IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received October 3, 2021, accepted October 26, 2021, date of publication October 28, 2021, date of current version November 8, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3123970

A System for the Detection of Polyphonic Sound on a University Campus Based on CapsNet-RNN

LIYAN LUO[1,2], LIUJUN ZHANG[1,2], MEI WANG[1,3], ZHENGHONG LIU[1,2], XIN LIU[3], RUIBIN HE[1,2], AND YE JIN[1,2]

[1]Provincial Ministry of Education Key Laboratory of Cognitive Radio and Information Processing, Guilin University of Electronic Technology, Guilin 541004, China
[2]Guangxi Key Laboratory of Wireless Broadband Communication and Signal Processing, Guilin University of Electronic Technology, Guilin 541004, China
[3]College of Information Science and Engineering, Guilin University of Technology, Guilin 541004, China

Corresponding author: Mei Wang (zlj173218215@gmail.com)

This work was supported in part by the Subsidization of the Project of Guangxi Technology Base and Talent Special Project under Grant GuiKeAD20159018, in part by the Project of Guangxi Natural Science Foundation under Grant 2020GXNSFAA159004, in part by the National Natural Science Foundation of China under Grant 61771151, in part by the Guangxi Keypoint Research and Invention Program under Grant AB17292058, in part by the Innovation Project of GUET Graduate Education under Grant 2021YCXS024, and in part by the Director Fund for Provincial Ministry of Education Key Laboratory of Cognitive Radio and Signal Processing under Grant CRKL200104.

ABSTRACT In recent decades, surveillance and home security systems based on video analysis have been proposed for the automatic detection of abnormal situations. Nevertheless, in several real applications, it may be easier to detect a given event from audio information, and the use of audio surveillance systems can greatly improve the robustness and reliability of event detection. In this paper, a novel system for the detection of polyphonic urban noise is proposed for on-campus audio surveillance. The system aggregates different acoustic features to improve the classification accuracy of urban noise. A combination model composed of a capsule neural network (CapsNet) and recurrent neural network (RNN) is employed as the classifier. CapsNet overcomes some limitations of convolutional neural networks (CNNs), such as the loss of position information after max-pooling, and the RNN mainly models the temporal dependency of context information. The combination of these networks further improves the accuracy and robustness of polyphonic sound events detection. Moreover, a monitoring platform is designed to visualize noise maps and acoustic event information. The deployment architecture of the system is used in real environments, and experiments were also conducted on two public datasets. The results demonstrate that the proposed method is superior to existing state-of-art methods for the polyphonic sound event detection task.

INDEX TERMS Deep learning, polyphonic sound event detection, feature aggregation, monitoring platform.

I. INTRODUCTION

In real environments, visual information is generally not sufficient to reliably convey what occurs in a city, for example, a car horn in a no-honking area that is undetectable from video streams can be detected by audio analysis systems. Furthermore, abnormal events can occur at night and out of the camera's line of sight. In contrast, surveillance systems developed on the basis of audio analysis are not affected by changes in lighting and they do not have blind spots. By using only one mono microphone and one camera to integrate visual and audio data into the scene analysis, automatic surveillance systems' detection ability can be enhanced [1]. In recent years, urban environment sound detection has attracted increasingly more attention, and has been applied to various devices, such as audio surveillance devices [2], healthcare monitoring devices [3], [4], urban sound analytics devices [5], and smart home devices [6].

Although audio surveillance is critical for the detection of urban noise in real environments, there remain numerous problems of anomalous sound detection. One of the main issues is that sound events are usually generated by overlapping and mixing a lot of diverse sources, in many cases, multiple audio events may occur simultaneously, therefore it is difficult to generate pure training data. Moreover, it is very difficult to expand the number of sound categories. Furthermore, abnormal sounds will be superimposed on great levels of background noise, in some cases, occurring far away from the microphone, leading to very low signal-to-noise ratios (SNRs). Urban noise detection is an important component of

The associate editor coordinating the review of this manuscript and approving it for publication was Wei-Yen Hsu.

audio surveillance systems. Therefore, the focus of this article is to detect noise on a university campus, in terms of screams, car horns, glass breaking and firecrackers, on a university campus.

In recent decades, substantial efforts have been devoted to dealing with audio streams for applications including speech recognition [7], [8] and intelligent transportation [4], [9], [10]. The detection performance mainly depends on the use of features and classifiers, and the main features include perceptual linear predictive (PLP), linear predictive coding (LPC), log-mel spectrograms, and mel-frequency cepstral coefficients (MFCCs). In addition to the appropriate features, a satisfactory classifier plays an essential role in detecting sound events. Ordinary classifiers include non-negative matrix factorization (NMF) [11], support vector machines (SVMs) [12], Gaussian mixture models (GMMs) [13], multi-layer perceptrons (MLPs) [14], hidden Markov models (HMMs) [15], etc. The modeling ability of these traditional classification algorithms is poor for complex signals, and they can only achieve good performance on monophonic sound events. In a real environment, due to the mixing of multiple sound sources, sound events may occur simultaneously with partial or total overlap. Moreover, abundant environmental noise also exists for sound events which worsens the performance of the traditional classification models. Recent research shows that classifiers established based on deep learning algorithms are more effective as compared with traditional classifiers. Convolutional neural networks (CNNs) are one of the outstanding deep neural network structures, in [16] a CNN was used as a classifier, which has the ability to learn both time and frequency invariances by directly processing multi-dimensional information via global sharing. Even in a noisy environment with SNR $= 0$ dB, the recognition accuracy of monophonic sound event detection by the proposed method still reached 97.4% as compared with the HMM and SVM models, the accuracy of which was only 50%. However, CNNs are characterized by difficulties in the modeling of continuous audio streams because there is a max-pooling layer behind each convolution layer, which leads to the loss of position information after max-pooling. Recurrent neural networks (RNNs) [17] achieve improved performance via the integration of historical information. In previous research [18], by combining the advantages of both CNNs and RNNs, good polyphonic sound event detection performance was achieved. In this method, a CNN is used to extract low-level features and compress the frequency axis, while an RNN enhances partial and whole relationships by learning long-term context information. Their combination greatly improves the performance of lasting sound event detection. In another previous study [19], the capsule neural network (CapsNet) architecture was used to detect polyphonic sound events and achieved good performance on DCASE datasets as compared with the state-of-the-art algorithms. CapsNet can overcome some limitations of CNNs, such as the loss of position information after max-pooling.

In this article, a new method, namely the combination of Capsule and RNN, CapsNet-RNN, is proposed. The capsule network contains two parts, namely a convolution layer and a capsule layer. The capsule layer can overcome the loss of position information after max-pooling, and the convolution layer is used to learn local information. Finally, the RNN models the temporal dependency of context information. Further, to apply the proposed model to real environments, this article proposes a system that uses sensor nodes to capture urban noise. The audio signals are then introduced into the CapsNet-RNN network, which classifies the noise with a variety of labels, including ‘‘car horn,’’ ‘‘glass breaking,’’ ‘‘scream,’’ ‘‘gun_shot,’’ etc. Ultimately, noise maps and acoustic event information are visualized by the monitoring platform. The classification model contains two sections, consisting of feature extraction and classification. The aggregation of multiple frequency features and filter features is employed in feature extraction, and the proposed combination of MFCCs and log-mel spectrograms can achieve higher classification accuracy. The architecture of the classifier contains a convolution layer, a capsule layer [20], and a recurrent layer. The convolution layer extracts the original information of acoustic feature aggregation and inputs it into the capsule layer. The capsule layer learns partial information and makes appropriate predictions for the final classification. Finally, the recurrent layer adds context information for the entire classification process. Fig. 1 presents the framework of the proposed system.

The main contributions of this research can be summarized as follows:

(1) A new model for polyphonic sound event detection that combines CapsNet and an RNN is proposed.

(2) The aggregation of different acoustic characteristics is used as features, and extensive experiments were carried out on DCASE Challenge datasets.

(3) A novel system that includes 100 sensor nodes was deployed to monitor the noise and events on a real-life university campus.

(4) A monitoring platform is designed to visualize real-time noise maps and acoustic event information.

The rest of this article is organized as follows. Section II gives a detailed introduction to the proposed method and its rationale. In Section III, the hardware and software used in the proposed system are introduced, and the implementation steps of the proposed system are described. Section IV presents the experimental results and discusses the details of each experiment. Finally, this research is summarized in Section V.

II. METHODOLOGY

The system this research provided aims to detect audio events from background sounds, and then to use the classifiers to classify them into M categories. There are two main processes in the proposed algorithm, namely feature extraction and polyphonic detection. In the feature extraction process, the

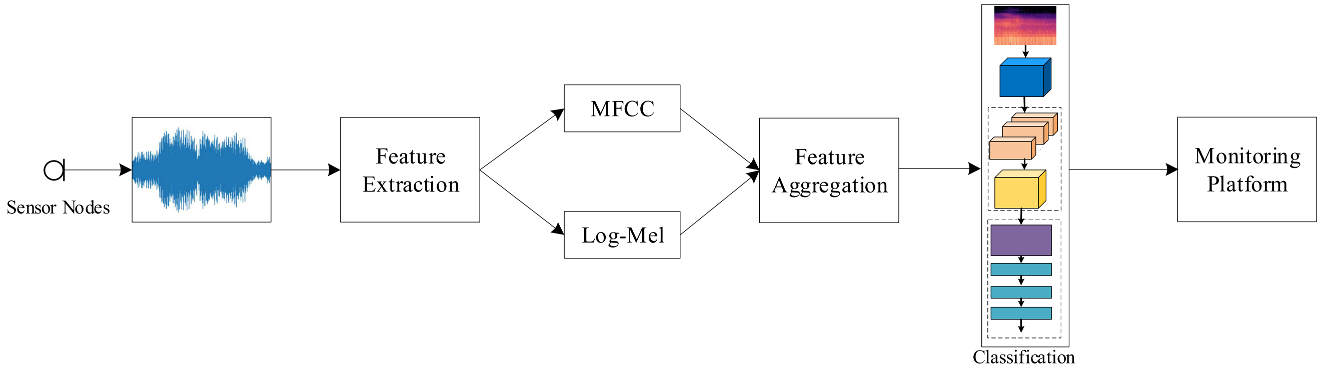


FIGURE 1. The framework of the proposed system.

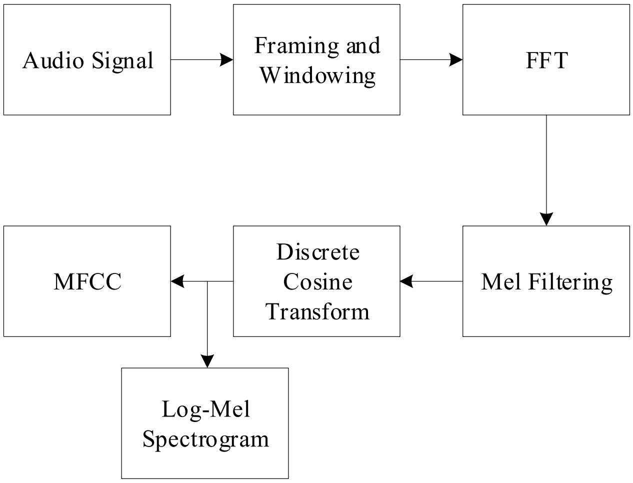


FIGURE 2. The process of feature extraction.

audio signal is converted into a time-frequency form with two dimensions, such as a feature vector $x_t \in \mathbb{R}^{F \times T}$, where F represents the frequency bands and T represents the total frame length in the feature map. In this article, MFCCs and log-mel spectrograms are integrated to compose a synthetic feature. The mel-frequency is proposed on the basis of the characteristics of human hearing, and it has a nonlinear relationship with Hz frequency. The combination of MFCCs and log-mel spectrograms further improves the detection accuracy. The following sections introduce the processes of the designed model for audio analysis, and the terms include: (1) the extraction of MFCCs and log-mel spectrograms, (2) feature aggregation, and (3) classification. Moreover, a detailed explanation of CapsNet-RNN is provided.

In actuality, the connection of human hearing with the frequency of the sound heard is nonlinear. To better research audio signals, mel-scale filter banks use the linearly-spaced frequency to imitate the human auditory system. The audio features extracted by such filter banks can better represent the sound characteristics. Fig. 2 shows the feature extraction processes of MFCCs and log-mel spectrograms, and the generation steps are described in it.

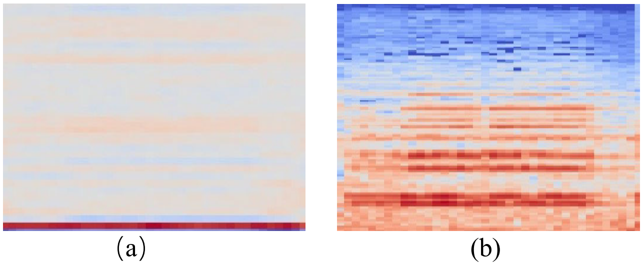


FIGURE 3. MFCC (a) and log-mel spectrogram (b).

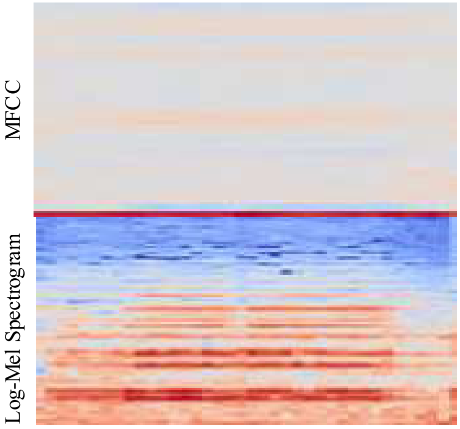


FIGURE 4. $\mathbf{F_{ML}}$ feature sets.

A. FEATURE AGGREGATION

Acoustic characteristics are important factors in classification tasks, and they affect the performance of environmental audio event recognition. Different acoustic features can describe sound signals from different angles. Many experiments have shown that the effective aggregation of features is able to improve the classification performance to a large extent. The log-mel spectrogram is obtained by mel-scale filter banks; it imitates the human auditory system and describes the global information of the spectrum of the audio signal and can visually reveal the differences between sounds. MFCCs are the result of the cosine transform of the real logarithm of the short-term energy spectrum expressed on a mel-frequency scale and MFCCs have been proved to proven to have superior efficient. An MFCC and log-mel spectrogram are respectively shown in Figs. 3 (a) and 3 (b).

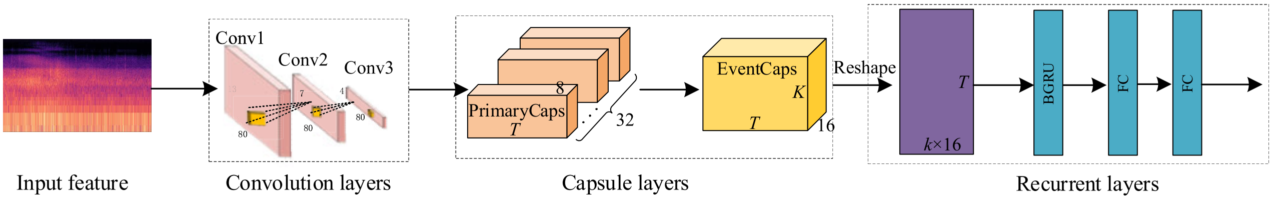


FIGURE 5. The framework of environmental sound classification.

Lower eigenvectors cannot adequately characterize audio events for neural network-based classification tasks. The aggregation of MFCCs and log-mel spectrograms can distinguish and supplement the sound signal from the global spectrum information, thereby further improving the detection robustness and reliability in several real applications. Let F_{ML} denote the linear aggregation of features, and via the same feature extraction method presented in a previous study [16], all audio streams are converted into 44100 Hz monophonic wave files and normalized to the range $[-1, 1]$ to ensure the same dynamic range. The STFT is calculated with a frame size equal to 40 ms and 50% overlap, after with a 40-band mel filter bank is performed to compute the mel band energies. The dimension of F_{ML} is $T \times 60$, where T is the number of frames in a sample, there are 60 frequency bins of input features, and the image of the combined features is displayed in Fig. 4.

B. CLASSIFICATION MODEL

This section provides a detailed explanation of the proposed model, which consists of the following three components. (1) The convolution layer is used to extract low-level features and compress the frequency axis via pooling; (2) the capsule layer is composed of PrimaryCaps and EventCaps. The essence of PrimaryCaps is a convolutional layer, which is mainly used to prepare for EventCaps, and the output of EventCaps is a vector whose size represents the probability of events; (3) the recurrent layer is employed to study temporal context data and reckon the likelihood of event activity. Fig. 5 shows the overview of the proposed model, and the hyperparameter settings of each layer are exhibited in Table 1.

1) CONVOLUTION LAYERS

The CNN can directly process multi-dimensional information and extract local features. In this study, to extract low-level features, three convolution layers are employed, and each layer is followed by a max-pooling layer. On the convolution layers, there are 256 filters of size 3×3 with two dimensions, and the setting of the stride is 1. In addition, a temporal size of T is maintained by the same-padding technique. In the following convolution, to improve the speed and stability of neural network training, the feature maps are normalized by batch normalization. The output is then adjusted with the rectified linear unit (ReLU) function [21] for nonlinear mapping, and the following presents the mapping relationship.

$$f(x) = \max(0, x). \tag{1}$$

The three max-pooling layers are only employed to reduce the dimension of the frequency axis, and the temporal dimension remains unchanged. Moreover, the sizes of the pooling kernels are set to 1×3, 1×2, and 1×2. By applying these settings appropriately, the size of the spectrum is decreased from the input $M = 60$ to $M = 20 \rightarrow 10 \rightarrow 5$ after the respective pooling layers. Regarding the input feature maps, $F_{ML} \in \mathbb{R}^{M \times T}$ is input into the convolution layers for local feature extraction, where M represents the spectral dimension of the input feature, while T refers to the temporal length. The output of the convolution layers is $F_{ML} \in \mathbb{R}^{N \times M' \times T}$, where N represents the number of two- dimensional filters while M' refers to the spectral dimension after the max-pooling layer.

2) CAPSULE LAYER

The capsule layer contains PrimaryCaps and EventCaps. The main idea of the capsule is to overcome the limitations of CNNs, such as the information loss after max-pooling. While the neurons of traditional neural networks are scalar, the capsule is a vector, and it can represent diverse attributes of a specific entity. The vector outputs (capsules) incorporate all the information detected in the input. In PrimaryCaps, there is a convolution capsule layer that has 32 channels, and each channel contains 8D capsule vectors. It is considered that through a group of weights, the lower-level capsules of PrimaryCaps can be linked to the higher-level capsules of EventCaps via the use of a dynamic routing mechanism. Moreover, the output of PrimaryCaps is $\mathbb{R}^{G \times T}$, where G represents 8D vectors with 32 channels.

The EventCaps layer is controlled by dynamic routing, the procedure of which is shown in Fig. 6. Recalling the original formulation in a previous study [22], the specific calculation process is as follows:

$$\hat{\boldsymbol{u}}_{j|i} = \boldsymbol{W}_{ij}\boldsymbol{u}_i, \tag{2}$$

$$\boldsymbol{s}_j = \sum\nolimits_i c_{ij}\hat{\boldsymbol{u}}_{j|i}, \tag{3}$$

$$\boldsymbol{v}_j = \frac{\| \boldsymbol{s}_j \|^2}{1 + \| \boldsymbol{s}_j \|^2} \frac{\boldsymbol{s}_j}{\| \boldsymbol{s}_j \|}, \tag{4}$$

where $\hat{\boldsymbol{u}}_{j|i}$ stands for the prediction vectors, $\boldsymbol{W}_{ij}$ stands for transformation matrices, $\boldsymbol{u}_i$ represents the output of low-level capsule i, and c_{ij} are the coupling coefficients between the capsule i and the capsule j, which are within the lower layer.

TABLE 1. The hyperparameters used in the proposed approach.

	Convolution layers			Capsule layers		Recurrent layers		
	Conv1	Conv2	Conv3	PrimaryCaps	EventCaps	GRU	FC	FC
kernel	256@3×3	256@3×3	256@3×3	32@3×3	-	-	-	-
stride	1×1	1×1	1×1	1×1	-	-	-	-
pooling size	1×3	1×2	1×2	-	-	-	-	-
activation function	ReLU	ReLU	ReLU	squashing	squashing	-	ReLU	sigmoid
num of hidden units	-	-	-	-	-	64	64	*K*
dim of capsule	-	-	-	8	16	-	-	-
the shape of output	256@*T*×20	256@*T*×10	256@*T*×5	32@*T*×5×8	*T*×*K*×16	*T*×64	*T*×64	*T*×*K*

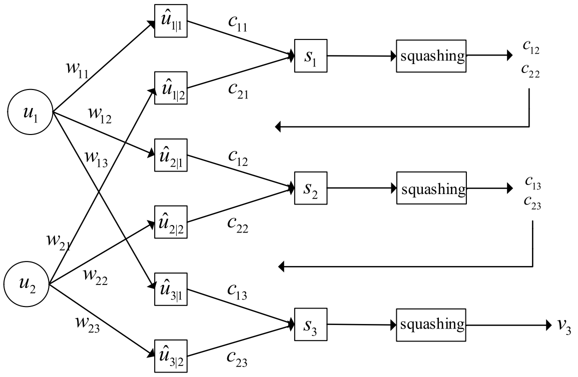


FIGURE 6. Dynamic routing.

All outputs $\hat{\boldsymbol{u}}_{j|i}$ are then accumulated to obtain $\boldsymbol{s}_j$, as given by (3). Equation (4) is the squashing function, which transforms the size of the vector to the probability within the interval (0,1).

The coupling coefficients are determined by applying the softmax function as follows:

$$c_{ij} = \frac{\exp(b_{ij})}{\sum_k \exp(b_{ik})}, \tag{5}$$

$$b_{ij} \leftarrow b_{ij} + \hat{u}_{j|i} \cdot v_j, \tag{6}$$

where b_{ij} is the initial log-prior probabilities, the value of which is set to 0 so that c_{ij} has the same initial value, and coupling coefficients b_{ij} are then recomputed on the basis of the similarity between the prediction vector $\hat{u}_{j|i}$ and the output of the high-level capsule v_j. The softmax function makes sure that the value of c_{ij} is between 0 and 1; accordingly, c_{ij} is the possibility that the lower layers send their outputs to the high-level layer. The output of EventCaps is $\mathbb{R}^{K\times G'\times T}$, where G' is a 16D vector and K stands for the number of event categories.

Margin loss function: The coupling coefficients c_{ij} are controlled by an independent margin loss, which is defined for each category k:

$$L_k = T_k \max(0, m^+ - \parallel v_k \parallel)^2 + \lambda(1 - T_k) \max(0, \parallel v_k \parallel - m^-)^2, \tag{7}$$

where T_k is the indicator function, and when the class K event exists, $T_k = 1$; otherwise, it is equal to 0. Moreover, m^+, m^-, and λ are super parameters that are respectively set to 0.9, 0.1, and 0.5. The total loss is calculated by summing the losses of all the output capsules. The Margin loss function is mainly optimized for c_{ij}. In the full connection layer of the RNN, the binary cross-entropy loss function is used for the final optimization.

3) RECURRENT LAYERS

Many works [23]–[25] have proven that temporal dependency is significant in detecting sound events. In this research, an RNN is used to add the unique position relationship of sound events by learning the temporal context information. After the capsule layer, the output $\mathbb{R}^{K\times G'\times T}$ is reshaped to $\mathbb{R}^{(K\times G')\times T}$ in each frame. The sequence is transformed to the sequence of recurrent feature vectors z_t:

$$z_t = [\mathbf{h}_t^{\mathrm{b}} \oplus \mathbf{h}_t^{\mathrm{f}}]\mathbf{W}_z + \mathbf{b}_z, \tag{8}$$

$$\mathrm{h}_t^{\mathrm{f}} = \mathcal{H}(\mathrm{x}_t, \mathbf{h}_{t-1}^{\mathrm{f}}), \tag{9}$$

$$\mathrm{h}_t^{\mathrm{b}} = \mathcal{H}(\mathrm{x}_t, \mathbf{h}_{t+1}^{\mathrm{b}}), \tag{10}$$

where $\mathbf{h}_t^{\mathrm{f}}$ and $\mathbf{h}_t^{\mathrm{b}}$ are respectively the forward and backward hidden state vectors of $\mathbb{R}^H$, H is the size of the recurrent unit, $\oplus$ indicates vector concatenation, $\mathbf{W}_z$ is a weight matrix, the size of which is set to $\mathbb{R}^{2H\times 2H}$, and $\mathbf{b}_z \in \mathbb{R}^{2H}$ is the bias term. Moreover, $\mathcal{H}$ stands for the hidden layer function of the recurrent layer. The output of the GRU layer is composed of the context information of the entire sequence, which is input into a time distributed dense layer and two completely linked layers, which have the sigmoid activation function, to obtain the likelihood of the events. $\mathbb{R}^{K\times T}$ is the final output, where K represents the probability of sound occurrence at time t.

III. DEPLOYMENT ARCHITECTURE

The proposed system is presented in Fig. 7, and is primarily composed of a remote server, solar panels, and wireless sensor nodes, each of which contains an acoustic sensor, a wireless transmission module, and an embedded processing board. Sound events and noise are collected via the sensor nodes by the endpoint detection algorithm; this algorithm is

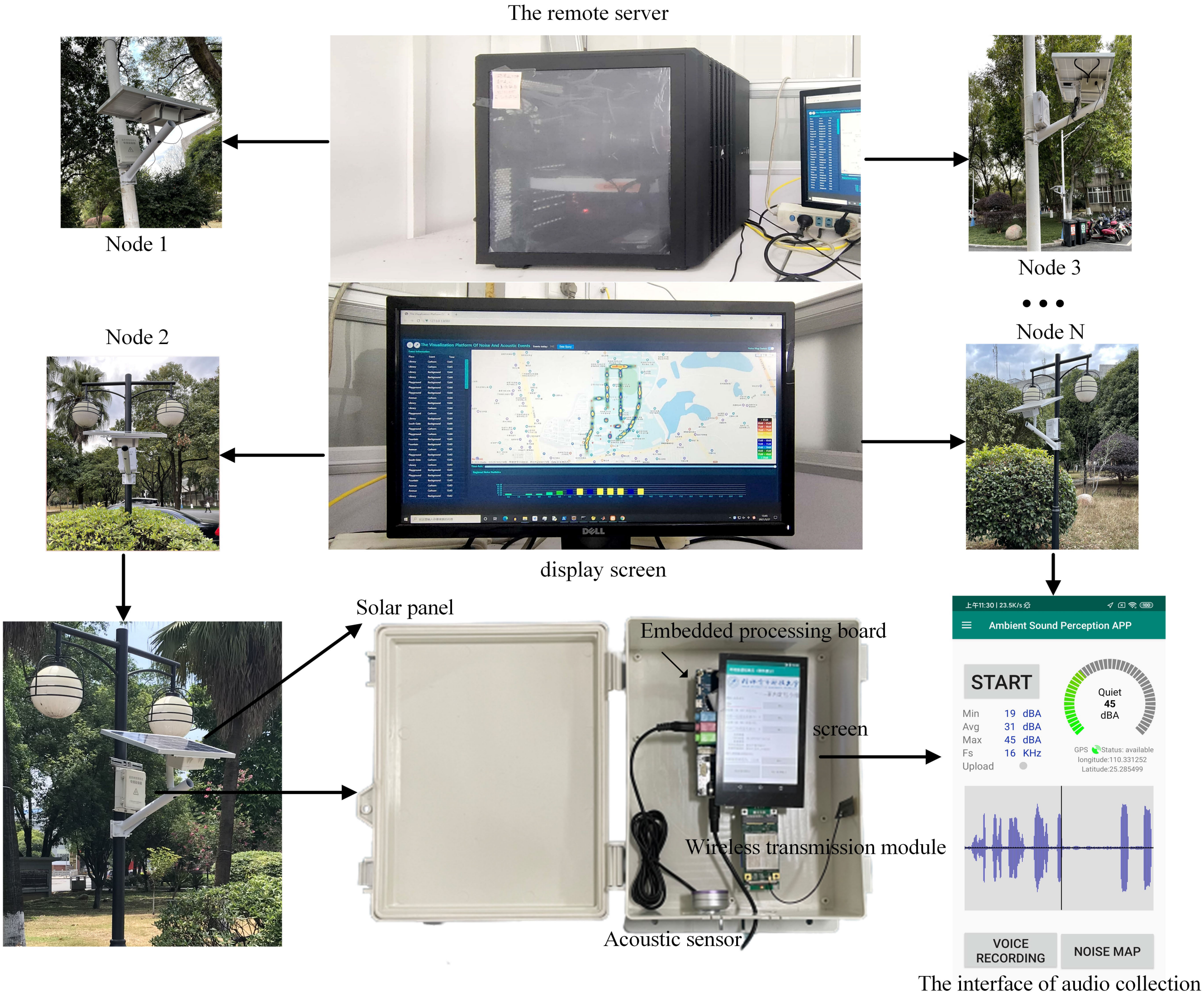
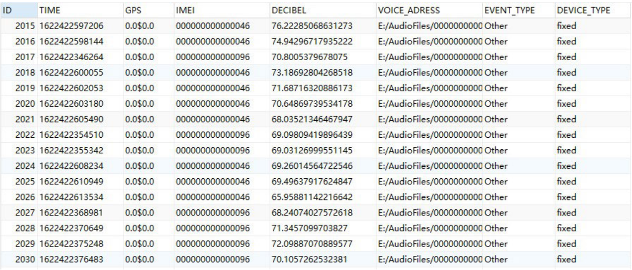


FIGURE 7. The overview of the proposed system.

written for the Android platform, and the detailed information of the software interface is shown in the figure. The collected audio is transmitted to and stored in the database of the remote server via the 4G network. Each piece of uploaded audio data contains its number, GPS information, time, decibel value, and storage address, as shown in Figure 8. Currently, internet attack activities are becoming increasingly more serious. To ensure the proper running of the server and the proposed model, the methods discussed by Ravi in [26] were introduced, namely DGA-Based Botnets and DNS Homographs Detection. The algorithm improves the resilience and robustness of the proposed system, thereby effectively avoiding the threat of internet attacks.

One hundred sensor nodes were deployed to monitor the noise and events on a university campus. According to the actual needs, these points were mainly arranged at the main roads, teaching buildings and gates, as shown in Fig. 9.



ID	TIME	GPS	IMEI	DECIBEL	VOICE_ADRESS	EVENT_TYPE	DEVICE_TYPE
2015	1622422597206	0.0$0.0	000000000000046	76.22285068631273	E:/AudioFiles/0000000000	Other	fixed
2016	1622422598144	0.0$0.0	000000000000046	74.94296717935222	E:/AudioFiles/0000000000	Other	fixed
2017	1622422346264	0.0$0.0	000000000000096	70.8005379678075	E:/AudioFiles/0000000000	Other	fixed
2018	1622422600055	0.0$0.0	000000000000046	73.18692804268518	E:/AudioFiles/0000000000	Other	fixed
2019	1622422602053	0.0$0.0	000000000000046	71.68716320886173	E:/AudioFiles/0000000000	Other	fixed
2020	1622422603180	0.0$0.0	000000000000046	70.64869739534178	E:/AudioFiles/0000000000	Other	fixed
2021	1622422605490	0.0$0.0	000000000000046	68.03521346467947	E:/AudioFiles/0000000000	Other	fixed
2022	1622422354510	0.0$0.0	000000000000096	69.09809419896439	E:/AudioFiles/0000000000	Other	fixed
2023	1622422355342	0.0$0.0	000000000000096	69.03126999551145	E:/AudioFiles/0000000000	Other	fixed
2024	1622422608234	0.0$0.0	000000000000046	69.26014564722546	E:/AudioFiles/0000000000	Other	fixed
2025	1622422610949	0.0$0.0	000000000000046	69.49637917624847	E:/AudioFiles/0000000000	Other	fixed
2026	1622422613534	0.0$0.0	000000000000046	65.95881142216642	E:/AudioFiles/0000000000	Other	fixed
2027	1622422368981	0.0$0.0	000000000000096	68.24074027572618	E:/AudioFiles/0000000000	Other	fixed
2028	1622422370649	0.0$0.0	000000000000096	71.3457099703827	E:/AudioFiles/0000000000	Other	fixed
2029	1622422375248	0.0$0.0	000000000000096	72.09887070889577	E:/AudioFiles/0000000000	Other	fixed
2030	1622422376483	0.0$0.0	000000000000096	70.1057262532381	E:/AudioFiles/0000000000	Other	fixed

FIGURE 8. Information about the sound collected by the proposed system.

In the proposed system, an audio event with SNR $= 20$ dB can be detected. A set of 100 sensor nodes, $R = \{r_i | i = 1, \ldots, 100\}$, was deployed around the campus. The distance between these sensor nodes was m meters, and the height from the ground was h meters.

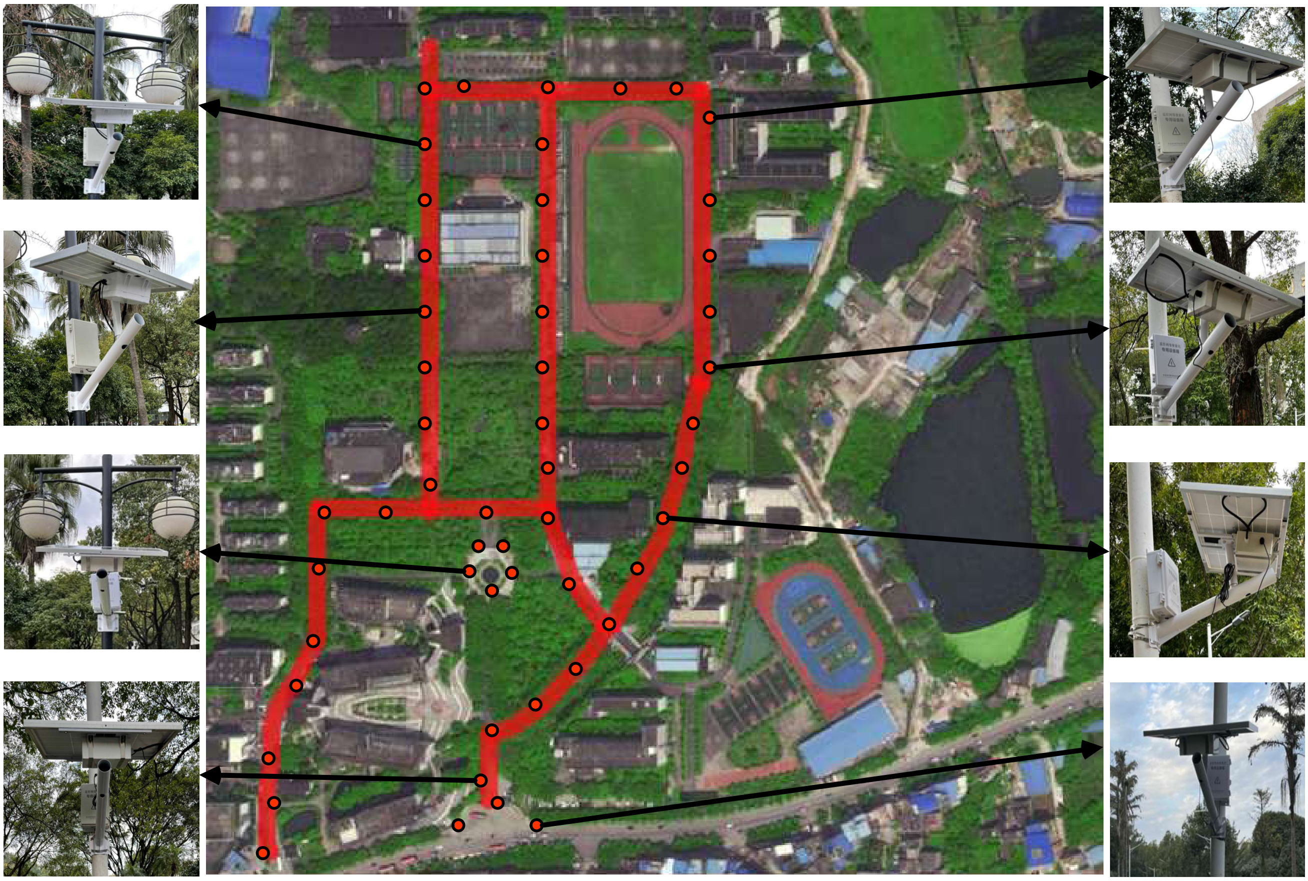


FIGURE 9. The deployment of sensor nodes around the university campus.

Two factors affected the choice of the distance m and the height h, including the distance d from the sensor node where the sound occurs, as well as the sound level of the target events that are intended to be spotted. The coverage capabilities of the sensor nodes were investigated by computing the SNR of the sound, which is defined as

$$\mathrm{SNR} = L_S(d_0) - A(d), \tag{11}$$

where $L_S(d)$ is the intensity level of the sound event at a reference distance d_0 from the sensor nodes. The factor $A(d)$ is the attenuation according to the ISO standard 9613-2 [27], consisting of atmospheric absorption, geometrical difference, ground effects and shielding through barriers. The following equation presents the calculation process:

$$A(d) = A_{\mathrm{div}}(d) + A_{\mathrm{atm}}(d) + A_{\mathrm{gr}}(d) + A_{\mathrm{bar}}(d). \tag{12}$$

These factors are determined by the specific environment. In particular,

(1) $A_{\mathrm{div}}(d)$ is the geometrical divergence. In actual scenarios, a sound source spreads evenly in all directions; this can be represented by a sphere, which is calculated as

$$A_{\mathrm{div}}(d) = 20 \log \frac{d}{d_0} + 11. \tag{13}$$

In certain circumstances, every time the distance from the source is doubled the intensity level will be reduced by 6 dB;

(2) $A_{\mathrm{atm}}(d)$ is the attenuation caused by the atmospheric absorption (in decibels), and can be calculated as follows:

$$A_{\mathrm{atm}} = \frac{\alpha \cdot d}{1000}, \tag{14}$$

where α is the atmospheric attenuation coefficient, which is determined by the frequency, the environmental temperature, and the comparative humidity of the air. For the calculation of environmental noise levels, α should be established based on average values determined by the surrounding weather range;

(3) $A_{\mathrm{gr}}(d)$ represents the ground effect, and mainly refers to the interference or influence of the sound reflected by the ground on the sound transmitted directly from the sound source. To compute $A_{\mathrm{gr}}(d)$, h_r and h_s are respectively considered the receiver height and the source height of the sensor nodes. According to a relevant standard [26], the sound source is divided into three parts: 1) the source region (with a size of $30 \cdot h_s$), the attenuation of which is A_s, 2) the middle region, which determines the attenuation A_r, and 3) the receiver region (with a size of $30 \cdot h_r$) around the receiver, which determines the attenuation A_{m}. The total

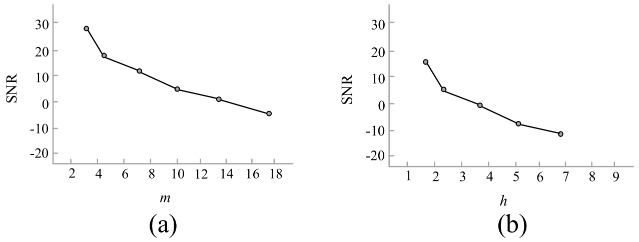


FIGURE 10. The SNR value with respect to (a) the distance *m* and (b) height *h*.

ground attenuation can then be computed as

$$A_{\mathrm{gr}} = A_{\mathrm{s}} + A_{\mathrm{r}} + A_{\mathrm{m}}. \tag{15}$$

At the frequency of 4 kHz, A_s and A_r can be calculated by the following equation:

$$A_r = A_s = 1.5 \cdot (1 - G). \tag{16}$$

For hard ground, the value of G is equal to 0. Therefore, the values of A_{s} and A_{r} are equal to 1.5. A_{m} can then be calculated by the following equations:

$$A_m(d) = 3 \cdot q(d) \cdot (1 - G), \tag{17}$$

where

$$q(d) = \left\{ \begin{array}{cc} 0 & d \leq 30(h_s + h_r) \\ 1 - \dfrac{30(h_s + h_r)}{d} & d > 30(h_s + h_r) \end{array} \right\}. \tag{18}$$

(4) $A_{\mathrm{bar}}(d)$ represents screening by barriers. In the scenarios considered in this study, this attenuation was neglected because the experiments were conducted on the roads of the campus.

To determine the appropriate distance m and height h, a series of experiments were carried out. Fig. 10 depicts the attenuation of the SNR with the distance m and the height h, from which it can be concluded that when the SNR is about 10 dB, the distance is 7 m and the height is 2 m. Therefore, for the deployment of the sensor nodes, $m = 7$ m and $h = 2$ m were selected.

To better monitor noise and sound events on the campus, Java was used to write a noise monitoring display platform, which directly displays the classification results obtained by the proposed algorithm. The time, place, class, and decibel size of the sound event are visualized by the noise platform. When an event with SNR $\geq$ 20 dB occurs, the sensor nodes are able to detect the event, and the landmarks on the noise graph will pulse briefly. The specific visualization of the monitoring platform is presented in Section IV.

IV. EXPERIMENTAL RESULTS

This part introduces the experimental data and the experimental setup, including the evaluation metrics used in the field of polyphonic sound event recognition [28], [29]. To verify the reliability of the proposed method, experiments were carried out in two scenarios. (1) An experiment was carried out on publicly available datasets, and the experimental outcomes of the proposed approach provided were then compared with the results of existing state-of-the-art approaches. (2) An analysis was conducted on actual life scenarios; this research's method and constructed system were applied to the real-life campus environment, and real-time data were analyzed. In the last, an analysis was conducted on the classification outcomes of the system, and the information of the noise and events were mapped to the monitoring platform.

A. THE DATA-SET

To evaluate on the performance of the proposed approach, it was assessed on two public datasets, namely the TUT Sound Events 2016 and TUT Sound Events 2017 datasets, which were the datasets adopted in the DCASE challenge [30], [31]. The outcomes were compared with the outcomes of the greatest state-of-the-art methods offered by the challenge organizers. Each network model was optimized by the use of a random search tactic [32]. The datasets are described as follows.

The TUT Sound Events 2016 dataset contains two daily environments, namely "home" (indoors) and "residential area" (outdoors). These two acoustic scenes are considered as two separate subsets and used for human activity monitoring and household monitoring in daily life. The audio samples from 16 independent sound event classes were randomly selected and artificially mixed. Each mixed audio sample is about 4-7 min long, and the total length of the audio samples in the dataset is about 10 hours. All the audio is mono with a resolution of 24 bits and a sampling rate of 44.1 kHz. In the scenario of the residential area, there are seven types of sound events, and the sound event categories are mainly associated with birds singing and cars passing by. In the scenario of the home, there are 11 categories.

Each scenario in the TUT Sound Events 2016 dataset was split into two sub-datasets, including a development set and an evaluation set. In this research, the dataset was divided based on the number of total samples available; the development set consisted of approximately 70% of the total samples, while the evaluation set consisted of 30% of the total samples. In addition, data division was also conducted on the development set, the output was four folds of training and testing data for cross-validation during the training process, and each recoding was employed just once as testing data. The process is shown in Fig. 11.

The TUT Sound Events 2017 dataset consists of recordings of acoustic scenes of traffic and human activities on a street and is used to detect audio events associated with human and abnormal situations. The dataset contains a total of 121 min of audio data and includes six different sound events, namely "people speaking," "people walking," "children," "large vehicle," "brakes squeaking," and "car." There are 24 recordings in the dataset, each of which is about 3-5 min long, and the recordings were sampled at 44.1 kHz and a resolution of 24 bits. The data division for cross-validation was the same as that for the TUT Sound Events 2016 dataset.

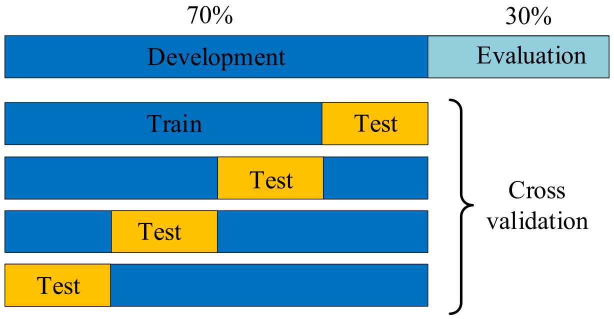


FIGURE 11. Database partitioning into training and evaluation sets.

B. EXPERIMENTAL SETUP

In this research, all the experiments were carried out by the use of a computer equipped with Intel® Core^{TM} I7-6800K processor, the Windows 10 platform, an NVIDIA 1080TI GPU, and 32 GB RAM. The feature extraction and classification algorithms were completed via learning libraries in the Python language, including Librosa, Sklearn, and TensorFlow. The detection accuracy of the proposed model is directly affected by the setting of the hyperparameters. Thus, to determine the most appropriate hyperparameters, a random search tactic was first used to determine the best number of layers of the network model. According to the methods used in previous research [33], [34], a series of experiments were carried out to obtain the best hyperparameters. Because the CapsNet-RNN model includes convolution layers, capsule layers, and recurrent layers, the performance of the convolution layer is closely related to the size and number of filters. In the experiments, the number of filters was respectively set to 128, 256, and 512, and the filter size was respectively set to 3 × 3 and 5 × 5. The same parameters used in a previous study [35] were set for the capsule layer. For the recurrent layers, the numbers of hidden units were respectively set to 64, 128, and 256, and the learning rate was within the range of 0.001 to 0.5. Regarding the activation functions, the sigmoid, tanh, and ReLU activation functions were selected for the experiments. All experiments were optimized by the ADAM optimizer, and the categorical cross-entropy loss function was used. The experiments were run on the GoogleAudioSet dataset [36], [37] for 500 epochs with a batch size of 64. This dataset is often used to evaluate environmental sound recognition methods. Four common environmental sounds in this dataset were selected, namely: "car horn," "glass breaking," "screams," and "gun_shots," each of which has about 1,000 pieces of data. All the audio is mono with a resolution of 24 bits and a sampling rate of 44.1 kHz. For the entire data set, the training set and test set were randomly divided according to the ratio of 8:2. Based on the experimental results, the performance of the model was good when the number of filters was 256 and the filter size was 3 × 3. Further, for the recurrent layers, the use of fewer hidden units was found to have a good effect on the attack detection rate. The model with 64 hidden units performed better than the others; when the number of hidden units was increased from 64 to 256, the attack detection rate decreased. Moreover, many studies [38], [39] have shown that a lower learning-rate has a better effect. In this experiment, when the learning-rate was 0.001, the detection accuracy was better than that of the model with other learning-rates; with the increase of the learning-rate, the performance of the model fluctuated due to overfitting. Regarding the activation function, the performance of ReLU was found to be better than that of the tanh. However, because the main work in the present study is multiclass classification, the sigmoid activation function was used for the fully connected layer. The final hyperparameter settings of each layer are exhibited in Table 1. To avoid the issue of overfitting, an early stopping strategy [40] was employed during training. Batch normalization was then used to accelerate the convergence and reduce the network sensitivity of the network to the initialization weights, and after each convolutional layer, a loss rate of 0.25 was employed.

The generally recognized measurement indicators proposed by Mesaros and Heittola were used in this study, and include the F1-score (F1) and error rate (ER), both of which are based on segments. On the basis of the activity representation, the intermediate statistics were computed as below:

True positive (TP): it refers to the number of events detected not only in the system output, but also in the annotation;

False positive (FP): it refers to the number of events detected only in the system output;

False negative (FN): it refers to the number of events detected only in the annotation;

Substitutions ($S(t)$): it represents the number of events A that the system misjudged A as B;

Insertions ($I(t)$): it refers to the number of events detected only in the system output and also did not belong to $S(t)$;

Deletions ($D(t)$): it refers to the number of events originally in the annotation that the system did not misjudge and that were not correctly detected by the system.

Moreover, F1 is the harmonic average value of precision (P) and recall (R). The following formula group presents the specific calculation process.

$$P = \frac{\sum TP}{\sum TP + \sum FP}, \tag{19}$$

$$R = \frac{\sum TP}{\sum TP + \sum FN}, \tag{20}$$

$$F1 = \frac{2P \cdot R}{P + R}. \tag{21}$$

The indicator ER was computed by calculating the intermediate statistics over the entire evaluation set. Therefore, the total ER can be computed according to the following equation:

$$ER = \frac{\sum_{t=1}^{T} S(t) + \sum_{t=1}^{T} I(t) + \sum_{t=1}^{T} D(t)}{\sum_{t=1}^{T} N(t)}, \tag{22}$$

where $N(t)$ stands for the total number of active sound events from the annotations, and T represents the sum of segments t.

The performance of the approach proposed in this research was evaluated via different conditions, namely different features, different classifiers, and different methods. Meanwhile, several experiments were performed on two public datasets, namely the TUT Sound Events 2016 and TUT Sound Events 2017 datasets.

C. COMPARISON OF DIFFERENT FEATURES

A total of six state-of-the-art features were selected for the experiment, three of which were the short-time Fourier transform (STFT), MFCCs, and the log-mel spectrogram, For the STFT, the audio signals were generated with a sampling rate of 44.1 kHz, and the STFT was calculated with a frame overlap of 50% and a frame size of 40 ms. For each frame, the STFT was calculated on 1024 points, and the dimension was 80 × 513. The other three features were the aggregations of the MFCCs, the log-mel spectrogram, and the STFT; the aggregation of the MFCCs and log-mel spectrogram was F_{ML}, the aggregation of the MFCCs and STFT was F_{MS}, and the aggregation of the log-mel spectrogram and STFT was F_{LS}. In the experiments, to compare the performance of various features performances in the same situation, the same classifiers for CapsNet-RNN were used for all the features. Table 2 reports the results on the TUT Sound Events 2016 and TUT Sound Events 2017 datasets.

Both the F1 and ER values for F_{ML} achieved relative improvements as compared to the use of any other features. The performance of the aggregated feature F_{ML} achieved the highest F1 value of 70.21 and ER value of 0.4 using the TUT Sound Events 2016 dataset, while respective values of 81.35 and 0.57 were achieved on the TUT Sound Events 2017 dataset. In addition, the results indicate that the aggregation of the MFCCs and log-mel spectrogram highlights the class discrimination ability. Moreover, the performances of the three individual input features, namely the MFCCs, log-mel spectrogram, and STFT, were worse than the performances of the aggregated features.

D. COMPARISON OF VARIOUS CLASSIFIERS

In this experiment, the performances of various classifiers were compared in the same situation, and the aggregated feature F_{ML} was used for the six different classifiers. Table 1 displays the parameter settings of CapsNet-RNN The parameter settings of the other classifiers, including the SVM, CNN, RNN, CRNN, and CapsNet classifiers, were optimized with a random search strategy, and the main parameters were set as follows. SVM: sigmoid kernel function, one-vs-rest multiclass training; CNN: five convolutional layers and the ReLU activation function; RNN: two layers with a bidirectional gated recurrent unit and a time-distributed fully-connected (dense) layer; CRNN: three convolutional layers and a recurrent layer with a bidirectional gated recurrent unit; CapsNet: only feature detector (convolution layers) and capsule layers. Table 3 reports the outcomes of the different classifiers.

TABLE 2. **The results obtained by different features.**

TUT Sound Events 2016			TUT Sound Events 2017		
Feature	F1	ER	Feature	F1	ER
MFCC	67.52	0.44	MFCC	71.68	0.65
Log-Mel	67.75	0.41	Log-Mel	77.46	0.60
STFT	59.34	0.51	STFT	67.42	0.71
F_{MS}	67.83	0.43	F_{MS}	78.35	0.62
F_{LS}	67.83	0.45	F_{LS}	77.47	0.68
$\boldsymbol{F_{ML}}$	**70.21**	**0.40**	$\boldsymbol{F_{ML}}$	**81.35**	**0.57**

TABLE 3. **The results of different classifiers.**

TUT Sound Events 2016			TUT Sound Events 2017		
Classifiers	F1	ER	Classifiers	F1	ER
SVM	37.4	0.67	SVM	35.5	0.85
CNN	40.2	0.54	CNN	39.5	0.79
RNN	39.5	0.55	RNN	40.1	0.80
CRNN	66.3	0.43	CRNN	42.3	0.78
CapsNet	65.6	0.67	CapsNet	73.47	0.66
CapsNet-RNN	**70.21**	**0.56**	**CapsNet-RNN**	**81.35**	**0.57**

As presented in Table 3, on the TUT Sound Events 2016 dataset, the CapsNet-RNN network achieved respective improvements of the F1 and ER values of 4.61% and 11% as compared with CapsNet, and improvements of 7.88% and 9%, respectively, on the TUT-Sound Events 2017 dataset.

The best results of the DCASE 2017 Challenge were F1 = 41.7% and ER = 0.79, and those for the DCASE 2016 Challenge were F1 = 66.4% and ER = 0.48. In contrast, the combination of the proposed model and the new features F_{ML} was found to greatly improve the performance. These findings imply that after the inclusion of temporal context information, the performance of the model achieved a relative improvement in detecting polyphonic sound events. In the cases of the CNN and other classifiers, there was only a slight performance improvement due to the loss of position information.

E. COMPARISON OF DIFFERENT METHODS

The method proposed in this study was compared with existing up-to-date methods, and the five compared systems are described as follows:

MFCC-CNN [18]: the model contains three convolutional layers, and the feature is MFCCs;

MFCC-CRNN [18]: the model includes three convolutional layers and a bidirectional GRU layer;

Binaural STFT-CapsNet [19]: the feature is a binaural STFT and the classifier is an original CapsNet network that only includes a feature detector and capsule layers;

TABLE 4. The results of different methods.

TUT Sound Events 2016			TUT Sound Events 2017		
Methods	F1	ER	Methods	F1	ER
MFCC+CNN [18]	59.8	0.56	Binaural mel energy+CRNN [41]	42.9	0.80
MFCC+CRNN [18]	66.4	0.48	Log-Mel+CRNN [41]	41.7	0.79
Binaural STFT+CapsNet [19]	–	0.69	Log-Mel+CapsNet [19]	–	0.58
F_{ML}+CapsNet-RNN	**70.21**	**0.45**	**F_{ML}+CapsNet-RNN**	**81.35**	**0.57**

Log-Mel-CapsNet [19]: the feature is a log-mel spectrogram, and the classifier is an original CapsNet network;

Binaural Mel energy-CRNN [41]: the model uses a CRNN as the classifier, and the binaural mel energy spectrogram as the input feature;

Log-Mel-CRNN [41]: the classifier is CRNN and the input feature is a log-mel spectrogram.

The results reported in previous studies [18], [41] are the best results achieved in the DCASE Challenge. According to the experimental results reported in Table 4, CapsNet [19] outperformed the results in [18], [41]. This indicates that CapsNet can overcome the limitations of CNNs, such as information loss after max-pooling. The proposed method combination of CapsNet and an RNN which yielded the best results and achieved the largest (F1, ER) improvements of (10.41%, 11%) and (33.5%, 23%), respectively. The RNN is used to add the unique position relationship of sound events by learning the temporal context information, as temporal dependency has proven to be important in the sound event analysis task.

V. ANALYSIS IN REAL-LIFE SCENARIOS

The proposed method was applied to a university campus to monitor environmental noise and audio events. The details of the deployment architecture were provided in Section II. The system was used to monitor four classes of events, including "car horn," "scream," "gun shot," and "glass breaking," and all other noise was considered as the background. In the real campus environment, there are a variety of sound sources at various distances from the sensor nodes; therefore, signals with diverse intensities and SNRs can be obtained. The proposed model was trained on the GoogleAudioSet dataset. During the experiment, the audio stream contained 100 car horn events, 89 scream events, 95 gun shot events,90 glass breaking events and 50 background events. Table 5 presents the recognition outcomes. The average F1 value reached 94.86%, and the average ER was only 5.4%.

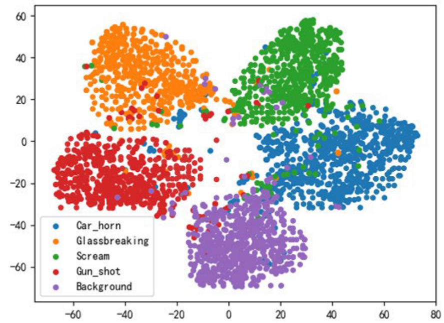


FIGURE 12. The scatter plot of the feature F_{ML}.

TABLE 5. The identification results in real-life scenarios.

Event type	F1	ER
Car horn	93.43	0.094
Scream	95.56	0.050
Gun shot	91.34	0.063
Glass breaking	96.12	0.031
Background	97.86	0.030
Average	**94.86**	**0.054**

The t-SNE algorithm [42] is a non-linear dimensionality reduction visualization method that can map high-dimensional data to a two-dimensional plane. To visualize the feature-learning process step by step, intermediate outputs were extracted from each neural network block. Fig. 12 shows the scatter plot of the features extracted from F_{ML} by the CapsNet, which indicates that the input data F_{ML} became more separable after the proposed network was applied for different sound categories.

In Fig. 13 (A), the identification process of an audio stream is introduced in detail. The output of each layer of the network model was described in Section II. The output vector $\mathbf{z}_i$ of recurrent layers contains historical contextual information of the entire sequence, and then the output of the CapsNet and $\mathbf{z}_i$ are then combined via a residual network to improve the system performance. Finally, the model obtains the identification result related to time.

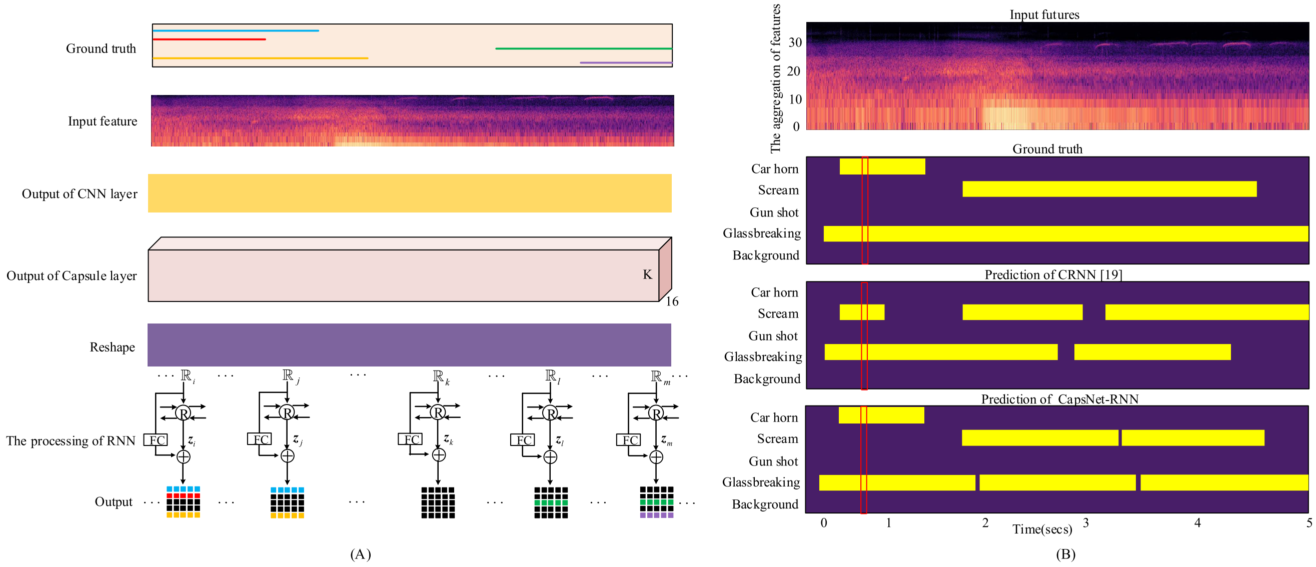
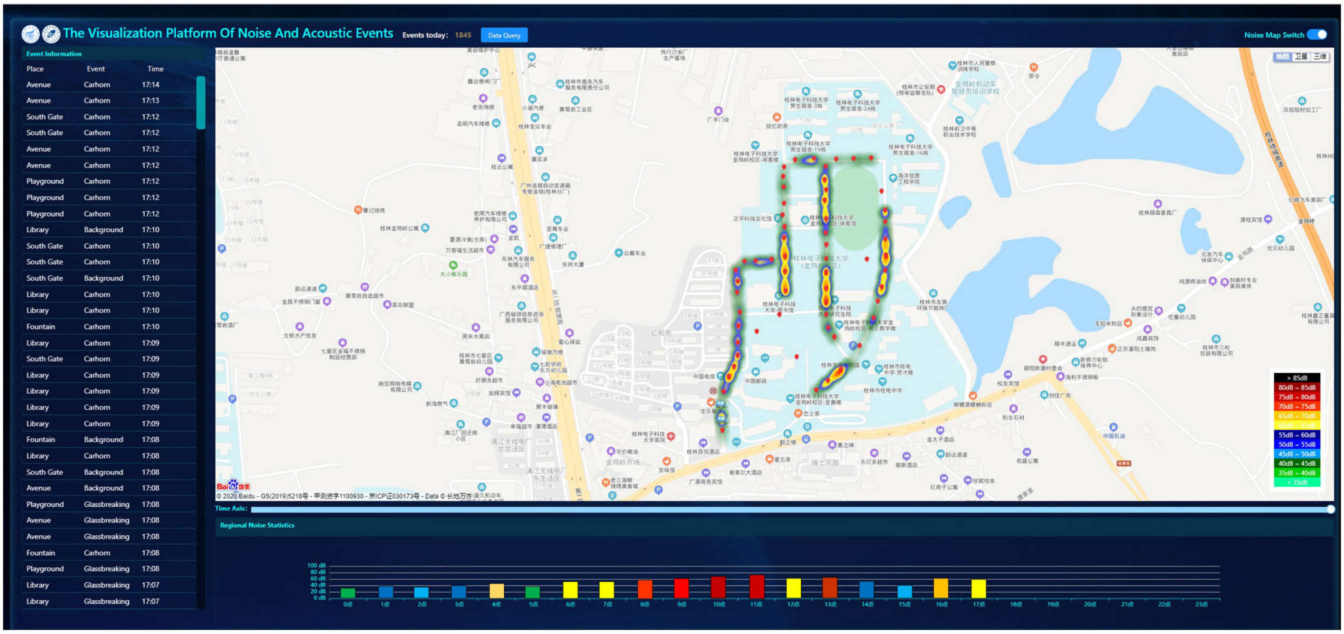


FIGURE 13. (A) The overview of proposed CapsNet-RNN. (B) Input features (a), ground truth (b) and prediction of CRNN (c) and CapsNet-RNN (d) from a sequence of test examples from real-time database.



Place	Event	Time
Avenue	Carhorn	17:14
Avenue	Carhorn	17:13
South Gate	Carhorn	17:12
South Gate	Carhorn	17:12
Avenue	Carhorn	17:12
Avenue	Carhorn	17:12
Playground	Carhorn	17:12
Playground	Carhorn	17:12
Playground	Carhorn	17:12
Library	Background	17:10
South Gate	Carhorn	17:10
South Gate	Carhorn	17:10
South Gate	Background	17:10
Library	Carhorn	17:10
Library	Carhorn	17:10
Fountain	Carhorn	17:10
Library	Carhorn	17:09
South Gate	Carhorn	17:09
Library	Carhorn	17:09
Library	Carhorn	17:09
Library	Carhorn	17:09
Library	Carhorn	17:09
Fountain	Background	17:08
Library	Carhorn	17:08
South Gate	Background	17:08
Avenue	Background	17:08
Playground	Glassbreaking	17:08
Avenue	Glassbreaking	17:08
Avenue	Glassbreaking	17:08
Fountain	Carhorn	17:08
Playground	Glassbreaking	17:08
Library	Glassbreaking	17:07
Library	Glassbreaking	17:07

FIGURE 14. The monitoring platform.

As shown in Fig. 13(B), to better observe the output of CapsNet-RNN, a 5s long audio stream was selected from the real-time database for analysis. According to the design of the output layer, the output of each model is displayed frame by frame. The figure presents the comparison of the event activity probabilities of CapsNet-RNN and the baseline system CRNN [18]. All possible sound events in the audio stream are "car horn," "scream," "gun_shot," "glass breaking," and "background." It can be seen from the figure that the sound of glass breaking lasts for almost the whole sequence, and the proposed model CapsNet-RNN could correctly detect almost all of it; in contrast, the CRNN model resulted in a large amount of loss of the sound of glass breaking. When a car horn and glass breaking occurred simultaneously, the overlap of the two different sound events led to the judgment error of the CRNN, which interpreted the car horn as a scream (as indicated by the red box in the figure). For CapsNet-RNN, the distinction of the polyphonic sound events is obvious, which indicates that the proposed model has a strong ability to distinguish mixed sound sources.

Finally, as shown in Fig. 14 a monitoring platform was developed to visualize the information of the audio events and background. On the map of the monitoring platform, 100 sensor nodes were labelled according to their actual locations. The colored circles represent the values of the noise, and interpolation was performed between two sensor nodes to form a linear distribution. On the map's left side, there is a column that displays the detailed data on sound events, including the location, event, and time. The data about the detected audio events, including the locations, events, and the times of occurrence, can be visualized throughout the day. When the system detects an audio event, the landmark will pulse briefly, and the event information will be shown in the left column.

VI. CONCLUSION

A novel audio detection system was designed in this study, and a total of 100 wireless sensor nodes were deployed to monitor abnormal events on a university campus in real-time. In the proposed system, audio streams are collected by wireless sensor nodes, and are then identified and classified via the proposed CapsNet-RNN network. A monitoring platform was also developed to display the detailed sound event information and indicates when and where abnormal events occur on the campus. In addition, the system can be used for long-term open-air monitoring as it is characterized by low cost, and its power is supplied by a solar panel. The proposed system can also be combined with existing video surveillance equipment for the monitoring of abnormal audio on roads to achieve complementary advantages. To evaluate the performance of the proposed approach for polyphonic audio event detection, detailed experiments were conducted on two public DCASE datasets, and in comparison with the existing algorithms, the proposed method performed better. In a real-life scenario on a university campus, the average F1 value of the proposed model reached 95.1%, and the ER value was only 5.2%. In future research, the proposed system will be applied on a highway to detect accidents to ensure that emergency teams can intervene quickly. Moreover, the architectural advantages of the proposed recognition algorithm will be improved to be better adapted to polyphonic audio.

REFERENCES

[1] J. B Li, K. Ma, S. Qu, P.-Y. Huang, and F. Metze, "Audio-visual event recognition through the lens of adversary," 2020, *arXiv:2011.07430*.

[2] M. Crocco, M. Cristani, A. Trucco, and V. Murino, "Audio surveillance: A systematic review," *ACM Comput. Surv.*, vol. 48, no. 4, pp. 1–46, May 2016.

[3] R. Alsina-Pagès, J. Navarro, F. Alías, and M. Hervás, "HomeSound: Real-time audio event detection based on high performance computing for behaviour and surveillance remote monitoring," *Sensors*, vol. 17, no. 4, p. 854, Apr. 2017.

[4] P. Foggia, N. Petkov, A. Saggese, N. Strisciuglio, and M. Vento, "Reliable detection of audio events in highly noisy environments," *Pattern Recognit. Lett.*, vol. 65, pp. 22–28, Nov. 2015.

[5] J. Salamon and J. P. Bello, "Deep convolutional neural networks and data augmentation for environmental sound classification," *IEEE Signal Process. Lett.*, vol. 24, no. 3, pp. 279–283, Mar. 2017.

[6] S. Krstulovic, "Audio event recognition in the smart home," in *Computational Analysis of Sound Scenes and Events*, 1st ed., T. Virtanen, M. D. Plumbley, and D. Ellis, Eds. Cham, Switzerland: Springer, 2018, pp. 335–371.

[7] L. Xia, G. Chen, X. Xu, J. Cui, and Y. Gao, "Audiovisual speech recognition: A review and forecast," *Int. J. Adv. Robot. Syst.*, vol. 17, no. 6, pp. 1–17, Nov. 2020.

[8] L. Besacier, E. Barnard, A. Karpov, and T. Schultz, "Automatic speech recognition for under-resourced languages: A survey," *Speech Commun.*, vol. 56, pp. 85–100, Jan. 2014.

[9] Y. Li, X. Li, Y. Zhang, M. Liu, and W. Wang, "Anomalous sound detection using deep audio representation and a BLSTM network for audio surveillance of roads," *IEEE Access*, vol. 6, pp. 58043–58055, 2018.

[10] S. Chandrakala and S. L. Jayalakshmi, "Environmental audio scene and sound event recognition for autonomous surveillance: A survey and comparative studies," *ACM Comput. Surv.*, vol. 52, no. 3, pp. 1–34, Jul. 2019.

[11] J. J. Carabias-Orti, T. Virtanen, P. Vera-Candeas, N. Ruiz-Reyes, and F. J. Canadas-Quesada, "Musical instrument sound multi-excitation model for non-negative spectrogram factorization," *IEEE J. Sel. Topics Signal Process.*, vol. 5, no. 6, pp. 1144–1158, Oct. 2011.

[12] G. Guo and S. Li, "Content-based audio classification and retrieval by support vector machines," *IEEE Trans. Neural Netw.*, vol. 14, no. 1, pp. 209–215, Feb. 2003.

[13] X. Zhang, Q. He, and X. Feng, "Acoustic feature extraction by tensor-based sparse representation for sound effects classification," in *Proc. IEEE ICASSP*, Apr. 2015, pp. 166–170.

[14] R. Zhang, X. Tang, P. Gong, P. Wang, C. Zhou, X. Zhu, D. Liang, and Z. Wang, "Low-noise reconstruction method for coded-aperture gamma camera based on multi-layer perceptron," *Nucl. Eng. Technol.*, vol. 52, no. 10, pp. 2250–2261, Oct. 2020.

[15] A. Rabaoui, Z. Lachiri, and N. Ellouze, "Using HMM-based classifier adapted to background noises with improved sounds features for audio surveillance application," *Int. J. Signal Process.*, vol. 5, no. 1, pp. 46–55, 2008.

[16] Z. Mushtaq and S.-F. Su, "Environmental sound classification using a regularized deep convolutional neural network with data augmentation," *Appl. Acoust.*, vol. 167, Oct. 2020, Art. no. 107389.

[17] S. W. Byun, B. R. Shin, S. P. Lee, and H. S. Han, "Emotion recognition from speech using deep recurrent neural networks with acoustic features," *Basic Clin. Pharmacol. Toxicol., Meeting Abstract*, vol. 123, pp. 43–44, Nov. 2018.

[18] E. Cakir, G. Parascandolo, T. Heittola, H. Huttunen, and T. Virtanen, "Convolutional recurrent neural networks for polyphonic sound event detection," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 25, no. 6, pp. 1291–1303, Jun. 2017.

[19] F. Vesperini, L. Gabrielli, E. Principi, and S. Squartini, "Polyphonic sound event detection by using capsule neural networks," *IEEE J. Sel. Topics Signal Process.*, vol. 13, no. 2, pp. 310–322, May 2019.

[20] S. K. Sahu, P. Kumar, and A. P. Singh, "Dynamic routing using inter capsule routing protocol between capsules," in *Proc. AMSS 20th Int. Conf. Comput. Modelling Simulation (UKSim)*, Mar. 2018, pp. 1–5.

[21] G. Sato, M. Konoshima, T. Ohwa, H. Tamura, and J. Ohkubo, "Quadratic unconstrained binary optimization formulation for rectified-linear-unit-type functions," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 99, no. 4, Apr. 2019, Art. no. 042106.

[22] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic routing between capsules," in *Proc. Adv. Neural Inf. Process. Syst*, 2017, pp. 3856–3866.

[23] G. Parascandolo, H. Huttunen, and T. Virtanen, "Recurrent neural networks for polyphonic sound event detection in real life recordings," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2016, pp. 6440–6444.

[24] C. C. Chatterjee, M. Mulimani, and S. G. Koolagudi, "Polyphonic sound event detection using transposed convolutional recurrent neural network," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2020, pp. 661–665.

[25] Y. Xu, Q. Kong, Q. Huang, W. Wang, and M. D. Plumbley, "Convolutional gated recurrent neural network incorporating spatial features for audio tagging," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, May 2017, pp. 3461–3466.

[26] V. Ravi, M. Alazab, S. Srinivasan, A. Arunachalam, and K. P. Soman, "Adversarial defense: DGA-based botnets and DNS homographs detection through integrated deep learning," *IEEE Trans. Eng. Manag.*, early access, Mar. 12, 2021, doi: 10.1109/TEM.2021.3059664.

[27] E. F. Ray, "Applications of attenuations and reflections in ISO 9613-2, acoustics—Attenuation of sound during propagation outdoors—Part 2: General method of calculation," in *Proc. Inter-Noise Noise-Con Congr. Conf.*, Baltimore, MD, USA, 2004, pp. 834–842.

[28] A. Mesaros, T. Heittola, and T. Virtanen, "Metrics for polyphonic sound event detection," *Appl. Sci.*, vol. 6, no. 6, p. 162, May 2016.

[29] J. Kang, J. Kim, K. Kim, and M. Sohn, "Complex activity recognition using polyphonic sound event detection," in *Innovative Mobile and Internet Services in Ubiquitous Computing* (Advances in Intelligent Systems and Computing), L. Barolli, F. Xhafa, N. Javaid, and T. Enokido, Eds., 2019, pp. 675–684.

[30] A. Mesaros, T. Heittola, A. Diment, B. Elizalde, A. Shah, E. Vincent, B. Raj, and T. Virtanen, "DCASE 2017 challenge setup: Tasks, datasets and baseline system," in *Proc. Detection Classification Acoust. Scenes Events Workshop*, Munich, Germany, 2017, pp. 85–92.

[31] A. Mesaros, T. Heittola, and T. Virtanen, "TUT database for acoustic scene classification and sound event detection," in *Proc. 24th Eur. Signal Process. Conf. (EUSIPCO)*, Budapest, Hungary, Aug. 2016, pp. 1128–1132.

[32] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *J. Mach. Learn. Res.*, vol. 13, pp. 281–305, Feb. 2012.

[33] R. Vinayakumar, M. Alazab, K. Soman, P. Poornachandran, A. Al-Nemrat, and S. Venkatraman, "Deep learning approach for intelligent intrusion detection system," *IEEE Access*, vol. 7, pp. 41525–41550, 2019, doi: 10.1109/ACCESS.2019.2895334.

[34] R. Vinayakumar, K. P. Soman, and P. Poornachandran, "Applying convolutional neural network for network intrusion detection," in *Proc. Int. Conf. Adv. Comput., Commun. Informat. (ICACCI)*, Sep. 2017, pp. 1222–1228, doi: 10.1109/ICACCI.2017.8126009.

[35] K. Sreelakshmi, S. Akarsh, R. Vinayakumar, and K. P. Soman, "Capsule neural networks and visualization for segregation of plastic and non-plastic wastes," in *Proc. 5th Int. Conf. Adv. Comput. Commun. Syst. (ICACCS)*, Mar. 2019, pp. 631–636, doi: 10.1109/ICACCS.2019.8728405.

[36] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. ICML*, 2013, vol. 30, no. 1, p. 3.

[37] J. Liang and R. Liu, "Stacked denoising autoencoder and dropout together to prevent overfitting in deep neural network," in *Proc. 8th Int. Congr. Image Signal Process. (CISP)*, 2016, pp. 697–701.

[38] R. Vinayakumar, K. P. Soman, and P. Poornachandran, "A comparative analysis of deep learning approaches for network intrusion detection systems (N-IDSs): Deep learning for N-IDSs," *Int. J. Digit. Crime Forensics*, vol. 11, no. 3, pp. 65–89, Jul. 2019.

[39] R. Vinayakumar, K. P. Soman, and P. Poornachandran, "Evaluating effectiveness of shallow and deep networks to intrusion detection system," in *Proc. Int. Conf. Adv. Comput., Commun. Informat. (ICACCI)*, Sep. 2017, pp. 1282–1289, doi: 10.1109/ICACCI.2017.8126018.

[40] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," *J. Mach. Learn. Res.*, vol. 9, pp. 249–256, May 2010.

[41] S. Adavanne and T. Virtanen, "A report on sound event detection with different binaural features," 2017, *arXiv:1710.02997*.

[42] R. Vinayakumar, M. Alazab, S. Srinivasan, Q.-V. Pham, S. K. Padannayil, and K. Simran, "A visualized botnet detection system based deep learning for the Internet of Things networks of smart cities," *IEEE Trans. Ind. Appl.*, vol. 56, no. 4, pp. 4436–4456, Jul. 2020, doi: 10.1109/TIA.2020.2971952.

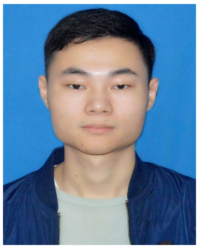
LIYAN LUO was born in June 1987. She received the B.S. degree from the Guilin University of Electronic Technology, Guilin, China, in 2010, and the Ph.D. degree from Xidian University, Xi'an, China, in 2015. She was a Master Tutor at the Guilin University of Electronic Technology. She is currently working at the Postdoctoral Research Station, Guilin University of Electronic Technology, for her postdoctoral research. She has published more than ten research papers in correlative journals and conferences. Her research interests include environmental sound signal processing, indoor positioning, sound source localization, and deep learning.

LIUJUN ZHANG received the B.S. degree in communication engineering from Henan Normal University, Xinxiang, China, in 2018. He is currently pursuing the M.S. degree in information and communication engineering with the Guilin University of Electronic Technology. His research interests include deep learning, acoustic sensing, and audio signal processing.

MEI WANG received the B.S., M.S., and Ph.D. degrees from Xidian University, Xi'an, China, in 1984, 1989, and 2003, respectively. In 2006, she was a Visiting Scholar with the University of Central Florida. She is currently a Ph.D. Tutor with Xidian University and the Guilin University of Electronic Technology. She has published more than 50 research papers in correlative journals and conferences. Her research interests include location awareness and co-location, sensor networks, and energy efficiency optimization.

ZHENGHONG LIU received the M.S. degree from the Guilin University of Electronic Technology. He is currently a Master Tutor with the Guilin University of Electronic Technology, Guilin, China. His major research interests include wireless broadband communication, wireless sensor networks, and radio frequency communication circuit technology research and application development. He has published more than 20 research papers in correlative journals and conferences.

XIN LIU was born in June 1983. He is currently a Master Tutor with the Guilin University of Technology. He is currently an Associate Professor with the Guilin University of Technology, Guilin, China. His research interests include cognitive radio, intelligent anti-jamming communication decision making, and TDOA passive location.



RUIBIN HE received the B.S. degree in marine electrical and electronic engineering from Guangzhou Maritime University, Guangzhou, China, in 2020. He is currently pursuing the M.S. degree in information and communication engineering with the Guilin University of Electronic Technology. His research interests include acoustic sensing and audio signal processing.

YE JIN received the B.S. degree in communication engineering from Yanshan University, Qinhuangdao, China, in 2020. She is currently pursuing the M.S. degree in electronic information with the Guilin University of Electronic Technology University, Guilin, China. Her research interests include acoustic sensing and audio signal processing.