IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2019.DOI

Recurrent Trend Predictive Neural Network for Multi-Sensor Fire Detection

MERT NAKIP[1], (Student Member, IEEE), CÜNEYT GÜZELİŞ[2], OSMAN YILDIZ[3]

[1]Institute of Theoretical and Applied Informatics, Polish Academy of Sciences (PAN), 44100, Gliwice, Poland
[2]Department of Electrical and Electronics Engineering, Yaşar University, 35100, Izmir, Turkey
[3]EDS Elektronik Destek San. ve Tic. Ltd. Şti., 35785, Istanbul, Turkey

Corresponding author: Mert Nakip (e-mail: mnakip@iitis.pl).

ABSTRACT We propose a Recurrent Trend Predictive Neural Network (rTPNN) for multi-sensor fire detection based on the trend as well as level prediction and fusion of sensor readings. The rTPNN model significantly differs from the existing methods due to recurrent sensor data processing employed in its architecture. rTPNN performs trend prediction and level prediction for the time series of each sensor reading and captures trends on multivariate time series data produced by multi-sensor detector. We compare the performance of the rTPNN model with that of each of the Linear Regression (LR), Nonlinear Perceptron (NP), Multi-Layer Perceptron (MLP), Kendall-τ combined with MLP, Probabilistic Bayesian Neural Network (PBNN), Long-Short Term Memory (LSTM), and Support Vector Machine (SVM) on a publicly available fire data set. Our results show that rTPNN model significantly outperforms all of the other models (with 96% accuracy) while it is the only model that achieves high True Positive and True Negative rates (both above 92%) at the same time. rTPNN also triggers an alarm in only 11 s from the start of the fire, where this duration is 22 s for the second-best model. Moreover, we present that the execution time of rTPNN is acceptable for real-time applications.

INDEX TERMS Fire detection, trend prediction, multi-sensor, sensor fusion, recurrent neural networks, machine learning

I. INTRODUCTION

Nowadays, most fire detectors only detect smoke, although the fire is a process that consists of smoke, flame, several kinds of gasses, temperature and humidity [1]. The current percentage of the smoke detectors to the rest in the market of fire detectors is reported as around 60%, however, it is predicted that this percentage will decrease to 50% in 2025 [2]. That is, the fire detectors other than the smoke detectors will comprise half of the market.

The most important success measure for the fire detector is to detect fire situations correctly (in other words, achieving a low False Negative Rate (FNR)). Although the smoke detectors are able to achieve acceptably low FNR, the reason for the decreasing percentage of smoke detectors is that they are weak to differentiate the pollination, cigarette smoke, and smoke of cooking from the fire [3]. That is, most smoke detectors give False Positive alarms while detecting such cases. False Positive alarm means that the detector alarms as fire even the situation is neither fire nor related to a fire. According to Duisburg Fire Department, only 84% of the fire alarm calls that are automatically triggered by the fire detectors are real fire situations [4]. In addition, 44% of the False Positive alarms are triggered due to the aerosols, almost 20% of which are dust and water aerosols [5]. The False Positive alarms waste the time and money of the fire departments and decrease the comfort level of the users during the daily usage of detectors. Reference [6] shows that multi-sensor design with the decision-maker based on machine learning is able to decrease the False Positive Rate (FPR) significantly. However, the performance of such a multi-sensor fire detector needs further improvement in terms of FNR and FPR.

In order to decrease both FNR and FPR, in this paper, we propose the Recurrent Trend Predictive Neural Network (rTPNN) architecture for multi-sensor fire detectors. rTPNN detects the fire based on multi-sensor readings and the predicted trend and level of that readings. That is, rTPNN constructs a relationship between the decision of fire and the trend of the sensor data as opposed to the conventional approach that takes into account only the current value of the sensor data. With possessing this property, rTPNN is robust to the instant changes of the sensor data, which may

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3087736, IEEE Access

be caused by an internal defect of the detector or an external disturbance. The rTPNN model can be used for either the multi-sensor fire detectors or single-sensor detectors. However, rTPNN provides a suitable architecture for multi-sensor fire detectors, which combines the information from different sensors. rTPNN extends the idea of Trend Predictive Neural Network (TPNN) that was firstly proposed by the first two authors of this paper in a conference paper [7]: However, the architecture of rTPNN significantly differs from the architecture of TPNN as follows: 1) rTPNN performs both level prediction and trend prediction while TPNN only performs trend prediction. 2) The recurrent structure of rTPNN enables the learning of all parameters during the training of the model whereas in TPNN the parameters for trend prediction are determined in a trial and error manner. 3) The structure of rTPNN, which learns trend and level prediction, does not perform the extra operations required for TPNN in real-time operation. With these properties, rTPNN is more suitable for real-time applications.

Furthermore, the recurrent internal architecture of the rTPNN model successfully captures the trends in the time series data of each sensor that minimizes the error for the output of rTPNN; hence, it significantly improves the overall prediction performance of the neural network. This paper demonstrates that trend prediction and sensory data fusion properties of the developed rTPNN make a great impact on the fire detection performance of the detectors in terms of sensitivity, specificity, and fire detection time. Note that the rTPNN is a neural network architecture that is suitable and promising for multivariate time series applications beyond multi-sensor fire detection.

In this paper, we evaluate the performance of the multi-sensor fire detector based on rTPNN for the 9 different real-life fire experiments from the publicly available data set [8], [9]. For these experiments, for the multi-sensor detector, we compare the performance of our rTPNN model with that of each of the Linear Regression (LR), Nonlinear Perceptron (NP), Multi-Layer Perceptron (MLP), Kendall-τ combined with MLP (Kendall-MLP), Probabilistic Bayesian Neural Network (PBNN), Long-Short Term Memory (LSTM), and Support Vector Machine (SVM). We also compare the performance of rTPNN with those of the single-sensor detectors. While the used part of the publicly available data set contains 25796 samples, it provides meaningful performance evaluation results as well as a fair comparison of the models. Our results show that the rTPNN model outperforms all of the machine learning models while it achieves 1) prediction performance with high generalization ability, 2) low percentages for both FNR and FPR at the same time, and 3) early detection of fire.

The rest of this paper is organized as follows: In Section II, we state the difference between our work and the state-of-the-art works. In Section III, we introduce the rTPNN model and explain it in detail. In Section IV, we present our results for the performance of the rTPNN and its comparison with the state-of-the-art methods. In Section V, we conclude our study.

II. RELATED WORKS

We now present the relationship between our work and the works in the literature. We may categorize the works in the literature into two categories as the works that aim to detect fire for the outdoor and that for the indoor.

First, in the current literature, there are plenty of works that focus on the fire detection systems for outdoor applications (generally for forest fires). For outdoor fire detection, the first group of works uses image-based techniques. Reference [10] proposes an algorithm that classifies the fire pixels based on the colors. In addition, Reference [11] uses Convolutional Neural Network (CNN) for the recognition of the smoke from the satellite image. CNN-based algorithms are also used to detect the fire for the outdoor in [12], especially for the forests in [13]–[15]. Furthermore, for the forest fire detection from the images, References [16]–[18] use Neural Network (NN)-based algorithms and Reference [19] proposes Hough Feature Transformation based algorithm. The second group of works uses sensors for outdoor fire detection. Reference [20] proposes the usage of the frequencies in the bandwidth of the microwaves, namely plasma frequency for the sensor-based detection of the outdoor fire. Each of References [21]–[24] uses the sensor network that is generally comprised of multiple fire detectors located separately. The sensor networks are also used for the outdoor fire detection with the Artificial Intelligence (AI)-based decision systems, which are specifically designed by using NNs in [25], [26], and Deep Learning model in [27]. Moreover, References [28]–[30] combine the sensor networks with vision-based systems in order to detect the outdoor fire accurately. Besides the many differences with individual works in the category of outdoor fire detection systems, the main difference between our work and the works in this category is as follows: These works aim to detect the fire for the outdoor applications whereas our purpose is to design a novel fire detection system that will be implemented in commonly used fire detectors for the indoor.

The works in the second category focus on fire detection for the indoor applications on which this paper also focuses. In this category, References [31]–[41] detects the fire for the indoor by using only images of the considered spaces, since References [42], [43] combine the images with sensor data. In contrast, we use multiple sensors and combine the data from the sensors by means of rTPNN model to detect fire.

The contrasts between vision-based and sensor-based fire detectors may be examined separately for outdoor and indoor applications. For outdoor applications, vision-based fire detection systems have a higher potential to achieve better performance than sensor-based systems [44]. Supporting this idea, for the detection of forest fires, the accuracy has been reported to be above 95% for vision-based detectors [11], [13], [14] while it is between 80% and 95% for sensor-based detectors [21], [25], [26]. One of the main reasons is that the concentration levels of gases may be too low to detect fire via sensor-based systems. On the other hand,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3087736, IEEE Access

for indoor applications, there are different trade-offs for the performances of vision-based and sensor-based systems in different fire cases. For example, whereas the sensor-based systems may detect smouldering fires earlier than vision-based systems, this may be vice-versa for flaming fires. However, the performance may not be the first criterion for selecting a fire detector for indoor applications. One of the most important trade-offs between sensor-based and vision-based fire detectors is that the sensor-based fire detectors can be used in environments for which privacy is a major issue, such as bedrooms, personal areas, and public restrooms. The cost of the fire detector is another important concern that should be taken into account especially for small indoor spaces.

For the indoor fire detection, Reference [45] examines the performance of the gas sensor based fire detectors, References [46]–[48] use metal oxide gas sensors, and References [49], [50] use array of gas sensors. However, it is known that the reading of the gas sensors are highly dependent on the temperature and the humidity of the environment [51]. In addition, References [52], [53] uses multiple-sensor system, where each sensor module is placed at the separated locations. These systems are effective for the localization of fire source but not for the suitable and efficiently usable for the indoor fire detection. Thus, in this paper, we use the combination of multiple sensors, which may be located at the same position and satisfy the necessary information to the rTPNN-based decision-maker.

Another group of works designs and uses the multi-sensor detector in many different ways for indoor fire detection. The multi-sensor fire detector is used with decision-makers based on simple algorithms in [54], [55], fuzzy logic in [1], [56]–[59], and data fusion in [60], [61]. References [62]–[66] implement the NN-based models for the decision-maker of the multi-sensor fire detector. Whereas all of these works detect fire by using only the reading of the sensor at the current time, we consider the trend of each sensor reading over time via the trend prediction in the novel architectural design of the rTPNN.

Finally, Reference [7] proposes the usage of trend combined with MLP neural network, where the trend is computed based on exponential smoothing. Reference [67] uses the combination of trend with MLP and improves it by computing the trend based on Kendall-τ trending algorithm. In both of these works, the trend is computed distinctly from the decision-maker (MLP in these works); thus, either the computation of trend does not consider the error in fire decision at all or the tuning of the parameters for the trend computation should be performed additionally to the training of decision-maker. In contrast, we propose a novel NN architecture, namely rTPNN, which consists of both the computation of trend and the decision-maker together and designed specifically for fire detection with a multi-sensor detector.

III. RECURRENT TREND PREDICTIVE NEURAL NETWORK

In this section, we first describe the general architecture, the inputs and the output of the recurrent and end-to-end trainable version of TPNN, namely rTPNN[1]. Then, we give the detailed explanation of the internal architecture of each module in rTPNN. Last, we explain how we learn the model parameters of rTPNN.

As shown in Fig. 1, the inputs of the rTPNN model are the reading of the sensors at discrete time k, $\{x_i^k\}_{i\in\{1,\dots,I\}}$ and that at $k-1$, $\{x_i^{k-1}\}_{i\in\{1,\dots,I\}}$, where I is the total number of sensors in the hardware implementation of fire detector. The output of rTPNN, y^k is the state of the fire. At each discrete time k, y^k takes value in the range $[0,1]$, where $y^k=0$ advocates that there is no fire and $y^k=1$ advocates otherwise. In the practical usage of rTPNN, if the value of y^k is greater than a threshold γ, we state that there is fire[2]. Otherwise, we state that there is no fire.

In the architecture of rTPNN in Fig. 1, there is one Sensor Data Processing (SDP) module for each sensor i of fire detector, which is denoted by SDP_i. At each discrete time k, for sensor i, the SDP_i calculates the predicted trend t_i^k and the predicted level l_i^k of sensor i. The outputs of the SDPs are connected to the Fire Predictor (FP) module which predicts y^k and is designed as fully connected layers.

A. SENSOR DATA PROCESSING (SDP) MODULE

Before we predict the fire state via the FP module in Fig. 1, we first process the reading of each sensor i via the SDP_i module. The reason for using an SDP_i module is that we would like SDP_i to learn the relationship between the sensor reading x_i^k and each of the predicted trend t_i^k and the predicted level l_i^k. Each module SDP_i consists of the following units: Trend Predictor (TP_i) and Level Predictor (LP_i). At each discrete time k, this module takes x_i^k and x_i^{k-1} as the inputs and returns vector $[t_i^k, l_i^k, x_i^k]$ as the output.

1) Trend Predictor

TP_i unit of the SDP_i module calculates the trend of the reading of sensor i as

$$t_i^k = \alpha_i^1(x_i^k - x_i^{k-1}) + \alpha_i^2 t_i^{k-1} \tag{1}$$

Shortly, in (1), we take the weighted sum of the change in the sensor reading from $k-1$ to k and the previous value of the predicted trend. Although the idea of the trend prediction is based on the Holt-Winters double exponential smoothing [68], we now learn the prediction of the trend via a recurrent neural network without the condition $\alpha_i^1 = 1-\alpha_i^2$. That is, clearly, (1) states for equation of a linear recurrent neuron, which is shown in Fig. 2.

[1]We share the codes for the implementation of the rTPNN model at https://github.com/mertnakip/Recurrent-Trend-Predictive-Network.

[2]The value of γ might be selected empirically to achieve the best performance of rTPNN.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3087736, IEEE Access

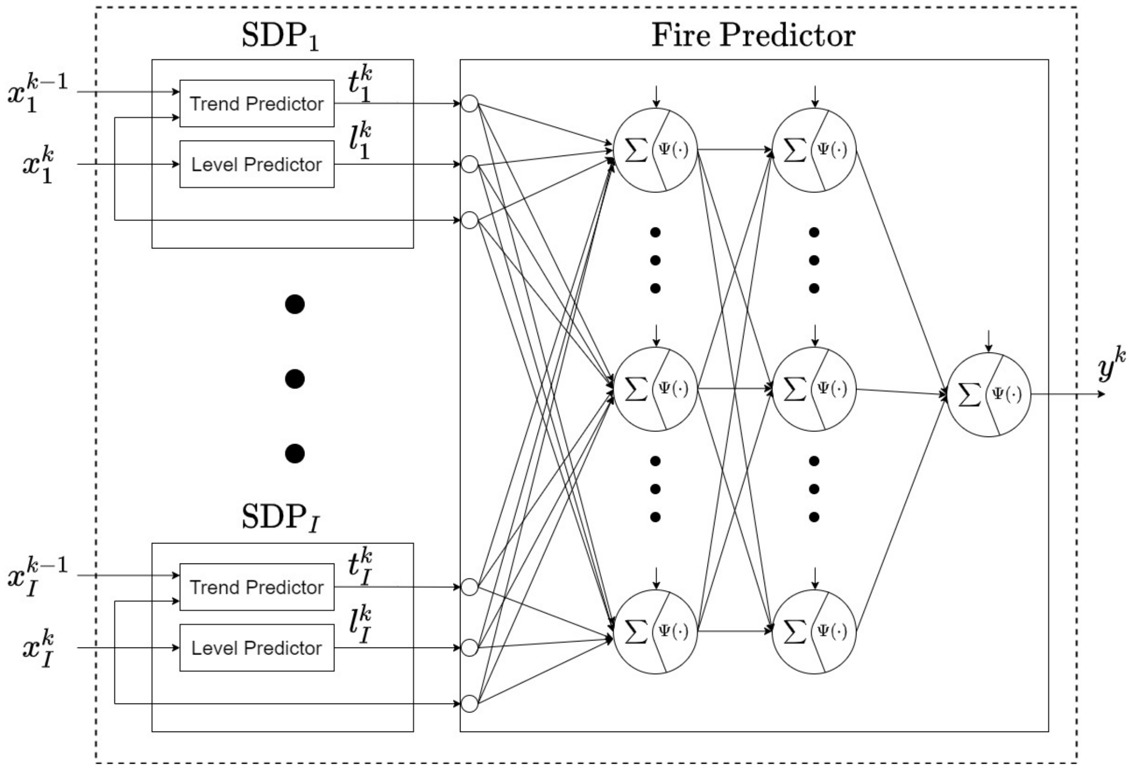

FIGURE 1. The architecture of the Recurrent Trend Predictive Neural Network (rTPNN)

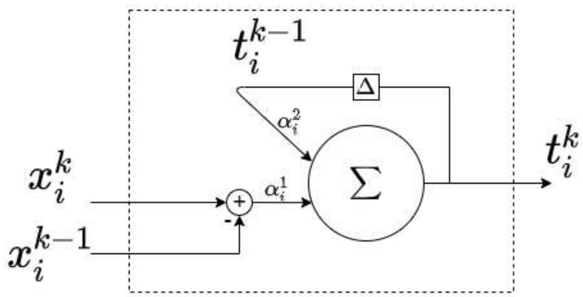

FIGURE 2. The inner architecture of the Trend Predictor unit for sensor i

In Fig. 2, we see that x_i^k and x_i^{k-1} are the inputs and the predicted trend t_i^k is the output of TP$_i$ unit. In this unit, $(x_i^k - x_i^{k-1})$ and the previous state of the predicted trend t_i^{k-1} is fed into the linear recurrent neuron with the connection weights α_i^1 and α_i^2, respectively. The values of these parameters are learned during the training stage of rTPNN.

2) Level Predictor

LP$_i$ unit of the SDP$_i$ module predicts the level of the reading of sensor i based on the well-known simple exponential smoothing (in other words, Holt Linear) [69], [70] as

$$l_i^k = \beta_i^1 x_i^k + \beta_i^2 l_i^{k-1} \tag{2}$$

Although (2) is the equation of Holt Linear, in our LP$_i$ unit, there is no direct relationship between β_i^1 and β_i^2, and it is represented as a linear recurrent neuron. The recurrent neuron of (2) is shown in Fig. 3.

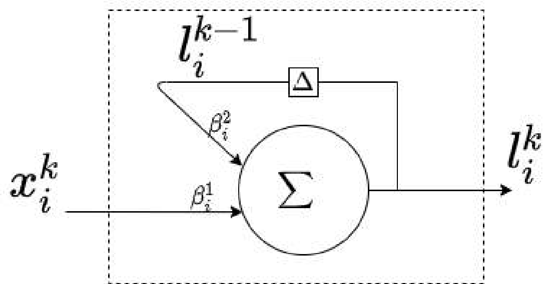

FIGURE 3. The inner architecture of the Level Predictor unit for sensor i

As shown in Fig. 3, at discrete time k, the only input of LP$_i$ unit is x_i^k and the output is l_i^k, where β_i^1 is the connection weight of x_i^k. In addition, the previous state of the predicted level l_i^{k-1} is fed into the linear neuron with connection weight β_i^2. Since LP$_i$ unit is a recurrent neuron, the parameters β_i^1 and β_i^2 are learned during the training of rTPNN.

B. FIRE PREDICTOR (FP) MODULE

In Fire Predictor (FP) module of rTPNN in Fig. 1, we use fully connected dense layer, whose inputs are $\{t_i^k, l_i^k, x_i^k\}_{i\in\{1,\dots,I\}}$, and output is y^k at time k. The FP module consists of H hidden layers and an output layer. Each hidden layer h is comprised of n_h fully connected neurons. In

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3087736, IEEE Access

addition, at the output layer of FP module, we use one neuron for the prediction of the fire state.

In Fig. 1, in FP module, each arrow represents a trainable parameter, namely connection weight or bias. We also let $\mathbf{W}_h$ denote the matrix of the connection weights for the inputs of hidden layer h, and $\mathbf{b}_h$ denote the vector of bias parameters of h. Thus, at each time k, the forward pass of the FP module is computed as

$$\mathbf{z}_1^k = \Psi(\mathbf{W}_1 \begin{bmatrix} t_1^k \\ l_1^k \\ x_1^k \\ \vdots \\ t_I^k \\ l_I^k \\ x_I^k \end{bmatrix}) + \mathbf{b}_1$$

$$\mathbf{z}_h^k = \Psi(\mathbf{W}_h \mathbf{z}_{h-1}^k) + \mathbf{b}_h \qquad \forall h \in \{2, \dots, H\}$$

$$y^k = \Psi(\mathbf{W}_{H+1}\, \mathbf{z}_H^k) + \mathbf{b}_{H+1}$$

where $\mathbf{z}_h^k$ is the output of hidden layer h. Furthermore, we set the activation function $\Psi(\cdot) = sigmoid(\cdot)$, where $sigmoid(a) = 1/(1+e^{-a})$. Note that $\Psi(.)$ is an element-wise operator for the vectors or matrices.

C. TRAINING OF rTPNN

We first let d^k denote the experimentally collected value of the fire state. At each discrete time $k \in [1, T]$, in order to collect x_i^k's and the corresponding d^k's, the fire experiments are performed. At k, if there is fire in the experiment, we set $d^k = 1$; otherwise, $d^k = 0$.

For the training of rTPNN model, we measure the prediction error of the model via the Mean Squared Error (MSE) as

$$E_{MSE} = \frac{1}{T} \sum_{k=1}^{T} (d^k - y^k)^2 \tag{3}$$

The parameters in $\{\mathrm{SDP}_i\}_{i \in \{1,\dots I\}}$ modules, $\{\gamma_i^1, \gamma_i^2\}_{i \in \{1,\dots I\}}$, $\{\beta_i^1, \beta_i^2\}_{i \in \{1,\dots I\}}$, and the connection weights and biases in FP module, $\{\mathbf{W}_h, \mathbf{b}_h\}_{h \in \{1,\dots,H+1\}}$, are learned in training stage of rTPNN to minimize the prediction error E_{MSE}.

IV. RESULTS

In this section, we evaluate the performance of the rTPNN model and compare that with the performance of each of Linear Regression (LR), Nonlinear Perceptron (NP), Multi-Layer Perceptron (MLP), Kendall-τ combined with MLP (Kendall-MLP), Probabilistic Bayesian Neural Network (PBNN), Long-Short Term Memory (LSTM) and Support Vector Machine (SVM) on fire experiments. To this end, we first review the data set that is used in this paper.

A. FIRE DATA SET

During the results of this paper, we evaluated the performance of the rTPNN model on the open-access data set, which is published in [8], [9].

This data set is comprised of 27 experiments each of which consists of the relative time to ignition in seconds as well as the sensor readings for an experiment. During these experiments, there are separately located sensors that measure the Temperature, Smoke Obscuration, and the concentrations of Carbon Monoxide, Carbon Dioxide, and Oxygen; however, the sensors that measure all of these metrics are only available in the (or close to the) bedroom in the testbed of data collection. Thus, we use only the experiments that are executed in the bedroom.

Moreover, since in this data set, three of the experiments (Experiment (Exp)-3, Exp-30, and Exp-32) are aborted due to the failure of ignition, we do not include those in our results. Thus, we use remaining 9 experiments.[3] Accordingly, we use 10-fold cross-validation (CV) over the rest of the experiments for the training and test of the fire prediction methods.

Furthermore, in order to simplify the input set of each of the fire detection methods, we normalize each sensory data x_i^k as $x_i^k \leftarrow \frac{x_i^k}{\max_k x_i^k}$. Note that while the original value of sensory data is greater than zero, the normalized data is in the range $[0, 1]$. In addition, for each experiment, we assume that the fire starts after 10 seconds (which is only two samples in the data set) from the ignition time. This assumption shows the expected time for the first triggered alarm but does not prevent fire detectors to detect fire earlier.

1) 10-Fold Cross-Validation (CV)

In order to evaluate the performance of models for the cases that are unseen by the models and the robustness of the models against the selection of the training set, we aim to perform 10-fold cross-validation (CV) on the data of 9 experiments that are mentioned above. To this end, we first create the input-output pair, denoted by $\{\mathbf{X}, \mathbf{Y}\}$, from the time series data of each experiment as follows.

Let assume that the number of samples in the time series data equals $T+1$. Moreover, x_i^k is the value for sensor i, and y^k is a binary label that states the fire situation at each $k \in \{0, T\}$ in this time series data. Then, $\mathbf{X}$ is a three-dimensional input tensor, whose size is $T \times 2 \times I$. The entry (k, p, i) of $\mathbf{X}$ equals $x_i^{(k+1)-p}$, where $k \in \{0, \dots, T-1\}$, $p \in \{0, 1\}$ and $i \in \{0, \dots, I-1\}$. $\mathbf{Y}$ is an output vector, whose length is T. The entry (k) of $\mathbf{Y}$ equals y^k. At the end of this operation, we have 25796 samples (i.e. $T = 25796$). Note that for 73.5% of these samples, the desired output is no fire; that is, the data set is imbalanced. In addition, while we observe Temperature, Smoke Obscuration, Carbon Monoxide, Carbon Dioxide and Oxygen, we have 5 sensors (i.e. $I = 5$). Accordingly, in Fig. 4, we show the input tensor and output (label) vector for rTPNN model.

Next, we shuffle these 25796 samples randomly and save them. Finally, we perform CV on the shuffled data set. Note

[3] We use the following experiments: SDC04-SDC07, and SDC36-SDC40. The number of samples in each of these experiments is respectively as follows: 1168, 222, 1372, 570, 5790, 4065, 3780, 2718, and 6120. Note that, for each experiment, the number of samples will decrease by one while we take $\{x_i^{k-1}\}_{i \in \{1,\dots,I\}}$ as an input.

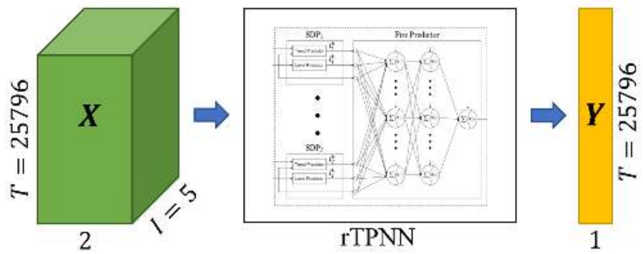

FIGURE 4. The dimensions of the input tensor and output vector of rTPNN in the case where all samples are fed into the model at once.

that in order to achieve a fair comparison between the models, we shuffle the data set once and use it for all models.

B. PARAMETER TUNING FOR STATE-OF-THE-ART MODELS AGAINST WHICH rTPNN IS COMPARED

We now explain the internal architectures, parameter tuning and the implementation details of the machine learning models that we have selected to compare with rTPNN model, which are LR, NP, MLP, Kendall-MLP, PBNN, LSTM and SVM.

In common, for all of these models, except SVM, we apply thresholding on the output of the model, y^k, after the training of the model is completed. To this end, we update y^k as

$$y^k \leftarrow \mathbb{1}_{y^k \geq \gamma} \tag{4}$$

In other words, we set $y^k = 1$, if $y^k \geq \gamma$ and $y^k = 0$, if $y^k < \gamma$, where γ denotes the value of threshold. Furthermore, for each of the nonlinear models (rTPNN, NP, MLP, PBNN, LSTM and Kendall-MLP), we exhaustively search for the best value of γ between 0 and 1 that maximizes the performance of the model on the training set.

1) Recurrent Trend Predictive Neural Network (rTPNN)

Recall that we need to determine only the number of hidden layers H and the number of neurons n_h at each hidden layer $h \in \{1, \dots, H\}$. To this end, we first set $H = 3$ because it is known that the neural network with at least three nonlinear layers is a universal approximator [71]. Then, we set the number of neurons $n_h = I.(H - h)$ at each hidden layer $h \in \{1, \dots, H-1\}$, and $n_h = \lceil I/2 \rceil$ at $h = H$.

2) Linear Regression (LR)

We select the LR as the simplest linear benchmark model. Via LR, we simply compute the weighted sum of the sensory inputs at current time t. The input of LR is the collection $\{x_i^k\}_{i \in \{1,\dots,I\}}$, and the output is y^k.

3) Nonlinear Perceptron (NP)

We select the NP as a nonlinear benchmark model. The architecture of the NP model is comprised of a single neuron with a nonlinear activation function, whose input is $\{x_i^k\}_{i \in \{1,\dots,I\}}$ and output is y^k. In addition, we set the activation function of NP to $sigmoid$ function.

4) Multi-Layer Perceptron (MLP)

The recent works [6], [7] have shown that MLP is a highly competitive model for multi-sensor fire detection. Thus, we select MLP as one of the neural network models against which the rTPNN will be compared. We use an MLP model which is comprised of H hidden layers and an output layer with a single neuron (namely, output neuron), where we set $H = 3$. We set the number of neurons $n_h = I.(H - h)$ at each hidden layer $h \in \{1, \dots, H-1\}$, and $n_h = \lceil I/2 \rceil$ at $h = H$. In addition, we set the activation function of each neuron at each layer to $sigmoid$ function.

5) Kendall-τ combined with MLP (Kendall-MLP)

The Kendall-MLP model is proposed by Reference [67] for multi-sensor fire detection. It is a fire detection method that uses the trend of the sensor reading. This method computes the trend via Kendall τ and predicts the fire via MLP based on the sensor readings and trend of those. In [67], the results on the subset of the data set that is also used in this paper have shown that Kendall-MLP achieves better accuracy than MLP and Radial Basis Function (RBF) neural network. In this paper, for the MLP block in the Kendall-MLP model, we use the architecture which is described above.

6) Probabilistic Bayesian Neural Networks

For multi-sensor fire detection, References [64] and [65] have used the probabilistic neural networks which predict a distribution of fire probability at each time. In this paper, we compare the performance of our rTPNN model against the Probabilistic Bayesian Neural Network (PBNN) [72]. The architecture of PBNN is comprised of a batch normalization layer, $H = 3$ hidden layers, a dense layer with two neurons and an output layer which returns a distribution. At each time, the fire probability is predicted as the mean of the distribution at the output of PBNN. We set the number of neurons $n_h = I.(H-h)$ at each hidden layer $h \in \{1, \dots, H-1\}$, and $n_h = \lceil I/2 \rceil$ at $h = H$. We also set the activation function of each neuron at each hidden layer to $sigmoid$. In addition, during the training of PBNN, the negative log likelihood is considered as the cost function.

7) Long-Short Term Memory (LSTM)

The recent work [73] has proposed the usage of an unsupervised model, which is the LSTM based Variational Autoencoder, for fire detection and has presented the performance for each individual experiments in the data set that is considered in this paper. Since the results of this paper are obtained with the supervised models for fire detection, the comparison of LSTM based Variational Autoencoder against

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3087736, IEEE Access

rTPNN is not fair.[4] (In this comparison, the supervised models (including rTPNN) would be advantageous.) On the other hand, considering the success of LSTM based Variational Autoencoder for fire detection in [73] and that of LSTM for other problems on multivariate time series data, we use LSTM for performance comparison.

For the design of the LSTM model, we connect an LSTM layer to the input of the MLP architecture which is explained above. The inputs of the LSTM model are $\{x_i^k\}_{i\in\{1,\dots,I\}}$ and $\{x_i^{k-1}\}_{i\in\{1,\dots,I\}}$, and the output of that is y^k. In addition, for the LSTM layer, we set the number of LSTM units to the number of sensors, I.

8) Support Vector Machine (SVM)

Reference [6] used SVM with RBF kernel as the decision maker for multi-sensor fire detection and showed that the percentage of false positive alarms is less than 10%. Thus, we selected SVM as the machine learning classifier to be compared (where, the rest of the methods are the regression models). We use RBF as the kernel of SVM. In addition, we selected the value of C parameter from the set $\{0.1, 1, 10\}$ as to maximize the performance of SVM on training set.

C. EVALUATION OF FIRE DETECTION PERFORMANCE

We now compare the performance of the rTPNN model with that of each of the LR, NP, MLP, Kendall-MLP, PBNN, LSTM and SVM as well as the benchmark single-sensor detectors. To this end, we use sensors that measure Temperature, Smoke Obscuration, and Carbon Monoxide, Oxygen and Carbon Dioxide concentrations[5].

1) Fire Detection Accuracy

First, we compare the fire detection methods with respect to the CV performance. In this way, we analyze the generalization ability of the detectors as well as the fire detection ability.

In Table 1, we present the fire prediction accuracy of each of the detection methods on the training and test sets. In addition, in order to clearly show the ranking of the methods, the rows of this table are ordered with respect to the accuracy of the methods and are grouped into the Machine Learning based Multi-Sensor Detectors and Single-Sensor Detectors with a Threshold.

In this table, we see that the rTPNN based multi-sensor detector achieves 96% accuracy and significantly outperforms all of the other fire detectors. In addition, when we estimate the worst-case performance of the rTPNN model as the subtraction of two standard deviation from the mean, the worst-case performance is 93% for the test set which is still better than the performance of the majority of other methods.

Furthermore, Table 1 shows that the multi-sensor fire detectors (especially combined with nonlinear models[6]) significantly outperform the single-sensor detectors combined with a threshold. Moreover, although the second best performing methods, LSTM, Kendall-MLP and MLP perform close to each other between 88% and 93%, the standard deviation of the LSTM is higher than those of the Kendall-MLP and MLP. That is, the worst-case performance of LSTM is almost equal to that of Kendall-MLP.

2) Metrics of Confusion Matrix

In order to analyze the fire detection performance of the methods in a more detailed way, in Table 2, we present the percentage means of True Positive Rate (TPR), False Negative Rate (FNR), True Negative Rate (TNR), and False Positive Rate (FPR) as well as the standard deviations of those. Note that TPR + FNR = 100 and the standard deviation of TPR and that of FNR are equal to each other. Similarly, TNR + FPR = 100 and standard deviation of TNR is equal to that of FPR. Before the examination of the results in this table, we shall note that 73.5% of the data set is comprised of "no fire" samples.

The results in Table 2 show that there is a significant gap between the percentage TPR of the rTPNN based multi-sensor detector and those of the other multi-sensor detectors, where the value of this gap varies between 9% and 55%. Only the Oxygen sensor based detector with a threshold achieves higher TPR than rTPNN. The reason is that this detector triggers a fire alarm for the majority of the samples. Accordingly, FPR of the Oxygen sensor based detector is unacceptably high so the usage of this detector is unpractical.

Moreover, FPRs of all of the detectors except Oxygen sensor based detector are below 6% which shows that those models trigger a wrong alarm only for 6% of the no fire cases.

Furthermore, Table 2 shows that only the rTPNN model is able to achieve high performance (above 92%) for both the TPR and TNR, while 73.5% of the data set is comprised of no fire samples. That is, rTPNN is the only model within the investigated models that predicts both fire and no fire cases with high success.

3) F1 Score, MCC, and ROC curve

Now, we present the F1 Score, Matthews Correlation Coefficient (MCC) and Receiver Operating Characteristic (ROC) Curve of each of the compared multi-sensor methods for the CV experiment.

First, Fig. 5 (top left) shows the average F1 Scores for the multi-sensor detectors for CV experiment. In short, the F1 score is the harmonic mean of the sensitivity (namely,

[4]The reason that the comparison between unsupervised LSTM based Variational Autoencoder (LSTM-VAE) and supervised methods is unfair is as follows: Since LSTM-VAE is an unsupervised model, it should be trained by using only the data related with no fire cases. In this way, LSTM-VAE does not require fire data (which is one of the most important properties of the unsupervised approach). However, such an unsupervised method detects fire later than the supervised models, where the recent work [73] has reported that time to alarm of LSTM-VAE is about 180 seconds on average for the experiments that are considered in this paper.

[5]The headers for these sensor data in the data set are TCB_1, SMB_1, GASB_1, GASB_3, GASB_5.

[6]In this paper, rTPNN, LSTM, Kendall-MLP, MLP, NP, PBNN and SVM (with RBF kernel) are the nonlinear models.

TABLE 1. COMPARISON OF FIRE DETECTORS WITH RESPECT TO THE CROSS-VALIDATION PERCENTAGE ACCURACY

Fire Detectors		Training		Test	
		Mean	Std	Mean	Std
Machine Learning based Multi-Sensor Detectors	rTPNN	96.38	1.79	96.25	1.85
	LSTM	93.00	2.95	93.02	2.99
	Kendall-MLP	89.64	0.96	89.79	1.25
	MLP	87.55	1.63	87.56	1.27
	Nonlinear Perceptron	85.86	0.10	85.92	0.88
	PBNN	84.54	1.32	84.65	1.07
	SVM	83.78	0.11	83.78	1.05
	Linear Regression	81.13	0.18	81.18	1.19
Single-Sensor Detectors with a Threshold	Temperature	78.51	0.13	78.51	1.15
	Carbon Monoxide	75.73	0.12	75.73	1.10
	Carbon Dioxide	74.17	0.14	74.17	1.28
	Smoke	73.55	0.13	73.55	1.19
	Oxygen	33.39	0.14	33.39	1.28

TABLE 2. COMPARISON OF FIRE DETECTORS WITH RESPECT TO EACH OF THE TRUE POSITIVE RATE, TRUE NEGATIVE RATE, FALSE POSITIVE RATE, FALSE NEGATIVE RATE

Fire Detectors		True Positive Rate	False Negative Rate	STD TPR / STD FNR	True Negative Rate	False Positive Rate	STD TNR / STD FPR
Machine Learning based Multi-Sensor Detectors	rTPNN	92.29	7.71	1.77	97.65	2.35	2.79
	LSTM	83.48	16.52	10.75	96.45	3.55	1.32
	Kendall-MLP	67.80	32.20	8.29	97.59	2.41	2.87
	MLP	63.40	36.60	3.13	96.20	3.80	1.64
	Nonlinear Perceptron	57.71	42.29	1.57	96.02	3.98	0.57
	PBNN	56.14	43.86	10.63	94.78	5.22	5.81
	SVM	42.66	57.34	2.30	98.50	1.50	0.34
	Linear Regression	37.19	62.81	1.51	96.92	3.08	0.68
Single-Sensor Detectors with a Threshold	Temperature	18.91	81.09	0.19	100	0	0
	Carbon Monoxide	13.61	86.39	0.15	98.12	1.88	0.04
	Carbon Dioxide	2.50	97.50	0.09	100	0	0
	Smoke	0.18	99.82	0.02	100	0	0
	Oxygen	97.00	3.00	0.10	10.46	89.54	0.09

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3087736, IEEE Access

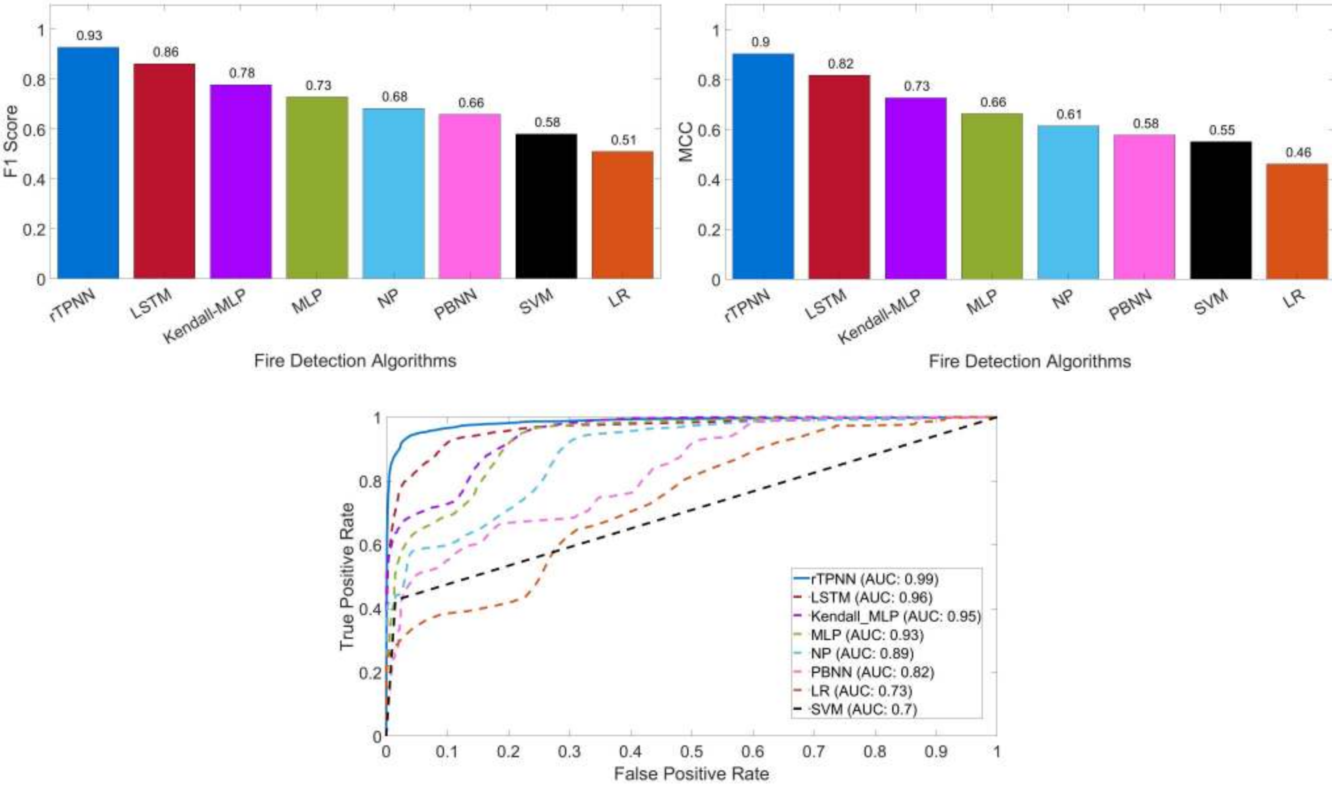

FIGURE 5. Comparison of the multi-sensor fire detectors with respect to the F1 Score (top left), MCC (top right) and ROC curve (bottom)

TPR percentage) and precision of the alarm decisions of a method; thus, it measures the balance between sensitivity and precision together. The sensitivity conveys how sensitive the method against the fire cases, and the precision conveys how accurate the alarm decisions of the method. Note that the F1 Score takes values in the range $[0, 1]$, where 1 indicates perfect sensitivity and precision, and 0 indicates vice-versa.

In this figure, we see that the F1 Score for rTPNN model is 0.93 which is very close to 1 and shows that the rTPNN model is highly successful in terms of both sensitivity and precision. In addition, the F1 Score of rTPNN model is significantly higher than those of the other models, where the difference is between 0.07 and 0.42.

Second, Fig. 5 (top right) displays the Matthews Correlation Coefficient (MCC) for each of the multi-sensor detectors. The MCC measures a binary classification performance by taking all of the true positives, true negatives, false positives, and false negatives into account. The MCC is a balanced measure for the imbalanced data sets and takes values in the range $[-1, 1]$, where 1 indicates perfect classification and -1 indicates vice-versa [74].

In this figure, we see that rTPNN significantly outperforms all of the other models with respect to MCC, where the difference is between 0.08 and 0.44. The results show that the performance gap between rTPNN and other models is widened about 2% from F1 Scores to MCC. The reason is that MCC provides a balanced measure between labels for our imbalanced data set while the label with more samples has a significant effect on the F1 Score. In addition, these results show that rTPNN is more robust against the imbalanced data set than the other models, so it outperforms the other models.

Finally, Fig. 5 (bottom) shows the ROC curve for each of the multi-sensor detectors. For each model, the ROC curve displays the trade-off between TPR (sensitivity) and FPR ($1-$specificity). Basically, as the performance of a model approaches perfect classification, its ROC curve converges to the upper left corner of the figure (where TPR = 1 and FPR = 0). In addition, note that an imaginary $45°$ line from the lower left to upper right corner states the performance of a random classifier. Furthermore, generally, the Area Under Curve (AUC) is also important to measure the performance of the model, where a higher AUC refers to better performance and the AUC takes values in $[0.5, 1]$.

Accordingly, in this figure, the ROC curve shows that rTPNN outperforms all of the other models and that the gap between the performance of rTPNN and perfect classifier is highly small. The ROC curve also shows that rTPNN treats both positive and negative labels almost equally while the 73.5% of the data set is comprised of no fire labels. Moreover, the legends of Fig. 5 (bottom) show that the AUC of rTPNN is 0.99, and the AUC gap between rTPNN and the second best-performing model LSTM is 0.03 which is a wide gap considering the best value of AUC is 1.

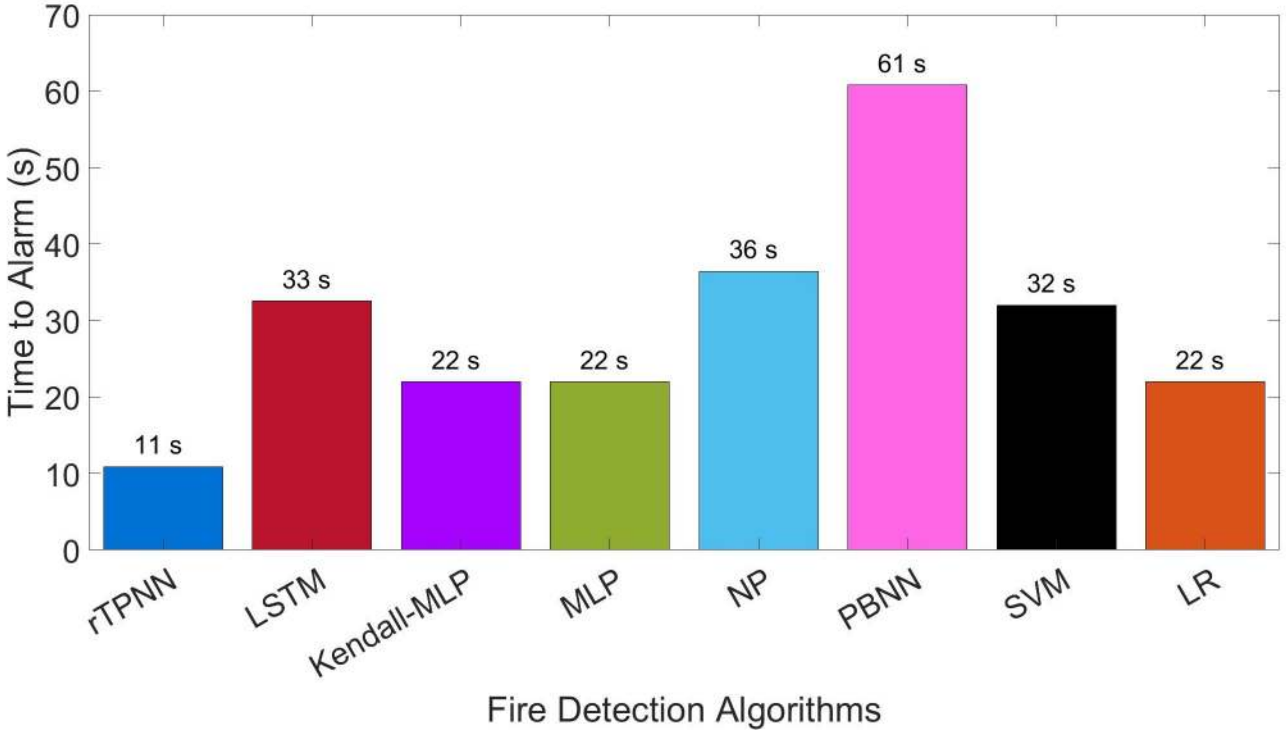

FIGURE 6. Comparison of the fire detectors with respect to the time passed between the start of the fire and the alarm

4) Time to Alarm

We now aim to compare the multi-sensor fire detectors (machine learning models) with respect to the time that is passed until the alarm, namely "time to alarm". To this end, we train and test each method on each fire experiment; then, we calculate the average time to alarm over the experiments. Note that in this experiment, we do not include the False Positive alarms for which the value of time to alarm is already negative.

In Fig. 6, we present the results for an average time to alarm of each model. The results in this figure show that the rTPNN model captures the fire significantly (at least 11 seconds) earlier than the other models due to its recurrent trend and level predictor structure. In addition, we see that each of the MLP, Kendall-MLP and LR triggers an alarm in 22 seconds while SVM, LSTM, NP and PBNN trigger in 32, 33, 36 and 61 seconds, respectively.

5) Analysis of the Learning Curve of rTPNN

In this subsection, we analyze the value of the MSE cost for each of the training and test sets in the 10th fold of CV during the training stage of rTPNN. These MSE costs are displayed for the shuffled data set that is used throughout the results of this paper in Fig. 7 (top) and for the data set that is not shuffled in Fig. 7 (bottom).

In Fig. 7 (top), we clearly see that training loss and test loss are similar to each other. The reason is that shuffling the data randomly distributes samples from different experiments; so that, the training set contains enough samples for both fire and no fire data from different cases to have better generalization ability. Supporting this reasoning, the results in Fig. 7 (bottom) show that the test loss increases as training loss decreases when the data set is not shuffled.

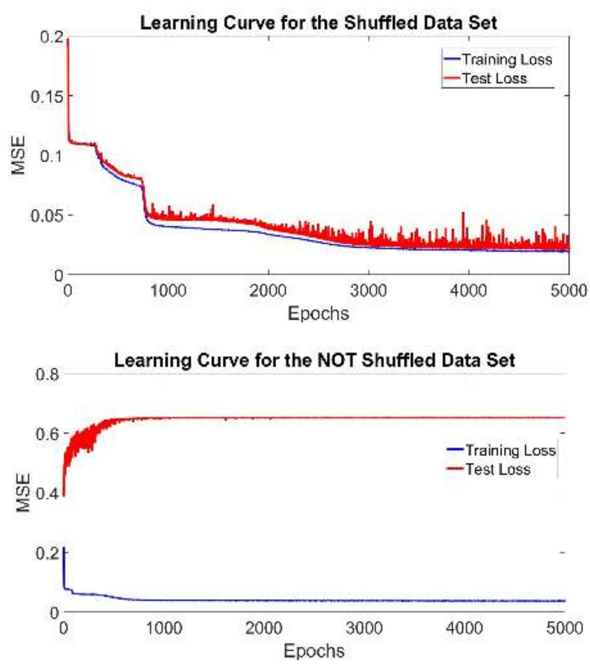

FIGURE 7. The MSE for training and test sets in 10th fold of CV during the training of rTPNN for 5000 epochs for shuffled data set (top) which is used during the results of this paper and for the data set which is not shuffled (bottom).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3087736, IEEE Access

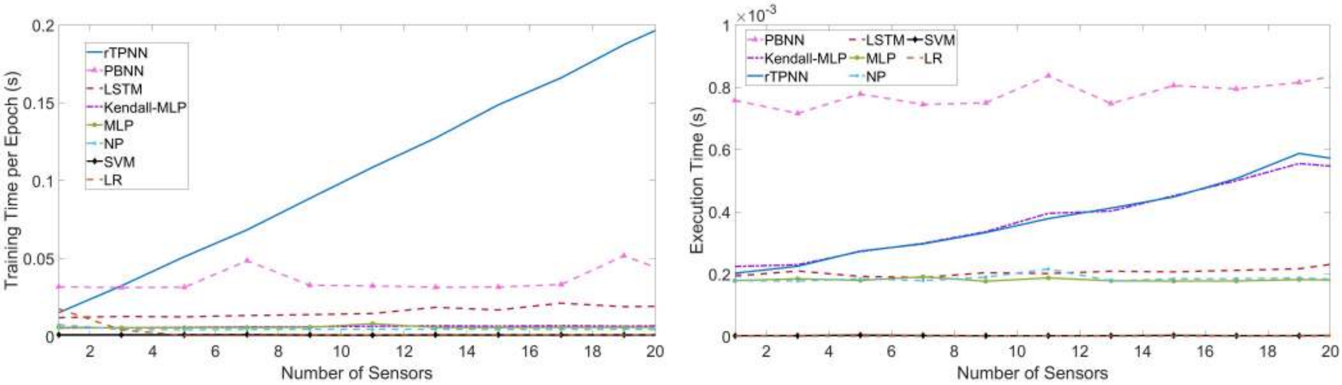

FIGURE 8. Comparison of the multi-sensor detectors with respect to the training time per epoch (left) and execution time (right) for increasing number of sensors

D. COMPUTATION TIME

We now present the comparison of the multi-sensor fire detectors with respect to each of the training and execution times. To this end, we measure the training time per epoch and the execution time per prediction on a single experiment[7] for the increasing number of sensors. For each number of sensors I, we select I random sensors from the set of all available sensors. In addition, we performed this experiment on the Google Colab platform with no accelerator.

In Fig. 8 (left), we present the training time per epoch for each fire detection method for the increasing number of sensors. We see that the training time of rTPNN increases linearly with the number of sensors. The reason is that in the internal architecture of rTPNN (in Fig. 1) for each sensor that is added to the fire detector, an SDP module is added to the neural network. In addition, it is seen that the training time rTPNN is higher than that of all of the other models for $I > 3$. On the other hand, for the fire detectors, the training of the detection method is performed offline, and the number of sensors is practical up to (around) 5 sensors due to the hardware costs; thus, the training time of rTPNN is highly acceptable for the practical implementation while its performance is superior to other methods.

Fig. 8 (right) shows the average execution time that is spent for a single prediction by a multi-sensor fire detector for increasing number of sensors. In this figure, we see that the execution time of rTPNN increases almost linearly with the number of sensors, where it is less than $0.6\ ms$ even for 20 sensors. In addition, the execution time of the slowest method is around $0.8\ ms$, which is acceptable for the real-time usage of the multi-sensor fire detectors.

V. CONCLUSION

We have proposed a novel Recurrent Trend Predictive Neural Network (rTPNN) which captures the trends of time series data via its recurrent internal architecture. rTPNN performs data fusion on the multivariate time series with its captured trend values and significantly improves the prediction performance.

We have evaluated the performance of rTPNN for the multi-sensor fire detection on the publicly available data set, and we presented the comparison of the performance of rTPNN with those of LR, NP, MLP, Kendall-MLP, PBNN, LSTM, and SVM. We have also demonstrated the performance evaluation via cross-validation in order to evaluate the generalization abilities of these models. According to our results, we have achieved the following conclusions: 1) The rTPNN model outperforms all of the machine learning models in terms of the prediction performance with high generalization ability. 2) rTPNN is the only model among the compared models which is able to achieve very low percentages for both False Negative Rate and False Positive Rate at the same time. 3) rTPNN triggers an alarm within only 11 s from the start of the fire, which is much earlier than the other models.

Moreover, we have shown that the execution time of the rTPNN model remains under $0.6\ ms$ up to 20 sensors connected to the fire detector. The execution time of rTPNN is comparable with that of the other models for up to 5 sensors while the performance of rTPNN is superior to that of the other models. Thus, the rTPNN model is a successful decision-maker for the real-time implementations of fire detectors due to its great prediction performance and highly acceptable execution time.

Even though we have presented rTPNN for the multi-sensor fire detection, rTPNN is a general neural network architecture that might be used for *any* prediction (as well as the classification and forecasting) problem on multivariate time series data. Based on the results of this paper, we interpret that the rTPNN model will have an impact on the various time series problems, especially where the trend of data plays a crucial role.

Future work will extend the application area of the rTPNN model beyond fire detection. In addition, we shall also examine the hardware implementation of the rTPNN-based multi-sensor fire detector and the performance evaluation of that via real-time experiments.

[7] For the computation time measurements, we evaluate the performance of the methods on the experiment SDC05; however, another experiment may be used instead of SDC05. The selection of the experiment will affect only the training time proportional to its number of samples.

REFERENCES

[1] 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 2014 IEEE Industry Application Society Annual Meeting. IEEE, 2014, pp. 1–6.

[2] E. O. M. Size, "Share & trends analysis report by application (cleaning & home, medical, food & beverages, spa & relaxation), by product, by sales channel, and segment forecasts, 2019-2025," Report ID, pp. 978–1, 2019.

[3] R. Chagger and D. Smith, "The causes of false fire alarms in buildings," Briefing Paper, 2014.

[4] F. Duisburg, "Statistics 2008," Duisburg, Germany, 2009.

[5] W. Kruell, T. Schultze, R. Tobera, and I. Willms, "Analysis of dust properties to solve the complex problem of non-fire sensitivity testing of optical smoke detectors," Procedia Engineering, vol. 62, pp. 859–867, 2013.

[6] M. Nakıp and C. Güzeliş, "Development of a multi-sensor fire detector based on machine learning models," in 2019 Innovations in Intelligent Systems and Applications Conference (ASYU). IEEE, pp. 1–6.

[7] ——, "Multi-sensor fire detector based on trend predictive neural network," in 2019 11th International Conference on Electrical and Electronics Engineering (ELECO). IEEE, 2019, pp. 600–604.

[8] R. D. Peacock, J. D. Averill, R. W. Bukowski, and P. A. Reneke, "NIST Report of Test FR 4016," Jan 2011. [Online]. Available: https://www.nist.gov/el/nist-report-test-fr-4016

[9] 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." 2003.

[10] V. Vipin, "Image processing based forest fire detection," International Journal of Emerging Technology and Advanced Engineering, vol. 2, no. 2, pp. 87–95, 2012.

[11] R. Ba, C. Chen, J. Yuan, W. Song, and S. Lo, "Smokenet: Satellite smoke scene detection using convolutional neural network with spatial and channel-wise attention," Remote Sensing, vol. 11, no. 14, p. 1702, 2019.

[12] Z. Xu, W. Wanguo, L. Xinrui, L. Bin, and T. Yuan, "Flame and smoke detection in substation based on wavelet analysis and convolution neural network," in Proceedings of the 2019 3rd International Conference on Innovation in Artificial Intelligence, 2019, pp. 248–252.

[13] S. H. Oh, S. W. Ghyme, S. K. Jung, and G.-W. Kim, "Early wildfire detection using convolutional neural network," in International Workshop on Frontiers of Computer Vision. Springer, 2020, pp. 18–30.

[14] Q. Zhang, J. Xu, L. Xu, and H. Guo, "Deep convolutional neural networks for forest fire detection," in 2016 International Forum on Management, Education and Information Technology Application. Atlantis Press, 2016.

[15] P. Barmpoutis, T. Stathaki, K. Dimitropoulos, and N. Grammalidis, "Early fire detection based on aerial 360-degree sensors, deep convolution neural networks and exploitation of fire dynamic textures," Remote Sensing, vol. 12, no. 19, p. 3177, 2020.

[16] P. Balasubramanian and K. Mari, "Multivariate regressive fully recurrent neural classifier for fire detection," IIOAB Journal, vol. 11, no. 5, pp. 14–22, 2020.

[17] H. Pan, D. Badawi, X. Zhang, and A. E. Cetin, "Additive neural network for forest fire detection," Signal, Image and Video Processing, pp. 1–8, 2019.

[18] A. Creţu, D. Radu, K. Nandor, C. Avram, C. Domuţa, and A. Aştilean, "Neural networks fire detection in a public safety fog system," in 2020 IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR). IEEE, 2020, pp. 1–6.

[19] P. Balasubramanian and K. Mari, "Empirical decomposed kernel hough feature transform based vector autoregressive bagging ensemble for forest fire detection," IIOAB Journal, vol. 11, no. 5, pp. 23–32, 2020.

[20] S. Masoumi, T. C. Baum, A. Ebrahimi, W. S. T. Rowe, and K. Ghorbani, "Reflection measurement of fire over microwave band: a promising active method for forest fire detection," IEEE Sensors Journal, pp. 1–1, 2020.

[21] M. Hefeeda and M. Bagheri, "Wireless sensor networks for early detection of forest fires," in 2007 IEEE International Conference on Mobile Adhoc and Sensor Systems, 2007, pp. 1–6.

[22] E. A. Kadir, S. L. Rosa, and R. A. Ramadhan, "Detection of forest fire used multi sensors system for peatland area in riau province," in AIP Conference Proceedings, vol. 2217, no. 1. AIP Publishing LLC, 2020, p. 020003.

[23] A. Lal and P. Prabu, "Fire detection and prevention in agriculture field using IoT."

[24] A. Jilbab, A. Bourouhou et al., "Efficient forest fire detection system based on data fusion applied in wireless sensor networks," vol. 12, no. 1, pp. 1–18, 2020.

[25] J. Zhang, Z. Ye, and K. Li, "Multi-sensor information fusion detection system for fire robot through back propagation neural network," Plos one, vol. 15, no. 7, p. e0236482, 2020.

[26] V. Dubey, P. Kumar, and N. Chauhan, "Forest fire detection system using IoT and artificial neural network," in International Conference on Innovative Computing and Communications. Springer, 2019, pp. 323–337.

[27] W. Benzekri, A. El Moussati, O. Moussaoui, and M. Berrajaa, "Early forest fire detection system using wireless sensor network and deep learning."

[28] B. C. Arrue, A. Ollero, and J. R. Matinez de Dios, "An intelligent system for false alarm reduction in infrared forest-fire detection," IEEE Intelligent Systems and their Applications, vol. 15, no. 3, pp. 64–73, 2000.

[29] A. Sharma, P. K. Singh, and Y. Kumar, "An integrated fire detection system using IoT and image processing technique for smart cities," Sustainable Cities and Society, vol. 61, p. 102332, 2020.

[30] E. Zervas, A. Mpimpoudis, C. Anagnostopoulos, O. Sekkas, and S. Hadjiefthymiades, "Multisensor data fusion for fire detection," Information Fusion, vol. 12, no. 3, pp. 150–159, 2011.

[31] S. Verstockt, A. Vanoosthuyse, S. Van Hoecke, P. Lambert, and R. Van de Walle, "Multi-sensor fire detection by fusing visual and non-visual flame features," in International Conference on Image and Signal Processing. Springer, 2010, pp. 333–341.

[32] J. Gotthans, T. Gotthans, and R. Marsalek, "Deep convolutional neural network for fire detection," in 2020 30th International Conference Radioelektronika (RADIOELEKTRONIKA). IEEE, 2020, pp. 1–6.

[33] K. Muhammad, J. J. Rodrigues, S. Kozlov, F. Piccialli, and V. H. C. de Albuquerque, "Energy-efficient monitoring of fire scenes for intelligent networks," IEEE Network, vol. 34, no. 3, pp. 108–115, 2020.

[34] Z. Wan, Y. Zhuo, H. Jiang, J. Tao, H. Qian, W. Xiang, and Y. Qian, "Fire detection from images based on single shot multibox detector," in International Conference on Computer Engineering and Networks. Springer, 2020, pp. 302–313.

[35] A. Jadon, M. Omama, A. Varshney, M. S. Ansari, and R. Sharma, "Firenet: A specialized lightweight fire & smoke detection model for real-time IoT applications," arXiv preprint arXiv:1905.11922, 2019.

[36] 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. 81 904–81 917, 2020.

[37] P. Li and W. Zhao, "Image fire detection algorithms based on convolutional neural networks," Case Studies in Thermal Engineering, p. 100625, 2020.

[38] Y. Valikhujaev, A. Abdusalomov, and Y. I. Cho, "Automatic fire and smoke detection method for surveillance systems based on dilated cnns," Atmosphere, vol. 11, no. 11, p. 1241, 2020.

[39] W. Wang and T. Li, "Fire video image detection based on a convolutional neural network," in Journal of Physics: Conference Series, vol. 1453, 2020, p. 012161.

[40] F. Gong, C. Li, W. Gong, X. Li, X. Yuan, Y. Ma, and T. Song, "A real-time fire detection method from video with multifeature fusion," Computational intelligence and neuroscience, vol. 2019, 2019.

[41] A. Khondaker, A. Khandaker, and J. Uddin, "Computer vision-based early fire detection using enhanced chromatic segmentation and optical flow analysis technique," The International Arab Journal of Information Technology, vol. 17, no. 6, pp. 947–953, 2020.

[42] 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, 2019.

[43] R. A. Sowah, K. Apeadu, F. Gatsi, K. O. Ampadu, and B. S. Mensah, "Hardware module design and software implementation of multisensor fire detection and notification system using fuzzy logic and convolutional neural networks (cnns)," Journal of Engineering, vol. 2020, 2020.

[44] A. E. Çetin, K. Dimitropoulos, B. Gouverneur, N. Grammalidis, O. Günay, Y. H. Habiboğlu, B. U. Töreyin, and S. Verstockt, "Video fire detection–review," Digital Signal Processing, vol. 23, no. 6, pp. 1827–1843, 2013.

[45] D. Gutmacher, U. Hoefer, and J. Wöllenstein, "Gas sensor technologies for fire detection," Sensors and Actuators B: Chemical, vol. 175, pp. 40–45, 2012.

[46] B. Charumporn, M. Yoshioka, T. Fujinaka, and S. Omatu, "Early stage fire detection using reliable metal oxide gas sensors and artificial neural networks," in Proceedings of the International Joint Conference on Neural Networks, 2003., vol. 4. IEEE, 2003, pp. 3185–3188.

[47] K. Lee, Y.-S. Shim, Y. G. Song, S. D. Han, Y.-S. Lee, and C.-Y. Kang, "Highly sensitive sensors based on metal-oxide nanocolumns for fire detection," Sensors, vol. 17, no. 2, p. 303, 2017.
[48] M. Yoshioka, T. Fujinaka, and S. Omatu, "Intelligent electronic nose systems with metal oxide gas sensors for fire detection," International Journal On Advances in Intelligent Systems Volume 2, Number 1, 2009, 2009.
[49] A. Solórzano, J. Fonollosa, L. Fernández, J. Eichmann, and S. Marco, "Fire detection using a gas sensor array with sensor fusion algorithms," in 2017 ISOCS/IEEE International Symposium on Olfaction and Electronic Nose (ISOEN). IEEE, 2017, pp. 1–3.
[50] F. Derbel, "Performance improvement of fire detectors by means of gas sensors and neural networks," Fire Safety Journal, vol. 39, no. 5, pp. 383–398, 2004.
[51] A. Uçar and R. Özalp, "Efficient android electronic nose design for recognition and perception of fruit odors using kernel extreme learning machines," Chemometrics and Intelligent Laboratory Systems, vol. 166, pp. 69–80, 2017.
[52] D. Sun, S. Zhang, G. Chen, W. Chang, M. Wang, and P. Guo, "Application of multi-sensor information fusion fire early warning system in intelligent fire protection field," in 2018 International Conference on Transportation & Logistics, Information & Communication, Smart City (TLICSC 2018). Atlantis Press, 2018.
[53] L. Salhi, T. Silverston, T. Yamazaki, and T. Miyoshi, "Early detection system for gas leakage and fire in smart home using machine learning," in 2019 IEEE International Conference on Consumer Electronics (ICCE). IEEE, 2019, pp. 1–6.
[54] H. Hu, G. Wang, Q. Zhang, J. Wang, J. Fang, and Y. Zhang, "Design wireless multi-sensor fire detection and alarm system based on arm," in 2009 9th International Conference on Electronic Measurement & Instruments. IEEE, 2009, pp. 3–285.
[55] J. A. Milke, M. E. Hulcher, C. L. Worrell, D. T. Gottuk, and F. W. Williams, "Investigation of multi-sensor algorithms for fire detection," Fire technology, vol. 39, no. 4, pp. 363–382, 2003.
[56] 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 Transactions on Industry Applications, vol. 53, no. 1, pp. 559–566, 2016.
[57] A. Kushnir and B. Kopchak, "Development of intelligent point multi-sensor fire detector with fuzzy correction block," in 2019 IEEE XVth International Conference on the Perspective Technologies and Methods in MEMS Design (MEMSTECH). IEEE, 2019, pp. 41–45.
[58] F. Rachman, N. Yanti, H. Hadiyanto, S. Suhaedi, Q. Hidayati, M. Widagda, and B. Saputra, "Design of the early fire detection based fuzzy logic using multisensor," in IOP Conference Series: Materials Science and Engineering, vol. 732, no. 1. IOP Publishing, 2020, p. 012039.
[59] P. Manjunatha, A. Verma, and A. Srividya, "Multi-sensor data fusion in cluster based wireless sensor networks using fuzzy logic method," in 2008 IEEE region 10 and the third international conference on industrial and information systems. IEEE, 2008, pp. 1–6.
[60] S. Chen, H. Bao, X. Zeng, and Y. Yang, "A fire detecting method based on multi-sensor data fusion," in SMC'03 Conference Proceedings. 2003 IEEE International Conference on Systems, Man and Cybernetics. Conference Theme-System Security and Assurance (Cat. No. 03CH37483), vol. 4. IEEE, 2003, pp. 3775–3780.
[61] W. Qu, J. Tang, and W. Niu, "Research on fire detection based on multi-source sensor data fusion," in International Conference on Mechatronics and Intelligent Robotics. Springer, 2019, pp. 629–635.
[62] D. Sharma, K. Singh, and S. Aggarwal, "Implementation of artificial neural fuzzy inference system in a real time fire detection mechanism," International Journal of Computer Applications, vol. 146, no. 10, pp. 31–37, 2016.
[63] T. Wang, Y. Xie, and H. Yan, "Research of multi sensor information fusion technology based on extension neural network," Mathematical Modelling of Engineering Problems, vol. 3, no. 3, pp. 129–34, 2016.
[64] S. L. Rose-Pehrsson, R. E. Shaffer, S. J. Hart, F. W. Williams, D. T. Gottuk, B. D. Strehlen, and S. A. Hill, "Multi-criteria fire detection systems using a probabilistic neural network," Sensors and Actuators B: Chemical, vol. 69, no. 3, pp. 325–335, 2000.
[65] S. L. Rose-Pehrsson, S. J. Hart, T. T. Street, F. W. Williams, M. H. Hammond, D. T. Gottuk, M. T. Wright, and J. T. Wong, "Early warning fire detection system using a probabilistic neural network," Fire technology, vol. 39, no. 2, pp. 147–171, 2003.
[66] M. I. Mobin, M. Abid-Ar-Rafi, M. N. Islam, and M. R. Hasan, "An intelligent fire detection and mitigation system safe from fire (sff)," International Journal of Computer Applications, vol. 133, no. 6, pp. 1–7, 2016.
[67] L. Wu, L. Chen, and X. Hao, "Multi-sensor data fusion algorithm for indoor fire early warning based on bp neural network," Information, vol. 12, no. 2, p. 59, 2021.
[68] P. R. Winters, "Forecasting sales by exponentially weighted moving averages," Management science, vol. 6, no. 3, pp. 324–342, 1960.
[69] C. C. Holt, "Forecasting trends and seasonal by exponentially weighted average," Office of Naval Research Memorandum, vol. 52, 1957.
[70] ——, "Forecasting seasonals and trends by exponentially weighted moving averages," International journal of forecasting, vol. 20, no. 1, pp. 5–10, 2004.
[71] F. Scarselli and A. C. Tsoi, "Universal approximation using feedforward neural networks: A survey of some existing methods, and some new results," Neural networks, vol. 11, no. 1, pp. 15–37, 1998.
[72] K. Salama, "Keras documentation: Probabilistic bayesian neural networks." [Online]. Available: https://keras.io/examples/keras_recipes/bayesian_neural_networks/
[73] 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. 30 636–30 653, 2021.
[74] S. Boughorbel, F. Jarray, and M. El-Anbari, "Optimal classifier for imbalanced data using matthews correlation coefficient metric," PloS one, vol. 12, no. 6, p. e0177678, 2017.

MERT NAKIP obtained his B.Sc. degree, with graduation rank #1, from the Electrical-Electronics Engineering at Yaşar University (Izmir, Turkey) in 2018. His design of a multi-sensor fire detector via machine learning methods was ranked #1 nationally at the Industry-Focused Undergraduate Graduation Projects Competition organized by TÜBİTAK (Turkish Scientific and Technological Research Council). He completed his M. Sc. thesis in Electrical-Electronics Engineering at Yaşar University (Izmir, Turkey) in 2020. His thesis focused on the application of machine learning methods to IoT and was supported by the National Graduate Scholarship Program of TÜBİTAK 2210C in High-Priority Technological Areas. He is currently a Research Assistant and a Ph.D. candidate at the Institute of Theoretical and Applied Informatics, Polish Academy of Sciences (Gliwice, Poland).

CÜNEYT GÜZELİŞ is Professor of Electrical Engineering and Director of the Graduate School at Yaşar University in Izmir, Turkey. His expertise is in nonlinear circuits and systems, machine learning, and systems biology. He has published over 50 SCI indexed journal papers with about 1000 SCI citations, 6 peer-reviewed book chapters, and more than 80 peer-reviewed conference papers. He has supervised 17 M.S. students and 14 Ph.D. students. He has served as Professor at Istanbul Technical University, as Dean of the Faculty of Engineering at Dokuz Eylül University, and as Director of the Graduate School of Natural and Applied Sciences at Izmir University of Economics. He has held visiting research and/or teaching positions at the University of California Berkeley, Aachen Applied Sciences, Paris-Est and Paris-Nord Universities. He has participated in over 20 scientific research projects funded by the national and international institutions, such as the British Council and the French National Council for Scientific Research.

OSMAN YILDIZ obtained his B.Sc. degree from the Department of Electronics and Communication at Istanbul Technical University (Istanbul, Turkey). He has been honored as a TÜBİTAK scholar during his undergraduate studies. He is the founder and the chief executive officer of EDS Electronic company. He also manages the R&D center at EDS. His expertise is in power electronics, fire and security systems, sensors, and IoT systems.