1 score of 0.95, with an improvement of 24% using only 700 data points.

INDEX TERMS Active learning, deep learning, uncertainty assessment, Bayesian neural networks, Monte Carlo dropout, multisensor-based fire detection, reliable fire detection systems.

I. INTRODUCTION

Currently, the most popular and widespread fire detection systems are based on smoke detection [1]. Smoke detection can be performed by either measuring light scattering (photoelectric) or using a physical process (ionization). However, photoelectric and ionization detectors only focus on measuring smoke particles, which are only part of a fire signatures, so they may not offer enough protection from various fire hazards [2]. Additionally, smoke detectors cannot discriminate between smoke particles from actual fires and particles from other normal events, such as burned toasts, cooking fumes, dust from building works, water steam from a shower, and so on [3]. Therefore, smoke sensor-based fire detection systems are highly vulnerable to false alarms [4].

To reduce false alarms while providing early warning, several fire detection systems using multiple sensors have been developed and reported in the literature [2], [5], [6]. They use one or more combinations of heterogeneous sensors, such as smoke sensors, thermal sensors, and different chemical sensors [7]. The detection performance of such multisensor-based fire detection systems depends on decision-making algorithms, also called information fusion methods. Traditionally, information fusion methods have

The associate editor coordinating the review of this manuscript and approving it for publication was Nagarajan Raghavan.

been studied based on the integration of information theories such as fuzzy logic [2], [5] and Dempster–Shafer (DS) theory [6]. These rule-based approaches are developed by designing appropriate logic rules based on experimental engineering knowledge. However, they only consider current observations and ignore historical information in the decision-making process, resulting in increased false alarms due to environmental noise or sensor faults [8]. Moreover, directly finding desired rules over high-dimensional data, such as multivariate time series, is extremely difficult due to their complex temporal and intersensor dependency.

In recent years, more advanced data-driven information fusion methods have been proposed [8]–[13]. Among these data-driven approaches, deep learning (DL) has received significant attention due to its effective pattern extraction and recognition capabilities by training from raw data itself. Wang *et al.* [9] applied three types of artificial neural networks (ANNs), backpropagation (BP), radial basis function (RBF), and probabilistic neural networks (PNNs), for multisensor fire detection. Pack *et al.* [10] also used a feed-forward neural network (FNN) and a convolutional neural network (CNN) to analyze high-dimensional multimodal data, including image and sensor signals. Xu *et al.* [11] proposed deep long short-term memory (LSTM) networks and variational autoencoders (VAEs) to improve the sensitivity and reliability of fire detection. Kim *et al.* [12] proposed a simulation-based learning framework for detecting fires. Then, the three types of encoder–decoder networks based on the FNN, fully convolutional network (FCN), and residual network (ResNet) were designed to provide trustworthy fire detection.

While recent literature shows that DL approaches have achieved significant improvements in the field of multisensor-based fire detection systems, they may generate erroneous predictions due to incomplete training datasets. Especially in the field of fire detection, collecting training data corresponding to various fire and nonfire scenarios in certain environments is not trivial [14]. To overcome this data scarcity problem, several studies have utilized synthetic data generated from fire simulators [12]. However, such simulators only reproduce fire scenarios, not nonfire ones. Collecting training data from nonfire situations, especially those causing false alarms (nuisances), is crucial to increase the reliability of fire detection models. Since nuisance scenarios may arise from numerous causes, such as electrical failures, sensor faults, and unpredictable human behaviors [15], preparing a training dataset of all nuisance scenarios is almost impossible at the development stage.

One solution for solving the aforementioned problem is collecting training data again at the inference stage and retraining DL models with them. However, since the amount of accumulated multisensor data is enormous, the cost required for storage and labeling is expensive. Moreover, the large portion of cumulative multisensor data is related to the normal state of the systems or environments [16], leading to meaningless retraining. Additionally, unlike rule-based approaches, finding data points where DL models struggle to correctly perform classification is difficult due to the nature of the black box decision-making process [17]. To overcome these challenges, we make the following contributions:

(1) We present an active learning framework for reliable fire detection systems that has five fundamental steps: sampling, logging, labeling, retraining, and deploying.
(2) To reduce storing and labeling costs, we introduce fixed-size informative experience memory, which is populated with meaningful retraining data, by assessing the uncertainty of the deep learning model.
(3) To quantify uncertainty, we design three types of Bayesian neural networks based on conventional DL models and evaluate them with various uncertainty assessment scoring functions.

The rest of this paper is organized as follows. Section II shows the overall architecture of the active learning framework for fire detection and presents an informative experience memory-based sampling approach to obtain meaningful retraining data. Section III illustrates the implementation of the Bayesian neural networks based on conventional time-series classification DL models. Section IV presents the details of the evaluation datasets and the results of the comparison. Section V provides concluding remarks.

II. ACTIVE LEARNING FRAMEWORK FOR RELIABLE FIRE DETECTION

In fire detection systems, it is essential to provide early warning while simultaneously ensuring false-alarm immunity. To achieve these goals, data-driven information fusion methods such as deep learning automatically process their decision-making by learning from training datasets. However, collecting all of the required training datasets in the development phase for reliable fire detection may not be trivial due to dynamic changes in environmental factors such as unpredictable human behaviors [3], [12], [15]. Moreover, according to [3], the exact reasons for most false alarms are unknown.

Therefore, DL models must be periodically updated with new observations. The simple approach to updating the DL models is ingesting all of the received multisensor data and labeling it for retraining. However, in this approach, the accumulated multisensor data are too large, leading to large-scale data storage and time-consuming labeling efforts. Moreover, the large portion of received multisensor data is related to the normal state of the environment, resulting in meaningless retraining and excessive costs. To alleviate these problems, in this section, we present an active learning framework for the continuous improvement of fire detection systems, as illustrated in Fig. 1.

A. FRAMEWORK OVERVIEW

In general, fire detection systems consist of multiple sensor nodes, IoT gateways, management servers, and end-users (e.g., firefighters, public residents, etc.) [10], [18]. Many sensor nodes are distributed on certain building locations and are connected to the gateways for transmitting observed

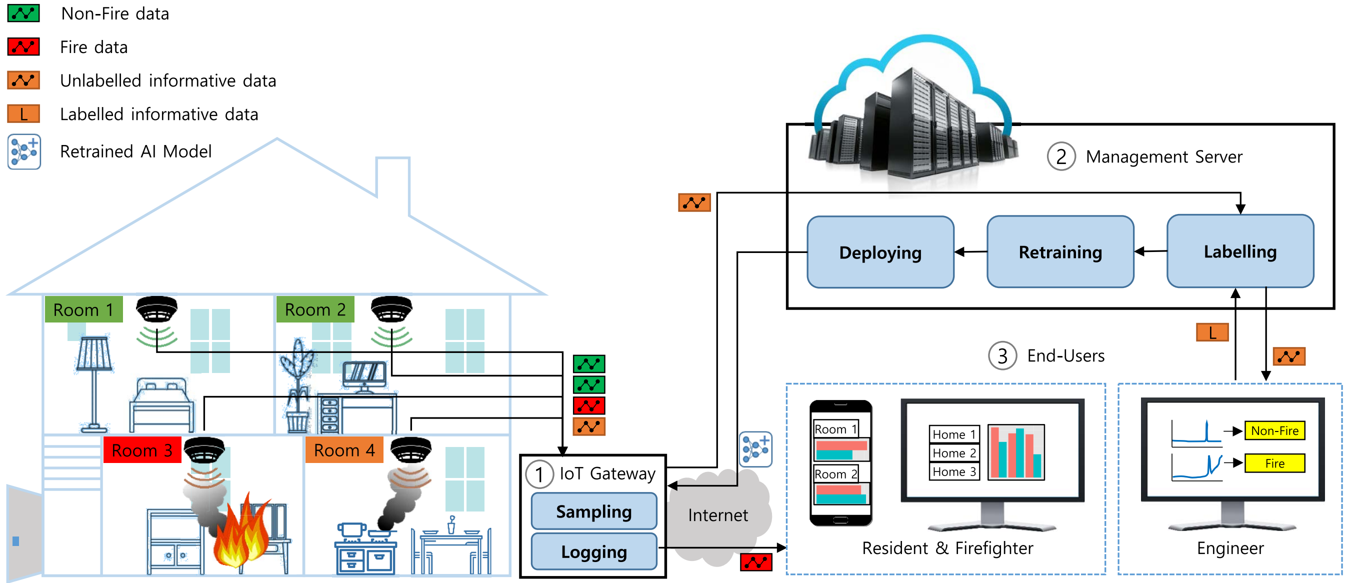


FIGURE 1. Overall active learning framework for continuous improvement of fire detection performance.

multisensor data to management servers or end users. Given this general system architecture, we introduce our active learning framework to continuously improve the reliability of fire detection. There are five fundamental steps, including sampling, logging, labeling, retraining, and deploying.

In the sampling step located in IoT gateways, the informative data points are sampled from incoming continuous streams of multiple sensors. Since a large portion of multisensor data streams are related to a normal state of the environment, most of them have no informational value in terms of retraining DL models. As illustrated in Fig. 1, the multisensor data streams transmitted from Room 1 and Room 2 can be ignored because nothing happened. In Room 3 and Room 4, the incoming data streams are important due to certain events that occur. In particular, data streams from Room 4 are essential to collect as retraining data because the events that occur may be falsely classified as fire. To selectively take these informative data points, we introduce uncertainty assessment based on Bayesian neural networks (BNNs) [19]. The details of the sampling strategy and BNNs are demonstrated in the following subsections. After the sampling step, the sampled data points are logged with timestamps to support labeling and stored in the fixed-size informative experience memory in the IoT gateways. The sampling and logging steps are continuously performed, and stored data in the informative experience memory are transmitted to the management server when the system engineers request the labeling task for retraining DL models.

In the labeling step, the system engineer only needs to label the queried data as fire or nonfire in batches by comparing the timestamp information with historically occurring events. Because only informative data points are sampled during the sampling step, system engineers do not have to struggle to label which region of the time series data corresponds to which classes.

After the labeling task is completed, the management server retrains the DL model associated with a certain IoT gateway with the labeled data and then deploys the retrained DL model to the IoT gateway.

B. INFORMATIVE EXPERIENCE MEMORY FOR ACTIVE LEARNING

In general, active learning can be classified into a stream-based sampling scheme and a pool-based sampling scheme, as illustrated in Fig. 2(a) and Fig. 2(b), respectively. In stream-based sampling [20], [21], informative data are sampled one at a time from data streams, and the sampling determines whether to label and use the data for retraining or to discard them. The problem with the stream-based scheme is that the desired selection criterion is difficult to define due to the exploitation–exploration dilemma [22]–[24]. If the threshold value for the selection criterion is too high, then a large amount of informative data is discarded. On the other hand, if this threshold is too low, a large amount of meaningless data will be collected. In the pool-based sampling scheme [25]–[27], all unlabeled data are gathered in a data pool; then, informative data are searched from this data pool. The problem with the pool-based sampling scheme is that a large amount of storage space is needed, and an exhaustive search in the pool is expensive and time-consuming [28].

To solve the aforementioned problem, we alternatively propose an informative experience memory-based active learning scheme (depicted in Fig. 2(c)). The fixed size informative experience memory is initially populated with the pairs of observation and corresponding uncertainty. After the informative experience memory is completely populated,

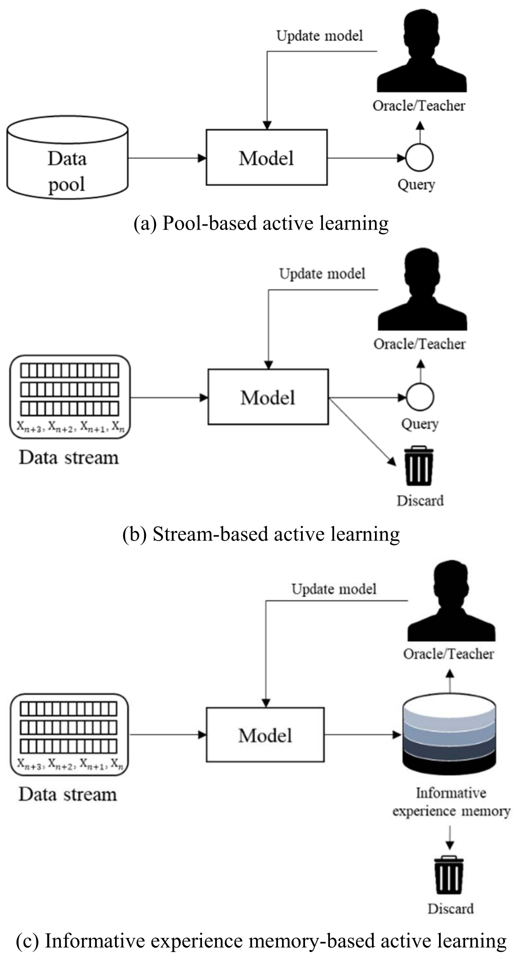


FIGURE 2. Differences between (a) pool-based active learning, (b) stream-based active learning and (c) proposed informative experience memory for active learning.

the stored pair with the lowest uncertainty is replaced with the incoming pair from the data stream when a new observation has a higher uncertainty than the lowest uncertainty in the informative experience memory.

C. UNCERTAINTY QUANTIFICATION BASED ON BAYESIAN NEURAL NETWORK

Deep learning approaches have shown state-of-the-art performance in the field of fire detection systems [7]. However, standard DL models do not provide information about model uncertainty, which describes what the model does not know because training data were not appropriate [29]. Bayesian neural networks (BNNs) can overcome this issue by the probabilistic interpretation of model parameters [30], [31].

To quantify the uncertainty of DL models, this study utilizes the Monte Carlo dropout (MC-Dropout) method, which can be regarded as a Bayesian approximation of the Gaussian process probabilistic models [32]. Basically, dropout is a regularization technique for reducing the overfitting problem by preventing coadaptations of weights [33]. The dropout randomly ignores a certain set of neurons with a

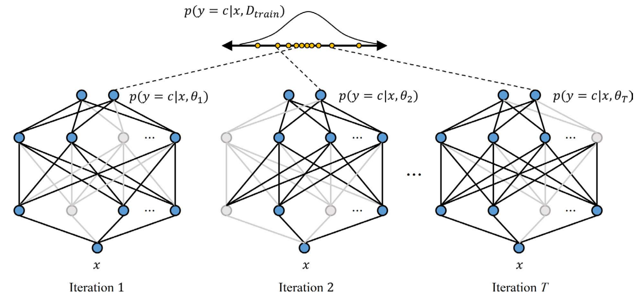


FIGURE 3. Uncertainty quantification based on MC-Dropout.

dropout probability rate only in the training phase. In contrast to dropout, MC-Dropout, which was first proposed by Gal *et al.* [32], uses dropout at both training and inference times. At the training time, the neural networks are trained using the data D_{train} with the usual dropout. At the inference time, the trained neural networks perform T iterative inferences over the same input data, as shown in Fig. 3. At each inference time, the weights of the trained neural networks are randomly dropped out, which results in the weights θ_t. Finally, the T softmax vectors are averaged to obtain the output for a given class c and input x.

$$p(y = c \mid x, D_{train}) = \frac{1}{T}\sum\nolimits_{t=1}^{T} p(y = c \mid x, \theta_t) \quad (1)$$

In the next subsection, we introduce acquisition functions that calculate the uncertainty assessment score of new data points by estimating uncertainty based on these BNNs.

D. ACQUISITION FUNCTIONS

To decide which data points from multisensor data streams update the informative experience memory, the uncertainty assessment score needs to be calculated as acquisition functions. In this paper, we explore four acquisition functions that are appropriate for classification tasks with Bayesian neural networks [34], [35]: predictive entropy, Bayesian active learning by disagreement (BALD), variation ratio, and mean standard deviation (mean STD).

Predictive entropy [36], which is the most ubiquitous measure, updates pools with data points whose predicted classification probability distributions have higher entropy.

$$H[y|x, D_{train}] := -\sum\nolimits_{c}\left(\frac{1}{T}\sum\nolimits_{t} p(y = c \mid x, \theta_t)\right) \cdot log\left(\frac{1}{T}\sum\nolimits_{t} p(y = c \mid x, \theta_t)\right) \quad (2)$$

Bayesian active learning by disagreement (BALD) updates the pool with data points when the mutual information between model predictions and the model parameters is higher. Data points that maximize this acquisition function are data points on which the model is uncertain on average, but there exist model parameters that produce disagreeing predictions with high certainty [37]. BALD consists of the entropy ions minus the conditional entropy ions as

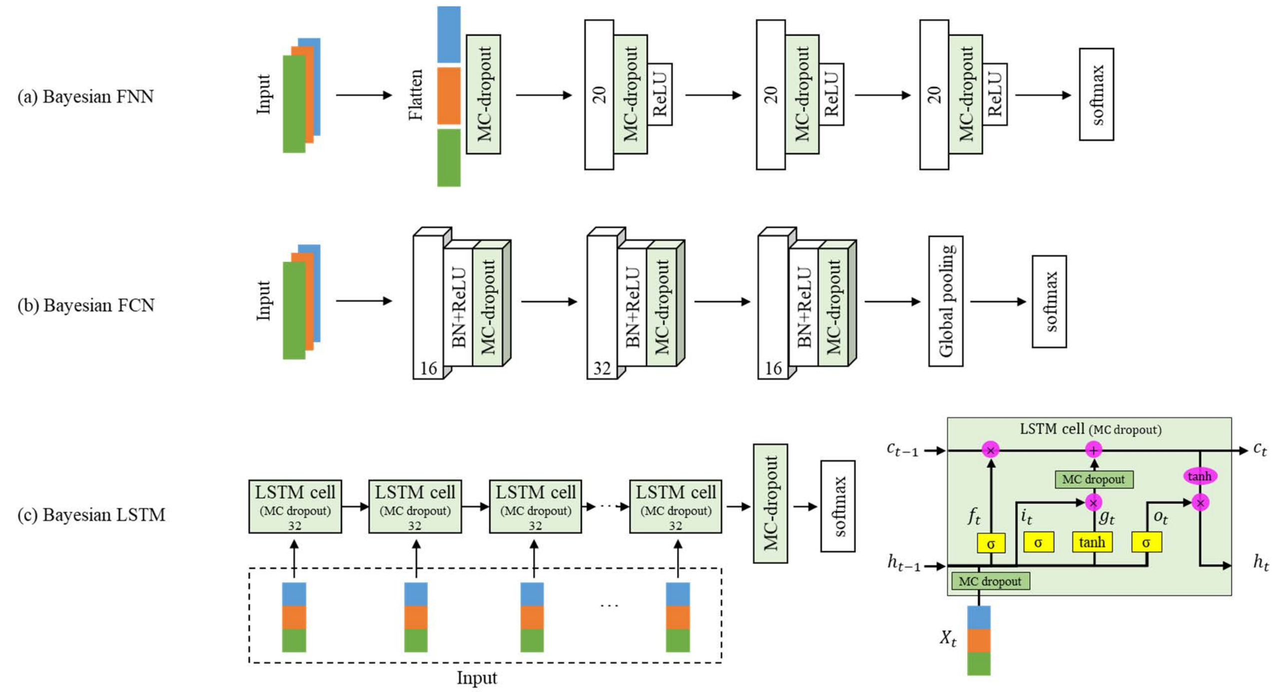


FIGURE 4. Detail structure of (a) Bayesian FNN, (b) Bayesian FCN, and (c) Bayesian LSTM.

formulated by:

$$I[y;\theta|x,D_{train}] := H[y|x,D_{train}] - \frac{1}{T}\sum\nolimits_t \sum\nolimits_c -p \times (y=c\,|\,x,\theta_t)\, logp\,(y=c\,|\,x,\theta_t) \quad (3)$$

The variation ratio is a simple measure of statistical dispersion in nominal distributions. It is defined as the proportion of cases that are not in the mode category, where f_m is the number of predictions falling into the modal class category [38].

$$v := 1 - \frac{f_m}{T} \quad (4)$$

The mean standard deviation (Mean STD) [39] is the standard deviation of the softmax output vectors within T iterative inferences.

$$\sigma = \frac{1}{C}\sum\nolimits_c \sqrt{\frac{1}{T}\sum\nolimits_t \left(p\,(y=c\,|\,x,\theta_t) - \hat{p}\,(y=c)\right)^2} \quad (5)$$

III. IMPLEMENTATION OF BAYESIAN NEURAL NETWORKS FOR MULTISENSOR FIRE DETECTION

In the field of fire detection, various DL models, including feedforward neural networks (FNNs), convolutional neural networks (CNNs), and long short-term memory, are used to detect fire. In this section, we implement BNNs based on these types of network structures. The detailed structures of the BNNs are depicted in Fig. 4.

A. BAYESIAN FEED FORWARD NEURAL NETWORK

A feed forward neural network (FNN) model [40] is the simplest and most traditional form of deep learning architecture. This form of architecture is also known as a multilayer perceptron or fully connected network since all neurons are connected over multiple layers. The implemented Bayesian FNN model contains three hidden layers composed of 20 neurons with a rectified linear unit (ReLU) as the activation function. Since the hidden layer in the Bayesian FNN is fully connected to the previous layer, the original input vector $x \in \mathbb{R}^{N\times M}$ is flattened into a one-dimensional vector, where N is the length of the input and M is the number of multiple sensors. The MC-dropout layer is applied next to the flattened input and each hidden layer.

B. BAYESIAN FULLY CONVOLUTIONAL NETWORK

Convolutional neural network (CNN) architectures have shown promising performance in the field of fire detection. In this work, we explore a fully convolutional network (FCN) since this architecture shows the highest performance on various multivariate time-series datasets [41]. The original FCN model is mainly composed of convolutional layers without pooling and fully connected (FC) layers. In our implementation, the Bayesian FCN contains three one-dimensional convolutional layers with strides equal to 1 and zero padding. The number of filters and kernel size of each convolutional layer are (16, 32, 16) and (8, 5, 3), respectively. Batch normalization and the ReLU activation function are applied to all convolutional layers, and a global average pooling layer is used before the output layer. In the case of CNNs such as FCNs, applying standard MC-dropout over the convolutional layer does not provide uncertainty well since the feature map activations are strongly correlated with each other [42]. Therefore, in our implementation of the Bayesian FCN, we utilize MC-spatial dropout, which drops out the entire feature maps rather than individual activations.

C. BAYESIAN LONG-SHORT TERM MEMORY

Long short-term memory (LSTM), which is a type of recurrent neural network (RNN), has also been widely applied in fire detection fields [11]. LSTM has an advantage for capturing temporal dynamic features based on their feedback connections [43]. In our implementation, the Bayesian LSTM is built on N single LSTM cells that have 32 hidden units. Each LSTM cell is composed of the internal states (cell state and hidden state), an input gate, an output gate, and a forget gate. The operation of the LSTM cell can be represented by the following equations:

$$\begin{aligned} f_t &= \sigma\left(\mathbf{W}_f X_t + \mathbf{U}_f h_{t-1} + \mathbf{b}_f\right) \\ i_t &= \sigma\left(\mathbf{W}_i X_t + \mathbf{U}_\mathbf{i} h_{t-1} + \mathbf{b}_i\right) \\ o_t &= \sigma\left(\mathbf{W}_o X_t + \mathbf{U}_o h_{t-1} + \mathbf{b}_o\right) \\ g_t &= \tanh\left(\mathbf{W}_g X_t + \mathbf{U}_\mathbf{g} h_{t-1} + \mathbf{b}_\mathbf{g}\right) \\ c_t &= f_t \odot c_{t-1} + i_t g_t \\ h_t &= o_t \odot \tanh(c_t) \end{aligned} \tag{6}$$

where σ is the sigmoid function and tanh is the hyperbolic tangent function. $\odot$ is the Hadamard product (elementwise product). **W**, **U**,and **b** are weight parameter vectors that contain 16 hidden units. The input gates i_t and g_t selectively take the information from the current input vector X_t and previous hidden state h_{t-1} to modify the cell state c_t. From the previous cell state c_{t-1}, the forget gate f_t discards irrelevant information. The output gates regulate the flow of information to the output hidden state h_t.

In the case of LSTM, applying standard dropout tends to limit the ability of the networks to retain their memory, hindering their performance [44]. Therefore, in our implementation of the Bayesian LSTM, MC-Dropout is applied to the input gate i_t and recurrent hidden state $i_t g_t$ inside the LSTM cell.

IV. EXPERIMENTAL EVALUATION

A. DATASET DESCRIPTIONS

For the training and evaluation, we use two types of real-world datasets: a fire dataset and a nuisance dataset. The fire dataset [45] is used to initially train the BNNs and evaluate whether the proposed framework detects fire accurately before and after the retraining process. The nuisance dataset [46] is used to retrain the BNNs and evaluate the reliability of the fire detection in scenarios where false alarms may be generated.

The two datasets were created as part of NIST's Home Smoke Alarm Project [47] and contain multivariate time-series data acquired from a manufactured home, as illustrated in Fig. 5.

The fire dataset consists of 11 experiments, each of which has a relative ignition time in seconds. During each

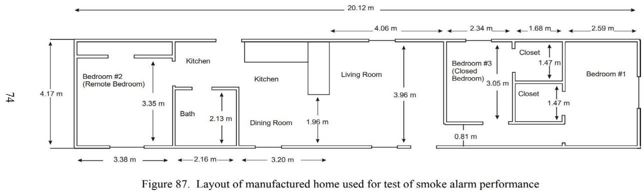


FIGURE 5. Manufactured home scenario for obtaining fire and nuisance datasets.

TABLE 1. Configuration of the fire dataset.

Test Name	Description	Length (s)	Used for
SDC02	Flaming Chair in Living Room	721	Train
SDC05	Flaming Mattress in Bedroom	1044	Train
SDC07	Flaming Mattress in Bedroom	1077	Train
SDC09	Flaming Mattress in Bedroom (Burn Room Door Closed)	1899	Train
SDC10	Flaming Chair in Living Room	1403	Train
SDC15	Flaming Chair in Living Room	1125	Train
SDC33	Flaming Chair in Living Room	1397	Test
SDC35	Flaming Chair in Living Room	1841	Test
SDC36	Flaming Mattress in Bedroom (Burn Room Door Closed)	2919	Test
SDC38	Flaming Mattress in Bedroom	639	Test
SDC39	Flaming Mattress in Bedroom	1071	Test

experiment, the time series data corresponding to temperature, smoke obscuration, and the concentrations of carbon monoxide, carbon dioxide, and oxygen are acquired from the separately located sensors. Since the nuisance dataset only contains temperature, smoke obscurations, and the concentration of carbon monoxide, other measurements, including carbon dioxide and oxygen, are ignored in our experiments. For the initial training of BNNs, we used time-series data of SDC02-SDC15 tests, and the remaining SDC33-SDC39 tests were used to evaluate the detection performance of BNNs. The utilized fire dataset is described in Table 1.

The nuisance dataset contains multivariate time series data that indicate changes in temperature, smoke obscuration, or the concentration of carbon monoxide mainly due to human activities such as cooking or smoking. The lengths of the time series data included in the nuisance dataset vary from 420 seconds to 1469 seconds. In our experiments, all of the nuisance datasets are labeled as the nonfire class. A detailed description of this dataset is shown in Table 2.

To train and evaluate our active learning framework with these two datasets, we extract a total of 44,385 segments by applying a sliding window to the time series data. Among the total number of segments, 15,136 are extracted from the fire dataset, and the remaining 29,294 are extracted from the nuisance dataset. Of the total segments extracted

TABLE 2. Configuration of the Nuisance dataset.

Test Name	Description	Length (s)	Used for
MHN06	Toasted bread, fan off	720	Retrain
MHN07	Toasted bread, fan off (open window)	976	Test
MHN08	225 g of bacon on gas burner, fan off	719	Retrain
MHN09	225 g of bacon on gas burner, fan off	807	Test
MHN10	Broiled pizza, fan off	856	Retrain
MHN11	150 g, (1/3 package) of spaghetti, fan off	900	Retrain
MHN12	300 g (2/3 package) of spaghetti, fan off	1037	Test
MHN13	14 g (1 tbls) butter, fan off	420	Retrain
MHN14	14 g (1 tbls) butter, fan off	600	Test
MHN15	Two cigarettes, fan off	420	Retrain
MHN16	Four hamburgers broiled, fan off	1260	Retrain
MHN17	Four hamburgers broiled, fan on	1152	Test
MHN18	10 tortillas, fan off	1271	Retrain
MHN19	Three hamburgers fried, fan off	728	Retrain
MHN20	Two bagel halves toasted, fan off	592	Retrain
MHN21	Two bagel halves toasted, fan on	732	Test
MHN22	Toasted bread, fan on	720	Retrain
MHN23	Toasted bread, fan on	720	Test
MHN24	225 g (1/2 package) of spaghetti, lid on, fan off	1390	Retrain
MHN25	Three hamburgers fried, fan on (open window)	900	Retrain
MHN26	Three hamburgers fried, fan on	780	Test
MHN27	225 g (1/2 package) of spaghetti, lid on, fan off	1264	Test
MHN28	14 g (1 tbls) margarine, cast iron pan, fan off	420	Retrain
MHN29	14 g (1 tbls) margarine, cast iron pan, fan on	472	Test
MHN30	14 g (1 tbls) butter, cast iron pan, fan on	399	Retrain
MHN31	14 g (1 tbls) butter, cast iron pan, fan off	510	Test
MHN32	Bake/broil pizza, fan off	1469	Retrain
MHN33	Bake/broil pizza, fan on	1421	Test
MHN34	450 g of French fried potatoes, fan off	1329	Retrain
MHN35	Four tea candles, fan off	1072	Retrain
MHN36	225 g of bacon on electric range, fan off	695	Retrain
MHN37	225 g of bacon on electric range, fan on	760	Test
MHN38	Two cigarettes, fan off	485	Retrain
MHN39	Four hamburgers broiled, fan on	1253	Retrain

from the fire dataset, 7,269 segments related to SDC02-SDC15 are used for training Bayesian neural networks, and the remaining 7,867 segments related to SDC33-SDC39 are used for evaluation. In the case of the nuisance dataset, 18,018 segments are used for retraining purposes, and the remaining 11,231 are used to evaluate the reliability of the fire detection. In summary, the ratio of training, retraining, and evaluation data are 16.3%, 40.5% and 43.0% respectively.

TABLE 3. The baseline fire detection performance of the initially trained Bayesian FNN.

Dropout rate	Accuracy	Precision	Recall	F1-score
0.1	0.8118	0.4503	0.9382	0.6085
0.3	0.8867	0.5869	0.9234	0.7176
0.5	0.9176	0.6745	0.9114	0.7752

B. EXPERIMENTAL SETTINGS

For all models, we use the input shape of (60, 3), which corresponds to the window size and number of sensors, respectively. The only preprocessing step in our experiment is min-max normalization on data segments for training, retraining, and evaluation. The Bayesian FNN, Bayesian FCN, and Bayesian LSTM are trained with the Adam optimizer [48] with a learning rate of 0.005. Since the purpose of fire detection is to classify current observations as fire or nonfire, we use cross entropy as a loss function. We choose the best model that achieves the lowest training loss and report its performance on the test set.

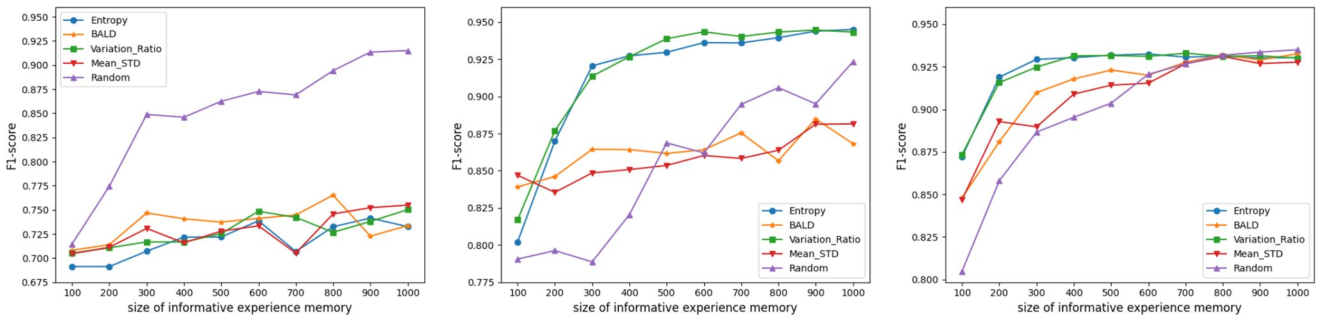
After the training process, the three BNNs are retrained with a learning rate of 0.001 using the data from a fixed size of informative experience memory filled with nuisance data. In the experiments, the size of informative experience memory varying from 100 to 1,000 is evaluated. To quantify the uncertainty of the nuisance data, we use 50 iterative inferences ($T = 50$). Additionally, we experiment with different dropout probability rates varying from 0.1 to 0.5 to evaluate the performance of uncertainty quantification of the BNNs.

Since the lengths of the time series included in the test dataset are different, the number of data points corresponding to the fire and nonfire classes are unbalanced. Therefore, in the experiments, we use precision, recall, and F1 score along with accuracy as evaluation metrics.

C. PERFORMANCE ANALYSIS OF BAYESIAN FNN

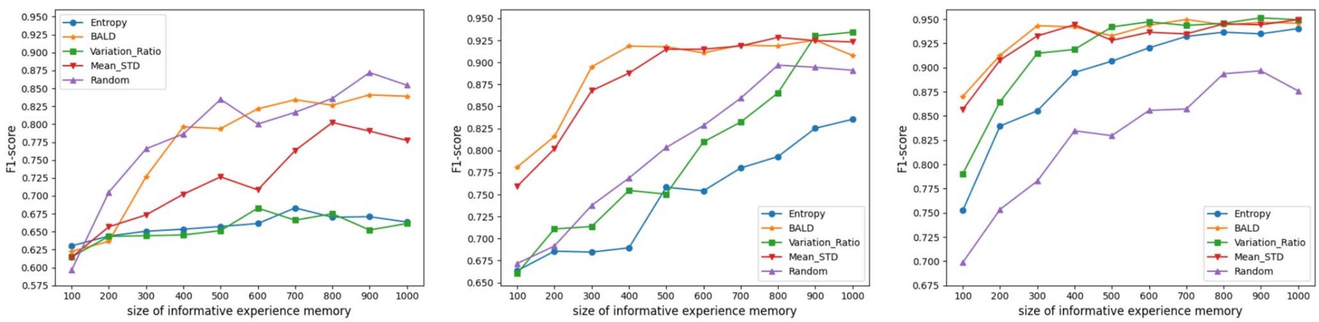
The accuracy, precision, recall, and F1 score of the initially trained Bayesian FNN with different dropout rates are shown in Table 3. As expected, the Bayesian FNN with a dropout rate of 0.1 has high recall scores of 0.9382, and recall scores decrease to 0.9114 with a dropout rate of 0.5.

On the other hand, the precision score increased from 0.4503 to 0.6745 as the dropout rate increased. This indicates that the Bayesian FNN with a low dropout rate tends to overfit the training data, while the Bayesian FNN with a higher dropout rate has a higher generalization capability. From these baseline performances, the increased F1 score of retrained Bayesian FNNs using varying sizes of informative experience memory with various acquisition functions,



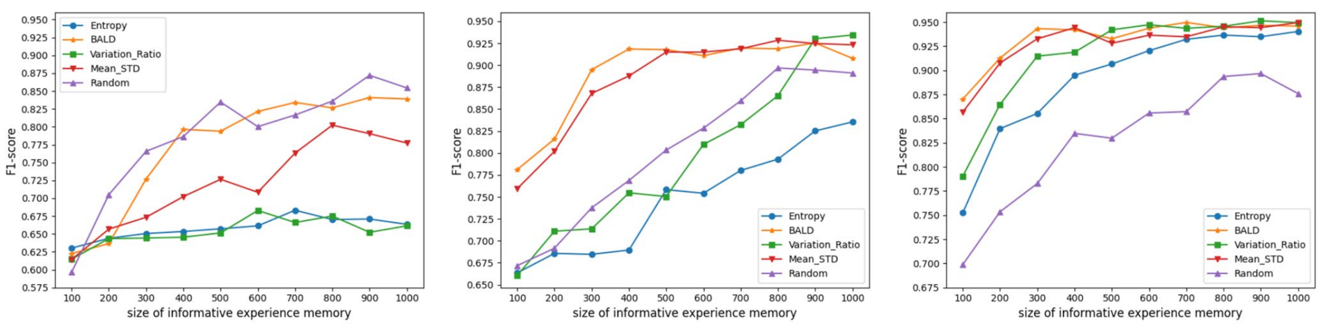
(a) Dropout rate: 0.1 (b) Dropout rate: 0.3 (c) Dropout rate: 0.5

FIGURE 6. The F1-score of Bayesian FNNs with four acquisition functions over different sizes of informative experience memory.



(a) Dropout rate: 0.1 (b) Dropout rate: 0.3 (c) Dropout rate: 0.5

FIGURE 7. The F1-score of Bayesian FCNs with four acquisition functions over different sizes of informative experience memory.



(a) Dropout rate: 0.1 (b) Dropout rate: 0.3 (c) Dropout rate: 0.5

FIGURE 8. The F1-score of Bayesian LSTMs with four acquisition functions over different sizes of informative experience memory.

including entropy, BALD, variation ratio, mean STD, and random sampling, are illustrated in Fig. 6.

Note that Bayesian FNN with Random indicates the retrained Bayesian FNN using randomly and uniformly sampled data points from the nuisance dataset. With a dropout rate of 0.1 (Fig. 6(a)), Bayesian FNNs with all acquisition functions show poorer performance than Random. At the lowest dropout rate of 0.1, it can be seen that uncertainty is not well quantified because the difference between iterative inferences is similar. In the case of the dropout rate of 0.3 in Fig. 6(b), retrained Bayesian FNNs with all acquisition functions achieve a higher F1 score than random acquisition with a large margin until the size of informative experience memory reaches 400. When informative experience memory is more than 500, entropy and variation ratio still maintain high performance, but BALD and mean STD do not achieve significant performance improvement. Among all acquisition functions, Bayesian FNN with Variation Ratio achieves the highest F1 score of 0.9434 at the informative experience memory size of 600.

For a dropout rate of 0.5, all acquisition functions quantify uncertainty well, leading to achieving higher performance than random. However, the increased F1 scores of Bayesian FNNs with all acquisition functions are bounded as 0.93.

Since a higher dropout rate leads to underfitting training and retraining data, the test F1 scores do not increase well.

D. PERFORMANCE ANALYSIS OF BAYESIAN FCN

Table 4 shows the performance of the initially trained Bayesian FCNs with varying dropout rates. Similar to the case of Bayesian FNN, a low dropout rate of 0.1 shows a higher recall score of 0.9422 and a lower precision score of 0.3522. As the dropout rate increases, the recall score decreases, and the precision score increases due to the regularization effect of dropout.

TABLE 4. The baseline fire detection performance of the initially trained Bayesian FCN.

Dropout rate	Accuracy	Precision	Recall	F1-score
0.1	0.7208	0.3522	0.9422	0.5128
0.3	0.8292	0.4759	0.9379	0.6314
0.5	0.8816	0.5744	0.9281	0.7096

As shown in Fig. 7(a), Bayesian FCNs with a dropout rate of 0.1 show similar trends with Bayesian FNNs with a dropout rate of 0.1 due to insufficient capability for quantifying uncertainty.

With a dropout rate of 0.3 (Fig. 7(b)), the F1 scores of retrained Bayesian FCNs with all acquisition functions increased as the size of the informative experiential memory increased. However, the performance improvement rate was very low compared to the Bayesian FNN with a dropout rate of 0.3, and as a result, the highest F1 score reached only 0.9088 despite using the largest memory size of 1000. This suggests that the Bayesian FCN requires a higher dropout rate than the Bayesian FNN for quantifying uncertainty.

Unlike Bayesian FNNs, Bayesian FCNs with a dropout rate of 0.5 (Fig. 7(c)) achieved, on average, a higher F1 score than Bayesian FCNs with a dropout rate of 0.3. This indicates that the Bayesian FCN does not suffer from an underfitting problem even if using a dropout rate of 0.5. The F1 scores of all acquisition functions except for random sampling increased steeply until the size of informative experience memory reached 400. Among all acquisition functions, the BALD acquisition function leads to the highest performance gain with an F1 score of 0.9520 when the size of the informative experience memory is 700.

E. PERFORMANCE ANALYSIS OF BAYESIAN LSTM

The performance of the initially trained Bayesian LSTMs with varying dropout rates is shown in Table 5. Recall scores of Bayesian LSTMs are 0.9647, 0.9349, and 0.9302, respectively, when the dropout rates are 0.1, 0.3, and 0.5, which are relatively high compared to Bayesian FNN and Bayesian FCN. The trend according to the increase in dropout rate is similar to Bayesian FNN and Bayesian FCN.

The retrained Bayesian LSTMs with a small dropout rate of 0.1 show relatively higher performance improvement than the Bayesian FNN and Bayesian FCN, especially when using the BALD acquisition function. Since the dropout operation of Bayesian LSTM recurrently affects the time dimension of input data, the final output of each iterative inference shows a higher difference than Bayesian FNN and Bayesian FCN, leading to better capability of uncertainty quantification. However, the retrained Bayesian LSTMs with a small dropout rate of 0.1 still have poorer performance than Bayesian LSTMs with a dropout rate of 0.3 and 0.5.

TABLE 5. The baseline fire detection performance of the initially trained Bayesian LSTM.

Dropout rate	Accuracy	Precision	Recall	F1-score
0.1	0.7858	0.4189	0.9647	0.5842
0.3	0.8383	0.4903	0.9349	0.6433
0.5	0.8923	0.5998	0.9302	0.7293

In the case of a dropout rate of 0.3, Bayesian LSTMs with BALD and mean STD acquisition functions achieve a higher F1 score than random sampling with a large margin on all sizes of informative experience memory. The F1 scores for BALD and mean STD increase steeply despite using the small size of informative experience memory, which is 0.9176 and 0.9148, respectively, when the size is 500. At the same size of the informative experience memory, on the other hand, an F1 score of variation ratio shows a poor score of 0.7504. This means that the sampled data with the highest uncertainty assessment scores calculated as BALD and mean STD induce meaningful retraining, whereas the data sampled with the variation ratio do not.

For a dropout rate of 0.5, Bayesian LSTM with all acquisition functions outperforms random sampling with a large margin. In the small size of informative experiential memory ranging from 100 to 400, the F1 score obtained the highest score in the order of BALD, Mean STD, Variation Ratio, and Entropy. In particular, Bayesian LSTM with BALD quickly reaches a high F1 score of 0.9433 even when using an informative experience memory of 300.

V. CONCLUSION

In this paper, we propose an active learning framework for fire detection systems. The framework is composed of five fundamental steps, including sampling, logging, labeling, retraining, and deploying, to continuously improve the reliability of fire detection while simultaneously minimizing storage and labeling costs. In the sampling step, the IoT gateway samples informative data points from incoming multisensor data streams and stores them into the informative experience memory. The fixed-size informative experience memory compares the uncertainty between incoming and stored data, maintaining only data with relatively high uncertainty on the fixed-size storage, allowing meaningful retraining with a small number of new data points. To assess the uncertainty of fire detection, we introduced BNNs based on

MC dropout, which provide a probabilistic interpretation of model parameters.

To show the effects of the BNNs with the informative experience memory-based active learning scheme, we implemented the three BNN models, Bayesian FNN, Bayesian FCN, and Bayesian LSTM, and analyzed them on the real-world fire and nuisance dataset. The experimental results show that the F1 score of the BNN model with a dropout rate of 0.3 or higher is improved even with a small amount of information experience memory. In addition, we compared various acquisition functions, including entropy, BALD, variation ratio, and mean STD. As a result, Bayesian FNN showed good performance when entropy or variation ratio was used, whereas Bayesian FCN and Bayesian LSTN performed better when BALD or mean STD was used. In particular, the Bayesian FCN with the BALD acquisition function outperformed the other models with an F1 score of 0.9520, with an improvement of 24% using only 700 data points. Consequently, we expect that the proposed active learning framework will be able to improve the reliability of the fire detection system.

REFERENCES

[1] J. Fonollosa, A. Solórzano, and S. Marco, "Chemical sensor systems and associated algorithms for fire detection: A review," *Sensors*, vol. 18, no. 2, pp. 1–39, Feb. 2018.

[2] R. Sowah, A. R. Ofoli, S. Krakani, and S. Fiawoo, "Hardware module design of a real-time multi-sensor fire detection and notification system using fuzzy logic," in *Proc. IEEE Ind. Appl. Soc. Annu. Meeting*, Oct. 2014, pp. 1–6.

[3] R. Chagger and D. Smith, "The causes of false fire alarms in buildings," BRE Trust, Watford, U.K., Tech. Rep. BC2982, 2014, no. 1.

[4] H. Ishii, T. Ono, Y. Yamauchi, and S. Ohtani, "An algorithm for improving the reliability of detection with processing of multiple sensors' signal," *Fire Saf. J.*, vol. 17, no. 6, pp. 469–484, 1991.

[5] R. A. Sowah, A. R. Ofoli, S. N. Krakani, and S. Y. Fiawoo, "Hardware design and web-based communication modules of a real-time multisensor fire detection and notification system using fuzzy logic," *IEEE Trans. Ind. Appl.*, vol. 53, no. 1, pp. 559–566, Jan. 2017.

[6] Q. Ding, Z. Peng, T. Liu, and Q. Tong, "Multi-sensor building fire alarm system with information fusion technology based on D-S evidence theory," *Algorithms*, vol. 7, no. 4, pp. 523–537, Oct. 2014.

[7] A. Gaur, A. Singh, A. Kumar, K. S. Kulkarni, S. Lala, K. Kapoor, V. Srivastava, A. Kumar, and S. C. Mukhopadhyay, "Fire sensing technologies: A review," *IEEE Sensors J.*, vol. 19, no. 9, pp. 3191–3202, May 2019.

[8] J. Baek, T. J. Alhindi, Y.-S. Jeong, M. K. Jeong, S. Seo, J. Kang, and Y. Heo, "Intelligent multi-sensor detection system for monitoring indoor building fires," *IEEE Sensors J.*, vol. 21, no. 24, pp. 27982–27992, Dec. 2021.

[9] X.-G. Wang, S.-M. Lo, and H.-P. Zhang, "Influence of feature extraction duration and step size on ANN based multisensor fire detection performance," *Proc. Eng.*, vol. 52, pp. 413–421, Jan. 2013, doi: 10.1016/j.proeng.2013.02.162.

[10] J. H. Park, S. Lee, S. Yun, H. Kim, and W.-T. Kim, "Dependable fire detection system with multifunctional artificial intelligence framework," *Sensors*, vol. 19, no. 9, p. 2025, Apr. 2019.

[11] Z. Xu, Y. Guo, and J. H. Saleh, "Advances toward the next generation fire detection: Deep LSTM variational autoencoder for improved sensitivity and reliability," *IEEE Access*, vol. 9, pp. 30636–30653, 2021.

[12] Y.-J. Kim, H. Kim, S. Lee, and W.-T. Kim, "Trustworthy building fire detection framework with simulation-based learning," *IEEE Access*, vol. 9, pp. 55777–55789, 2021.

[13] M. Nakip, C. Güzeliş, and O. Yildiz, "Recurrent trend predictive neural network for multi-sensor fire detection," *IEEE Access*, vol. 9, pp. 84204–84216, 2021.

[14] W. C. Tam, E. Y. Fu, R. Peacock, P. Reneke, J. Wang, J. Li, and T. Cleary, "Generating synthetic sensor data to facilitate machine learning paradigm for prediction of building fire hazard," *Fire Technol.*, to be published, doi: 10.1007/s10694-020-01022-9.

[15] R. Chagger, "The performance of multi-sensors in fire and false alarm tests," BRE Trust, Watford, U.K., Tech. Rep., Mar. 2018. [Online]. Available: https://bregroup.com/projects-reports/the-performance-of-multi-sensors-in-fire-and-false-alarm-tests/

[16] A. A. Cook, G. Misirli, and Z. Fan, "Anomaly detection for IoT time-series data: A survey," *IEEE Internet Things J.*, vol. 7, no. 7, pp. 6481–6494, Jul. 2020.

[17] A. Adadi and M. Berrada, "Peeking inside the black-box: A survey on explainable artificial intelligence (XAI)," *IEEE Access*, vol. 6, pp. 52138–52160, 2018.

[18] W. Chen, C. He, J. Lu, K. Yan, J. Liu, F. Zhou, X. Xu, and X. Hao, "Research and design of distributed fire alarm system of indoor Internet of Things based on LoRa," *Sci. Program.*, vol. 2021, pp. 1–12, Oct. 2021, doi: 10.1155/2021/7462331.

[19] Y. Mae, W. Kumagai, and T. Kanamori, "Uncertainty propagation for dropout-based Bayesian neural networks," *Neural Netw.*, vol. 144, pp. 394–406, Dec. 2021.

[20] J. Chae and S. Hong, "Stream-based active learning with multiple kernels," in *Proc. Int. Conf. Inf. Netw. (ICOIN)*, Jan. 2021, pp. 718–722.

[21] T. Pham, D. Kottke, G. Krempl, and B. Sick, "Stream-based active learning for sliding Windows under the influence of verification latency," *Mach. Learn.*, vol. 111, no. 6, pp. 2011–2036, Jun. 2022, doi: 10.1007/s10994-021-06099-z.

[22] S. Ebert, M. Fritz, and B. Schiele, "RALF: A reinforced active learning formulation for object class recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2012, pp. 3626–3633.

[23] S.-J. Huang, R. Jin, and Z.-H. Zhou, "Active learning by querying informative and representative examples," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 10, pp. 1936–1949, Oct. 2014.

[24] H. T. Nguyen and A. Smeulders, "Active learning using pre-clustering," in *Proc. 21st Int. Conf. Mach. Learn.*, 2004, pp. 79–86.

[25] D. A. Cohn, Z. Ghahramani, and M. I. Jordan, "Active learning with statistical models," *J. Artif. Intell. Res.*, vol. 4, no. 1, pp. 705–712, 1996.

[26] D. D. Lewis and J. Catlett, "Heterogeneous uncertainty sampling for supervised learning," in *Proc. ICML*, vol. 94, 1994, pp. 148–156.

[27] J. Katz-Samuels, J. Zhang, L. Jain, and K. Jamieson, "Improved algorithms for agnostic pool-based active classification," in *Proc. Int. Conf. Mach. Learn.*, Jul. 2021, pp. 5334–5344.

[28] Y. Cheng, Z. Chen, L. Liu, J. Wang, A. Agrawal, and A. Choudhary, "Feedback-driven multiclass active learning for data streams," in *Proc. 22nd ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2013, pp. 1311–1320.

[29] M. Abdar, F. Pourpanah, S. Hussain, D. Rezazadegan, L. Liu, M. Ghavamzadeh, P. Fieguth, X. Cao, A. Khosravi, U. R. Acharya, V. Makarenkov, and S. Nahavandi, "A review of uncertainty quantification in deep learning: Techniques, applications and challenges," 2020, *arXiv:2011.06225*.

[30] H. Wang and D.-Y. Yeung, "Towards Bayesian deep learning: A framework and some existing methods," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 12, pp. 3395–3408, Dec. 2016.

[31] L. V. Jospin, W. Buntine, F. Boussaid, H. Laga, and M. Bennamoun, "Hands-on Bayesian neural networks—A tutorial for deep learning users," 2020, *arXiv:2007.06823*.

[32] Y. Gal and Z. Ghahramani, "Dropout as a Bayesian approximation: Representing model uncertainty in deep learning," in *Proc. 33rd Int. Conf. Mach. Learn. (ICML)*, 2016, pp. 1050–1059.

[33] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.

[34] Y. Gal, R. Islam, and Z. Ghahramani, "Deep Bayesian active learning with image data," in *Proc. 34th Int. Conf. Mach. Learn.*, vol. 70, 2017, pp. 1183–1192.

[35] W. H. Beluch, T. Genewein, A. Nürnberger, and J. M. Kohler, "The power of ensembles for active learning in image classification," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 9368–9377.

[36] C. E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, no. 3, pp. 379–423, Jul. 1948.

[37] N. Houlsby, F. Huszár, Z. Ghahramani, and M. Lengyel, "Bayesian active learning for classification and preference learning," 2011, *arXiv:1112.5745*.
[38] L. Freeman, *Elementary Applied Statistics: For Students in Behavioral Science*. Hoboken, NJ, USA: Wiley, 1965. [Online]. Available: https://books.google.es/books?id=r4VRAAAAMAAJ
[39] M. Kampffmeyer, A.-B. Salberg, and R. Jenssen, "Semantic segmentation of small objects and modeling of uncertainty in urban remote sensing images using deep convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2016, pp. 1–9.
[40] Z. Wang, W. Yan, and T. Oates, "Time series classification from scratch with deep neural networks: A strong baseline," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, May 2017, pp. 1578–1585.
[41] H. I. Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Müller, "Deep learning for time series classification: A review," *Data Mining Knowl. Discovery*, vol. 33, no. 4, pp. 917–963, Jul. 2019.
[42] J. Tompson, R. Goroshin, A. Jain, Y. LeCun, and C. Bregler, "Efficient object localization using convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 648–656.
[43] A. Shrestha and A. Mahmood, "Review of deep learning algorithms and architectures," *IEEE Access*, vol. 7, pp. 53040–53065, 2019.
[44] S. Semeniuta, A. Severyn, and E. Barth, "Recurrent dropout without memory loss," 2016, *arXiv:1603.05118*.
[45] R. D. Peacock, J. D. Averill, R. W. Bukowski, and P. A. Reneke. (Jan. 2011). *NIST Report of Test FR 4016*. [Online]. Available: https://www.nist.gov/el/nist-report-test-fr-4016
[46] T. G. Cleary. (Jan. 2011). NIST Report of Test FR 4019. [Online]. Available: https://www.nist.gov/el/nist-report-test-fr-4019
[47] 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.
[48] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

YOUNG-JIN KIM received the B.S. and M.S. degrees in computer science and engineering and the Ph.D. degree from the Korea University of Technology and Education, Cheonan-si, South Korea, in 2014, 2016, and 2020, respectively. He is currently a Postdoctoral Researcher with the Smart Cyber-Physical Systems Laboratory, Korea University of Technology and Education. His research interests include machine learning and its application, deep learning, artificial intelligence systems design, and digital twin.

WON-TAE KIM received the Ph.D. degree in electronics engineering from Hanyang University, Seoul, Republic of Korea, in 2000. He is currently a Professor with the Department of Computer Science and Engineering, Korea University of Technology and Education, Cheonan-si, Republic of Korea. He had worked at ETRI as a Senior Researcher and the CPS Team Leader, from 2005 to 2015. His research interests include cyber-physical systems, digital twin, intelligence defined networking, wireless/mobile time sensitive networking, dependable computing, and system of systems engineering. He received the ministry award of 100 great achievements of national research and development, in 2014. He actively performs many interesting projects from governmental and industrial funds related with autonomous CPS and digital twin for various industry domains. He also serves as an Academic Committee Member of KICS and IKEEE, respectively. In addition, he supports essential governmental research planning committees, including the committee of Korean digital twin research and development roadmap.