RF Signal-Based UAV Detection and Mode Classification: A Joint Feature Engineering Generator and Multi-Channel Deep Neural Network Approach

Shubo Yang, Yang Luo, Wang Miao, Changhao Ge, Wenjian Sun, Chunbo Luo

Abstract—With the proliferation of Unmanned Aerial Vehicles (UAVs) to provide diverse critical services, such as surveillance, disaster management, and medicine delivery, the accurate detection of these small devices and the efficient classification of their flight modes are of paramount importance to guarantee their safe operation in our sky. Among the existing approaches, Radio Frequency (RF) based UAV detection and flight mode classification are gradually gaining research attention, as RF signals are less affected by complex environmental factors. The similarities between UAV RF signals and the diversity of frequency components make accurate detection and classification a particularly difficult task. To bridge this gap, we propose a joint Feature Engineering Generator (FEG) and Multi-Channel Deep Neural Network (MC-DNN) approach. The FEG first preprocesses the UAV RF data, and the MC-DNN learns the detection and classification. Specifically, in FEG, data truncation and normalization separate different frequency components, the moving average filter reduces the outliers in the RF signal, and the concatenation fully exploits the details of the dataset. In addition, the multi-channel input in MC-DNN separates multiple frequency components and reduces the interference between them. Cross-validation is adopted to prevent DNNs from overfitting and validates DNNs’ performance. A novel dataset that contains ten categories of RF signals from three types of UAVs is used to verify the effectiveness. Experiments show that the proposed method outperforms the state-of-the-art UAV detection and classification approaches in terms of UAVs’ type and flight modes classification accuracy of 98.4% and F1 score of 98.3%.

Index Terms—Unmanned aerial vehicles, UAV detection, UAV mode classification, Feature Engineering Generator, Multi-Channel Deep Neural Network

I. INTRODUCTION

Unmanned aerial vehicles (UAVs), also called drones, are gaining increasing popularity since they have high flexibility, ease of affordability, and exceptional capability. The recent advances in UAV technology have led to the proliferation of aerial services in our sky, e.g., emergency networks [1], healthcare system [2], surveillance system [3][4], disaster rescue [5], and agriculture [6]. Meanwhile, UAVs are employed to improve wireless communication, because they can provide connections between devices [7], increase the energy efficiency through trajectory optimization [8], assist resource allocation [9], and set up relay links [10].

This research was supported by the National Natural Science Foundation of China under grant 61871096, and National Key R&D Program of China under grant 2018YFB2101300.

Shubo Yang and Wenjian Sun are with the University of Electronic Science and Technology of China, Chengdu 611731, China (e-mail: 2018190607005@student.uestc.edu.cn and sunwenjian@uestc.edu.cn).

Yang Luo (corresponding author) and Chunbo Luo are with the University of Electronic Science and Technology of China, Chengdu 611731, China, and Yangtze Delta Region Institute (Huzhou), University of Electronic Science and Technology of China, Huzhou 313001, China (e-mail: luoyang@uestc.edu.cn and c.luo@uestc.edu.cn).

Wang Miao is with the University of Exeter, Exeter EX4 4QF, U.K. (e-mail: wang.miao@exeter.ac.uk).

Changhao Ge is with the University of Glasgow, Glasgow, U.K. (e-mail: 2357902G@student.gla.ac.uk).

Since the UAVs have flexibility, ease of affordability, and controllability, they may be utilized for malicious purposes and thereby pose potential security and privacy threats [14][15]. The threats contain eavesdropping, invading restricted regions, attacks on infrastructure, and colliding with people after losing control. To address these issues, efficient air traffic management becomes indispensable to ensure the safety of UAV flight and management [16], which is critical to both themselves and the flying environment, and property managers need to be aware of an approaching UAV. Hence, as the foundation for the following-up regulating measures, the methods for UAV detection and mode classification are urgently required.

Conventional detection methods, e.g., radars, acoustics, and vision, are constrained under some conditions. For example, the radar-based detection methods are restricted by the UAV sizes, the image-based methods have limits of distinguishing birds from drones, and acoustics-based methods are usually influenced by noise and have short detection distance [17]. Different from the conventional methods, RF signals can be detected at long distances and are less influenced by environmental factors. There is intensive research into classifying UAV types by RF signals. The work in [18] first detected the UAV signals, and then used neighborhood component analysis (NCA) and machine learning classifiers for classification of
and continuously listening to the communication between the operated at frequencies around 2.4 GHz [24]. By passively and modulation techniques [18]. Besides, most UA Vs are uniquely characterize UA Vs due to the UA Vs’ circuitry design signals from UA Vs to controllers. Most RF signals generated while the downlink one contains telemetry signals and video controlling RF signal commands from the controller to UA Vs, the uplink and the downlink. The uplink signal contains the RF signals generated between UA Vs and their controllers: on the UA V operation signals’ characteristics.

A. System model

15 UAV controllers types. Moreover, based on a Native Bayes approach and features of energy transient signal inputted into machine learning algorithms, 14 types of UAV RF signals are classified [19]. Bhattacherjee et al. in [20] utilized a keysight sensor to detect the UAV type by comparing the received RF signature with other UA Vs’ RF signatures in a database. However, most existing works focus on detecting UAV types instead of UAV flight modes, which contain the information of UA Vs’ operation status and are significant for UAV safety management. Therefore, in this paper, we proposed an effective RF signal-based method to not only detect UA Vs’ types but also classify their flight modes.

The challenges of RF signal-based approaches are the similarities contained in the signals and the features of different frequency components. To tackle the challenges, our method first uses a Feature Engineering Generator (FEG) to extract features from RF signals. With the data preprocessed by FEG, we further design a DNN and a multi-channel deep neural network (MC-DNN) to classify the flight modes of UAVs. The multi-channel design separates different frequency components and reduces their corresponding interferences. The effectiveness of the method is verified based on a practical dataset in [21]. The experiment results show that the proposed approach achieves an accuracy of 98.4% and an F1 score of 98.3%, and outperforms other state-of-the-art methods [21] [22] [23].

The rest of the paper is organized as follows. Section 2 summarizes the related work. Section 3 describes the system model and problems. Section 4 presents the RF preprocessing and DNN structure. Section 5 provides the experimental results of our method. Finally, Section 6 draws some conclusions.

SYSTEM MODEL AND PROBLEMS

A. System model

We introduce the system model as in Fig. 1 with a focus on the UAV operation signals’ characteristics.

1) RF signal acquisition: There are mainly two kinds of RF signals generated between UAVs and their controllers: the uplink and the downlink. The uplink signal contains the controlling RF signal commands from the controller to UAVs, while the downlink one contains telemetry signals and video signals from UAVs to controllers. Most RF signals generated uniquely characterize UAVs due to the UAVs’ circuitry design and modulation techniques [18]. Besides, most UAVs are operated at frequencies around 2.4 GHz [24]. By passively and continuously listening to the communication between the UAVs and controllers, the unique RF signals for different types of UAVs with different flight modes can be collected.

Capturing signals of a larger bandwidth gives comprehensive information of different frequency components. However, the devices can have bandwidth constraints to capture the RF signals. Thus, the whole bandwidth is divided into low-frequency and high-frequency components, where each component is captured by respective device.

2) Noise and interference in RF signals: The interference results from other wireless sources that are also operated in the same UAVs operating frequency band, such as Wi-Fi and Bluetooth. Besides, the signal to interference plus noise ratio (SINR) of the captured RF signals is related to the upper limit of classification. Thus, the captured signals with UAVs are modeled as the combination of UAV RF signals and the background signals with noise and interference, captured from the ambient environment without operating UAVs.

3) RF signals in frequency domain: The RF signals are typically captured in the time domain, but the signals in the frequency domain have latent characteristics. Besides, directly using time-domain signals for classification has some drawbacks: First, the time-domain signals usually have a large size of data, which requires high computation resources for preprocessing. The conversion to the frequency domain significantly reduces the data size. Second, if the time-domain signals are further divided into segments, the start and end of the segments are randomly chosen. This may result in a large portion of noise in some segments, and there are not enough features to conduct accurate UAV classifications. Third, many devices used to capture signals have bandwidth limits. The direct concatenation of bandwidth-limited time-domain signals is questionable while the frequency domain spectra are not. To avoid the abovementioned drawbacks and reveal more features, the RF signals are converted into the frequency domain by Fourier Transform.

B. Problems

The RF signal dataset includes the background RF signal and the RF signals of different UAVs with diverse flight modes. The UAV presence must be first classified. Consequently, the types of UAVs and flight modes of UAVs should be classified. The challenges lie in mainly four aspects: the similarities between distinct types of UAV signals, the similarities between the same type of UAVs with different flight modes, the existence of noise and interference, and distinguishing diverse features of different frequency components.

II. METHODOLOGY

To solve the problems, the proposed method consists of FEG preprocessing and MC-DNN. The method concentrates on boosting the discrepancies between each category of RF signals, separating different features represented by each frequency component, and learning representative features.

A. Feature Engineering Generator

The objective of feature engineering is to reveal features from raw data since the features represent the data better,
the more accurate performance obtained. Thus, the FEG aims at separating different frequency components and reducing the similarities of signals. FEG uses three techniques: data truncation and normalization, moving average filter, and concatenation.

1) **Data truncation and normalization:** The dataset of RF signal captured in the frequency domain is composed of low-frequency and high-frequency components. The components may own different features and exhibit different power levels. Thus, normalizing two components together leads to that the frequency component with small values is dominated by the other one. This means that the small values are normalized to nearly zero, and the value changes become almost invisible.

To address this issue, we truncate the two components into two sub-datasets. Each sub-dataset is normalized individually to fully extract different features.

2) **Moving average filter:** Since the existence of noise and the frequency spectra after Fourier Transform have oscillations, a $n$-point moving average filter is proposed to smooth the spectra and reduce the noise effects. While the noise is random, the UAV signals remain almost unchanged. The noise adds destructively in the filter, and the oscillations are reduced while keeping the substantial trend of UAV RF signals. The moving average filter is calculated as

$$
\bar{p} = \frac{1}{n} \sum_{i=0}^{n-1} p_i ,
$$

where $p_i$ is the input value, $\bar{p}$ is the output value, and $n$ is the number of inputs. The output is the mean of adjacent $n$ values. Due to different noise of the frequency components, the parameter $n$ in Eq. 1 is chosen separately. When the frequency signals in the component have larger oscillations, bigger $n$ is expected and more samples are averaged.

3) **Concatenation:** Using individually normalized sub-dataset for classification can’t fully exploit the complete details of the dataset. Hence, the sub-datasets are concatenated to provide a comprehensive view of RF samples. Some concatenation ways alter the sub-datasets and reduce features, such as multiplying coefficients with the first samples of the high-frequency component [21]. Multiplication achieves continuity between the two components, but it results in diminishing small values and changing values. Direct concatenation connects the components without modification and keeps distinct respective features. Besides, the continuity between the low-frequency and high-frequency components is not necessary for classification.

In conclusion, the overall FEG algorithm is presented in Algorithm 1. The RF data are truncated into low-frequency and high-frequency components, resegmented, operated by Fourier Transform, moving average filtered separately, concatenated together, and labeled.

### B. DNN structure

Given the RF signals preprocessed by FEG, DNNs are designed to solve the multi-class classification problem. A well-designed DNN is capable of adapting the relationship between the input and target. In this section, a DNN structure is first designed for classification. Next, the multi-channel technique is applied and a multi-channel input DNN is designed.

1) **Deep neural network:** Based on feedforward artificial neural networks called multilayer perceptron, a DNN in Fig. 2 is designed to classify the RF signals, which includes the model input and output, DNN structure, and loss function.

   a) **Input and target:** The DNN classifies the RF signal data, and the classes are encoded by one-hot encoding into sequences of numbers with 1 representing the corresponding class and 0 representing other classes. Each target for one piece of input data is a vector, and its dimension is the number of classes.

   b) **Deep neural network structure:** The DNN has $H$ hidden layers, with $N_h$ neurons in layer $h$. The leftmost layer is the input layer with $N_{IN}$ neurons, being equal to the dimension of input RF signals preprocessed by FEG. The rightmost layer is the output layer with $N_{OUT}$ neurons, being equal to the number of classes. Each layer receives all the outputs of the previous layer and operates the calculation as follows.

$$
\begin{align*}
    z_l &= W_l^T a_{l-1} + b_l , \\
    a_l &= \delta_l(z_l) ,
\end{align*}
$$

where $a_l$ is the output vector of layer $l$, $a_{l-1}$ is the output vector of the previous layer, $W_l$ is the weight vector, $b_l$ is the
bias vector, and \( \delta_i(\cdot) \) is an activation function, e.g., rectified linear unit function (ReLU) and Softmax function. The weights and biases of each layer are determined through a supervised learning process. A loss function is minimized by a gradient descent algorithm.

c) Loss function: The DNN’s loss function \( L \) is defined as the mean square error between the outputs and the targets as follows.

\[
L(d_i, \hat{d}_i) = \frac{1}{C} \sum_{c=1}^{C} (d_i(c) - \hat{d}_i(c))^2, \tag{3}
\]

where \( d_i \) is a vector of the targets, \( \hat{d}_i \) is a vector of the final layer outputs, and \( C \) is the total number of outputs.

d) Stratified K-fold cross-validation: To estimate the performance and effectiveness of the DNN on a limited dataset, stratified K-fold cross-validation [25] is adopted. The signals and targets are shuffled randomly and divided into \( K \) folds evenly. The number of samples per category in each fold is proportional to the category’s portion in the dataset. There are \( K \) training and testing cycles, where \( K - 1 \) folds are the training set and the remaining fold is for testing. Each fold is used to test once and train \( K - 1 \) times. Training on the same training set can lead to overfitting on the training set and perform badly on unseen data. The overall performance metrics are summarized by taking the mean of \( K \) results.

e) Confusion matrix: The confusion matrix, or error matrix, is used to evaluate the performance of a classifier [26] by giving details into the errors and their types. It visualizes the overall accuracy by comparing the actual targets and predicted classes. The columns of the confusion matrix represent the output class, while the rows represent the predicted classes. Several performance metrics are specified in the confusion matrix, e.g., recall, precision, false discovery rate (FDR), false-negative rate (FNR), accuracy, error, and F1 score.

2) Multi-channel DNN: Various factors contribute to the final classification result and the factors have little correlation with each other. The multi-channel input technique enables the model to consider more possible factors and prevent factors from interfering with each other. Hence, based on DNN, multi-channel DNN (MC-DNN) is designed. The two FEG preprocessed components are input separately to the MC-DNN in Fig. 3. The first channel input is the low-frequency component and the second channel input is the high-frequency component.

The two channels are followed by the first hidden layer, with \( N_{11} \) and \( N_{12} \) neurons connected to two inputs, respectively. Afterwards, there are \( H \) hidden layers and an output layer. The MC-DNN isolates the two frequency components, and better learns the respective classification features for each component.

3) Learning rate decay: Learning rate is a significant hyperparameter in training a DNN. If the learning rate is set too large, the parameters can learn too fast and oscillate around the optimal loss function minimization point without converging. On the contrary, if too small, the parameters can learn too slowly and overfit the training data. Both situations affect the DNN classification performance. Thus, the learning rate cosine decay technique [27] is adopted, where the learning rate decreases as follows.

\[
\eta_t = \frac{1}{2} (1 + \cos(\frac{t \pi}{T})) \eta, \tag{4}
\]

where the total number of epochs is \( T \), \( \eta \) is the initial learning rate, and \( \eta_t \) is the learning rate at epoch \( t \). The learning rate decreases from an initial value \( \eta \) to approximately 0 following the cosine function. The speed of cosine decay is slow at the beginning, linear in the middle, and slow again at the end.

III. EXPERIMENTS

We use the performance of DNN and data without preprocessing as the baseline. Every other step of FEG preprocessing is applied based on previous ones. After cumulatively applying the FEG steps, the preprocessed data are input into MC-DNN for experiments. The entire system model of the FEG and MC-DNN is presented in the flow chart in Fig. 4.

A. Dataset

To verify the effectiveness of our method, the dataset for UAV detection in [21] is adopted. It consists of data from three different types of UAVs: Parrot Bebop, Parrot AR Drone, and DJI Phantom 3. Each type of UAV has four flight modes: mode "On", mode "Hovering", mode "Flying without video recording", and mode "Flying with video recording". The dataset contains 10 categories of RF signals: background with no UAVs, four flight modes of UAV "Parrot Bebop", four flight modes of UAV "Parrot AR Drone" and mode on of UAV "DJI Phantom 3". Each category of data is collected...
by two RF receivers that intercept the UAV’s communications in each flight mode simultaneously. Because the RF receivers have bandwidth constraints, two receivers record low-frequency and high-frequency components, respectively. Then, the received time-domain data are labeled and stored subsequently. The originally captured data segments have $10^7$ samples per segment. Next, the segments are divided into smaller segments with $10^5$ samples to increase the amount of data for further supervised learning. The segments with $10^5$ samples are processed by Fourier Transform into the frequency domain. The high-frequency component and low-frequency component are then concatenated. To ensure the concatenation to be continuous, the first 10 samples of the high-frequency component are multiplied by a coefficient determined by the low-frequency component.

### B. DNN

The parameters for the DNN in Fig. 2 are as follows. DNN has five layers in total, with one input layer, three hidden layers, and one output layer. Each hidden layer has 128 neurons and the output layer has 10 neurons, the same as the number of classes. The activation function for the output layer is the Softmax function, and the one for other layers is the ReLU function. The DNN is trained by an Adam optimizer, minimizing the mean square error loss function. The number of epochs is set to 300 and the batch size is 32. Note that the batch size is set to the powers of 2 to make calculations more efficient. Using the dataset without signal FEG preprocessing as input to DNN, we get a baseline accuracy of 45.9% and an F1 score of 42.0%.

The details of confusion matrix plots are illustrated as follows (see Fig. ?? for an example). Ten inner rows represent the output classes and ten inner columns represent the ten target classes. The diagonal cells in green show the correct predicted samples and rate. Other cells in the inner rows and columns in red correspond to the number and portion of wrongly predicted samples. The top row and the leftmost column in yellow color demonstrate the F1 scores of ten class predictions in green font and the complementary of F1 score in red font. The top leftmost cell in orange averages all the F1 scores and the complementary ones. Besides, the purple bottom row illustrates the recall in green font and FNR in red font. The purple rightmost column presents the precision in green font and the FDR in red font. The bottom rightmost cell in white reveals the average accuracy in black and the complementary error rate in red. The precision, recall, and F1 score can be calculated as follows.

\[
precision = \frac{TP}{TP + FP} \tag{5}
\]

\[
recall = \frac{TP}{TP + FN} \tag{6}
\]

\[
F1\ score = \frac{2 \times precision \times recall}{precision + recall} = \frac{2TP}{2TP + FP + FN} \tag{7}
\]

where $TP$ means true positive, $FP$ means false positive, and $FN$ means false negative.

### C. Joint DNN and Feature Engineering Generator

We evaluate each FEG preprocessing technique step by step cumulatively, and the data are input to the DNN for training.

1) **Data truncation and normalization:** After applying data truncation and normalization, FEG preprocessing steps 1, 2, 3, and 5 in Fig. 4 are implemented. The DNN is trained using the low-frequency component and high-frequency component separately.

The low-frequency component of the data, plotted in Fig. 5(a) is first truncated and normalized. Note that the plotted low-frequency spectra are processed by a 10-point average filter for visualization. The low-frequency component of data is used to train the DNN, which achieves an accuracy of 52.5% and an F1 score of 47.1%. The confusion matrix of training DNN using the normalized low-frequency component is plotted in Fig. 6(a). The accuracies of some categories of flight modes are around 25%, which means the category is not classified correctly. The performance requires further improvements.

The high-frequency component in Fig. 5(b) is preprocessed in the same way as the low-frequency one, i.e., truncated and normalized. Note that the high-frequency component spectra plot is also processed by a 10-point average filter. Improved accuracy of 85.4% and F1 score of 84.1% are achieved by training DNN using the high-frequency component. The great improvement is because the high-frequency component has fewer similarities and more distinct features between the ten categories of signals. Besides, the data truncation and...
normalization avoid the high-frequency component from being dominated by the low-frequency component. The confusion matrix of using the high-frequency component to train is plotted in Fig. 6(b).

2) Moving average filter: In addition to the steps in section III-C.1, the performance of the moving average filter (step 4) is evaluated in this subsection. Specifically, the two components are processed by steps 1-5 and used to train DNN separately. Because the characteristics of each component of data are different, the parameter $n$ is different. To find an optimal result for each component, a sequence of values for the moving average filter parameter $n$ is tested. The accuracies and F1 scores are illustrated in Table I. Experiment results show that using moving average filters is effective on both components.

### Table I

| Data                     | n-point moving average filter |
|--------------------------|------------------------------|
|                          | 0   | 5   | 10  | 15  | 20  | 25  | 30  | 35  | 40  | 45  | 50  |
| Low-frequency component  | accuracy (%) | 52.5 | 58.7 | 58.0 | 61.1 | **65.5** | 64.1 | 60.8 | 64.1 | 61.7 | 63.3 | 62.9 |
|                          | F1 score (%) | 47.1 | 53.1 | 52.0 | 56.7 | **62.2** | 59.5 | 55.8 | 60.5 | 56.8 | 59.7 | 59.2 |
| High-frequency component | accuracy (%) | 85.4 | 84.2 | 82.7 | 89.0 | 87.9 | 87.2 | 89.8 | 89.6 | **90.6** | 91.5 | 90.4 |
|                          | F1 score (%) | 84.1 | 82.8 | 81.2 | 88.0 | 86.8 | 85.8 | 88.8 | 88.5 | **89.7** | 88.4 | 89.4 |

The accuracies and F1 scores of DNN with low-frequency data and high-frequency data processed by $n_l$ and $n_h$ point moving average filters as inputs, respectively. Preprocessing steps 1-5 in the flow chart are implemented.

Fig. 5. The spectra of the low-frequency component, high-frequency component, and direct concatenation of filtered and normalized components. Class 1 is background noise and class 2-10 are different flight modes of three kinds of UAVs.

(a) The average spectra of the low-frequency component of RF signals. Preprocessing steps 1, 2, 3, and 5 in the flow chart are implemented.
(b) The average spectra of the high-frequency component of RF signals. Preprocessing steps 1, 2, 3, and 5 in the flow chart are implemented.
(c) The average spectra of the concatenation of the 20-point moving average filtered and normalized low-frequency component and the 40-point moving average filtered and normalized high-frequency component. Preprocessing steps 1-6 in the flow chart are implemented.

Fig. 6. The confusion matrices of the DNN that take input of low-frequency component, high-frequency component, and direct concatenation of filtered and normalized components. Class 1 is background noise and class 2-10 are different flight modes of three kinds of UAVs.

(a) The confusion matrix of the DNN with normalized low-frequency component of data as input.
(b) The confusion matrix of the DNN with normalized high-frequency component of data as input.
(c) The confusion matrix of the DNN with the concatenation of filtered and normalized components as input.
The frequency signals have reduced noise and more distinct features after filtering. The low-frequency component achieves an accuracy of 65.5% and an F1 score of 62.2% after being preprocessed by a 20-point moving average filter. The 40-point moving average filtered high-frequency component achieves an accuracy of 90.6% and an F1 score of 89.7%.

3) Concatenation: All preprocessing steps 1-6 in FEG preprocessing are implemented. The low-frequency component filtered by the 20-point moving average filter and the high-frequency component filtered by the 40-point moving average filter are concatenated directly. The concatenated data are shown in Fig. 5(c). The accuracy of DNN trained with the concatenated data is 97.3% and the F1 score is 97.1%. The resulting confusion matrix is presented in Fig. 6(c).

D. Joint MC-DNN and Feature Engineering Generator

The DNN is developed into the MC-DNN, and then the learning rate decay is added. The input in this section is the data fully processed by FEG.

1) Multi-channel input: The designed MC-DNN in Fig. 3 has double-channel inputs, the first hidden layer consisting of two parts for two inputs, three hidden layers, and one output layer. There are 256 neurons in the first hidden layer, and 128 neurons in other hidden layers. The result confusion matrix of using the preprocessed data to train MC-DNN is shown in Fig. 7(a) where the accuracy is improved to 98.1% and the F1 score is improved to 97.9%.

2) Learning rate decay: The initial learning rate $\eta$ in Eq. (4) is set to 0.01. After using the fully FEG preprocessed data as input, the result confusion matrix is presented in Fig. 7(b) increasing the accuracy to 98.4% and the F1 score to 98.3%.

E. Comparison

The performance comparison between FEG techniques and DNN structures is presented in Table I as shown in Table II the accuracy and F1 score are achieved with each preprocessing technique and DNN structure are additionally applied. Eventually, our method achieves an accuracy of 98.4% and an F1 score of 98.3% for the classification. Meanwhile, the FEG plays an important role in advancing the performance, improving from 45.9% to 97.3%. The DNN structure improves from 97.3% to 98.4%. The baseline accuracy and F1 score of training DNN are only 45.9% and 42.0% since techniques are not applied to extract and learn the signal features. The improved accuracy and F1 score verify the effectiveness of our method.

Our method also outperforms other methods [21] [22] [23] applying on the same dataset. The work in [21] designed a three-layer DNN for classification with the original dataset directly as input. The overall accuracy and F1 score obtained are 46.8% and 43.0%. Furthermore, a Convolutional Neural Network (CNN) was designed for classification in [22], and the CNN consists of one dimensional (1D) convolutional layers, average 1D pooling layers, and a dropout layer. The proposed model derives an accuracy of 59.2% and an F1 score of 55.1% for the ten-class classification. The multi-channel 1D CNN in [23] includes a feature extractor and a classical MLP.

The multi-channel 1D CNN uses Adam optimizer and cross-entropy loss function. An accuracy of 87.4% and an F1 score of 77% are obtained by this model.

IV. Conclusion

We proposed a joint approach of FEG and MC-DNN to detect UAV presence and classify UAV flight modes. The challenges of RF classification mainly focus on the high similarities between categories of RF UAV signals and the different characteristics represented by frequency components of data. To address these challenges, our method first preprocessed the RF signals by FEG using data truncation and normalization, moving average filter, and concatenation. A carefully designed MC-DNN with learning rate cosine decay, modified based on DNN, was proposed to classify the preprocessed data. The experiments showed the effectiveness of our method, which classifies ten categories with an accuracy of 98.4% and F1 score of 98.3%, and outperforms the state-of-the-art solutions. The proposed method could be extended by other researches on UAV detection and classification performance improvement, including more effective feature extraction as well as novel classification models focusing on finer frequency details.

References

[1] N. Zhao, W. Lu, M. Sheng, Y. Chen, J. Tang, F. R. Yu, and K. Wong. UAV-Assisted Emergency Networks in Disasters. IEEE Wireless Communications, 26(1):45–51, 2019.
[2] Sana Ullah, Ki-II Kim, Kyong Hoon Kim, Muhammad Imran, Pervez Khan, Eduardo Tovar, and Farman Ali. UAV-enabled healthcare architecture: Issues and challenges. Future Generation Computer Systems, 97:425 – 432, 2019.
[3] Amarjot Singh, Devendra Patil, and SN Omkar. Eye in the Sky: Real-Time Drone Surveillance System (DSS) for Violent Individuals Identification Using ScatterNet Hybrid Deep Learning Network. June 2018.
[4] E. Semsch, M. Jakob, D. Pavlicek, and M. Pecheur. Autonomous UAV Surveillance in Complex Urban Environments. 2:82–85, 2009.
[5] Jiong Dong, Kaoru Ota, and Miaoxiang Dong. UAV-based Real-Time Survivor Detection System in Post-disaster Search and Rescue Operations. IEEE Journal on Miniaturization for Air and Space Systems, pages 1–1, 2021.
[6] P. Tripicchio, M. Satler, G. Dabiasis, E. Ruffaldi, and C. A. Avizzano. Towards Smart Farming and Sustainable Agriculture with Drones. pages 140–143, 2015.
[7] Y. Zeng, R. Zhang, and T. J. Lim. Wireless communications with unmanned aerial vehicles: opportunities and challenges. IEEE Communications Magazine, 54(5):36–42, 2016.
[8] Y. Zeng and R. Zhang. Energy-Efficient UAV Communication With Trajectory Optimization. IEEE Transactions on Wireless Communications, 16(6):3747–3760, 2017.
[9] Phuong Luong, Francois Gagnon, Le-Nam Tran, and Fabrice Labeau. Deep Reinforcement Learning Based Resource Allocation in Cooperative UAV-Assisted Wireless Networks. IEEE Transactions on Wireless Communications, pages 1–1, 2021.
[10] Bing Li, Shengjie Zhao, Rongqing Zhang, and Liqing Yang. Full-Duplex UAV Relaying for Multiple User Pairs. IEEE Internet of Things Journal, 8(6):4657–4667, 2021.
[11] K. Horapong, D. Chandrucka, N. Montree, and P. Buawan. Design and use of “Drone” to support the radio navigation aids flight inspection. pages 1–6, 2017.
[12] S. Sekander, H. Tabassum, and E. Hossain. Multi-Tier Drone Architecture for 5G/5G Cellular Networks: Challenges, Trends, and Prospects. IEEE Communications Magazine, 56(3):96–103, 2018.
[13] L. Gupta, R. Jain, and G. Vaszkun. Survey of Important Issues in UAV Communication Networks. IEEE Communications Surveys Tutorials, 18(2):1123–1152, 2016.
The comparison of different combinations of FEG and DNN techniques and against existing methods. The FEG preprocessing techniques are implemented cumulatively, and the number represents the preprocessing steps in Fig. 4. Since there are truncation and concatenation, the accuracies for low-frequency and high-frequency components are separately labeled in the brackets.

| Method                                      | Accuracy | F1 score |
|---------------------------------------------|----------|----------|
| Unpreprocessed data+DNN                     | 45.9%    | 42.0%    |
| Preprocessing steps 1,2,3,5+DNN             | 52.5% (low) | 85.4% (high) |
| Preprocessing steps 1-5+DNN                 | 65.5% (low) | 90.6% (high) |
| Preprocessing steps 1-6+DNN                 | 97.3%    | 97.1%    |
| Preprocessing steps 1-6+MC-DNN              | 98.1%    | 97.9%    |
| Preprocessing steps 1-6+MC-DNN+Learning rate decay | 98.4%    | 98.3%    |
| Classification method in [21]                | 46.8%    | 43.0%    |
| Classification method in [22]                | 59.2%    | 55.1%    |
| Classification method in [23]                | 87.4%    | /        |

Fig. 7. The confusion matrices of training MC-DNN and MC-DNN with learning rate decay using the fully FEG preprocessed data. Class 1 is background noise and class 2-10 are different flight modes of the three kinds of UAVs.