

Research Article

Specific Emitter Identification Based on a Hybrid Deep Neural Network for ACARS Authentication

Liyan Yin , Xin Xiang, Kun Liu, and Yuan Liang

Aviation Engineering College, Air Force Engineering University, Xi'an 710038, China

Correspondence should be addressed to Liyan Yin; 13379547517@126.com

Received 24 April 2022; Accepted 10 August 2022; Published 22 September 2022

Academic Editor: Yunfei Chen

Copyright © 2022 Liyan Yin et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

With the increasing variety and quantity of aircraft, there is a potential threat to the security of the Aircraft Communications Addressing and Reporting System (ACARS) due to the lack of reliable authentication measures. This paper proposes a novel specific emitter identification (SEI) algorithm based on a hybrid deep neural network (DNN) for ACARS authentication. Our deep learning architecture is a combination of Deep Residual Shrinkage Network (DRSN), Bidirectional-LSTM (Bi-LSTM), and attention mechanism (AM), which perform the functions of local and global feature learning and feature focusing, respectively, so that the individual information hidden in the signal waveform can be thoroughly mined. We introduce soft thresholding as a nonlinear transformation in the DRSN to enhance robustness against noise and adopt a low-cost training strategy for new data using transfer learning. The proposed SEI algorithm is optimized and evaluated based on real-world ACARS signals captured in the Xianyang airport. Experimental results demonstrate that our algorithm can distinguish authorized entities from unauthorized entities and obtain an identification accuracy of up to 0.980. In addition, the design rationality and the superiority over other algorithms are verified through the experiments.

1. Introduction

Aircraft Communications Addressing and Reporting System (ACARS) is a datalink communication system established between aircraft and ground stations via the very high frequency/high frequency (VHF/HF) channel, which enables real-time transmission of crucial information, including aircraft registration number, planned track, position coordinates, health conditions [1], etc. The ACARS is widely used in civil aviation due to its long-term operation and adequate surface infrastructure. However, ACARS is transmitted in clear text over the open radio frequency channel [2]. Therefore, it is vulnerable to threats posed by unauthorized entities that may disguise and tamper with information using low-cost transceivers. In particular, active attackers try to attempt to exploit avionic systems or create confusion for air traffic control (ATC), thereby jeopardizing flight security [3, 4].

Commercial airlines have adopted some authentication measures based on bit-level security mechanism, such as

the ACARS message security (AMS) protocol defined in ARINC Specific 823 [5], which run above the physical layer of the Open System Interconnection (OSI) reference model. But they are still at risk of being cracked by using compromised or broken encryption keys, or by impersonating an authorized device's identity [6]. Fortunately, it is almost impossible for unauthorized entities to imitate the intrinsic characteristics at the physical layer, i.e. radio frequency fingerprints (RFFs). The technique for identifying individual emitters using RFFs is called SEI [7], enabling the authentication system to work without large-scale infrastructure modification or protocol update, as shown in Figure 1. Besides, SEI provides an additional guarantee of security to detect unauthorized entities; therefore it can be used as a low-cost complement to traditional authentication schemes.

In the past years, machine learning (ML) has been proved to be an effective and efficient approach to realizing SEI [8]. Feature engineering is one of the most critical aspects of ML, which determines the upper bound of the SEI. ML relies on manually extracted features, including

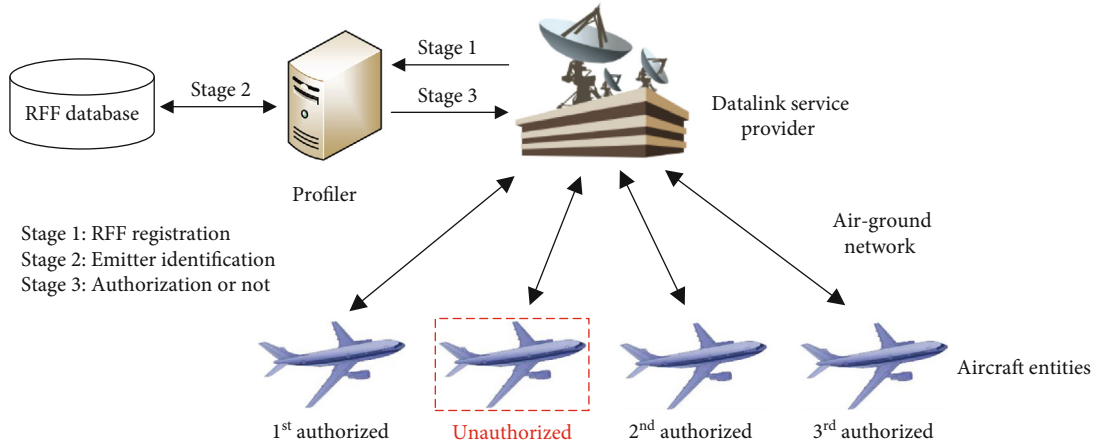


FIGURE 1: The SEI-aided ACARS authentication system, which contains RFF registration, emitter identification and authorization.

frequency and phase offset [9], IQ imbalance [10], discrete wavelet transformation (DWT) [11], nonlinear characteristics [12], etc. However, these features must be extracted for the specific transient period or steady-state period of a complete transmission, thereby undermining the generality of SEI [13]. At the same time, due to the short duration of the transient period, it is difficult to accurately extract the features of the transient signal. Steady state signals are also vulnerable to the impact of the acquisition environment, resulting in the distortion of the features. Furthermore, fine feature engineering consumes high computational costs and depends on professional experience and domain knowledge. Therefore, researchers tend to find more intelligent means to replace ML for implementing SEI [14].

Thanks to advances in computer hardware and algorithms, deep learning (DL), as a particular version of ML, has achieved great success in the field of image and natural language processing [15, 16], etc. Recently, various deep architectures including convolutional neural network (CNN), long short-term memory (LSTM), and residual network (ResNet) have illustrated the outstanding potential for radio signal classification. Quite a few SEI approaches still use the manually extracted features as the inputs of the DNN, such as Bispectrum [17], Hilbert-Huang transform (HHT) [18], and differential constellation trace figure (DCFT) [19]. However, the most significant advantage of DL, namely automatic feature extraction, has not been exploited to the full. In other works, traditional features are abandoned, and deep features are extracted directly from the raw time-series signal. Merchant et al. developed a framework for training a CNN using time-domain complex baseband error signals of the ZigBee devices [20]. Wang et al. designed an efficient SEI method for the Internet of things (IoT) based on a novel complex-valued neural network (CVNN) [21]. Wu et al. proposed an LSTM-based recurrent neural network (RNN) model that captures hardware-specific features of IQ data at the output of the analog-to-digital (ADC) of the USRP transmitter [22]. Of course, there are also a few works that discuss the application of deep learning to aircraft radio fingerprint identification. Zha et al. converted ADS-B signals to Contour Stellar

Images (CSI) and applied the architectures of AlexNet and GoogleNet to SEI [23]. Jian et al. used a deep architecture named ResNet-50-1D to capture salient, discriminative features from IQ samples transmitted by ADS-B radios [24]. Chen et al. used the inception-residual neural network model structure for large-scale ACARS and ADS-B radio signal classification [25]. These works focus on designing suitable network structures for a stronger capability of feature learning. Nevertheless, these deep architectures are designed for their respective target signals, and it is necessary to perform specific preprocessing in front of the network to guarantee the proper identification performance of the ACARS signals.

In this paper, we propose a novel SEI algorithm under an end-to-end DL architecture for ACARS authentication. First, the valid part of the raw signal is intercepted through preprocessing. Then, the inputs are propagated into a hybrid architecture composed of DRSN, Bi-LSTM, and AM, which perform the functions of local feature learning, global feature learning, and feature focusing, respectively. To the best of our knowledge, this is the first attempt to use a joint CNN and RNN-based architecture on SEI in the authentication system. Among them, DSRN uses a structure with soft thresholding to enhance the noise elimination capability. Based on the real-world captured ACARS signals from seven civil aircraft in Xianyang airport, a series of trials and experiments are carried out for hyperparameter selection and performance validation. Through discussions of the experimental results, the feasibility and superiority of the SEI algorithm are adequately studied. Our main contributions are as follows:

- (i) According to the ACARS protocol, the method of signal preprocessing is studied, in which we mainly discuss how to intercept the valid signal. This will enhance the efficiency and accuracy of our algorithm
- (ii) A hybrid DRSN-BiLSTM-AM deep architecture is proposed for the SEI of the ACARS authentication system. The hyperparameters of the model are

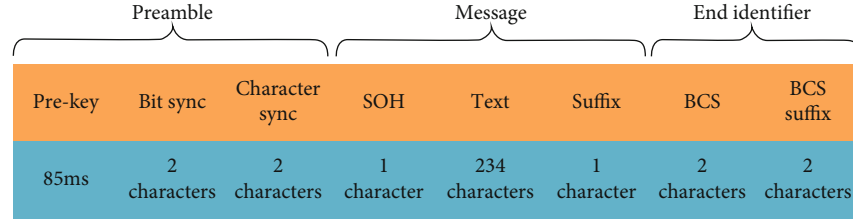


FIGURE 2: Frame structure of the transmission packet in ACARS.

TABLE 1: Character structure of transmission packet in ACARS.

Component	Length	Value
Pre-key	85 ms	All binary “ones”
Bit sync	2 characters	<+> (0x2B), <*> (0x2A)
Character sync	2 characters	<+> (0x2B), <*> (0x2A)
Start of header	1 character	<SOH> (0x01)
Text	234 characters maximum	\
Suffix	1 character	<ETX> (0x03) or <ETB> (0x17)
BCS	2 characters	\
BCS suffix	1 character	 (0x7F)

adjusted to the optimization through trials. In this way, the computational complexity and identification accuracy are balanced well

- (iii) Soft thresholding acts as a layer of the DRSN that performs a nonlinear transformation, and the branch of adaptive threshold selection is added simultaneously. This will make our DNN model insensitive to noise
- (iv) The strategy of transfer learning is introduced using limited training samples. This will greatly reduce the cost of training of our algorithm on new data

The remainder of this paper is organized as follows: Section 2 briefly describes the ACARS protocol and the corresponding preprocessing technique. Section 3 proposes the hybrid DNN model. Section 4 illustrates the implementation of the proposed algorithm. The comparative experimental results on a real-world dataset are covered in Section 5. And, the conclusion is given in Section 6.

2. Brief Description of ACARS and Signal Preprocessing

2.1. Brief Description of ACARS. The research content of this paper is limited to the downlink signal in VHF band of ACARS, whose protocol is defined in the Specification 618 by Aeronautical Radio, Inc. (ARINC) [26]. The protocol aims to create character-oriented data connectivity between aircraft and ground service providers. The encoding scheme of ACARS adopts non-return-zero inverse (NRZI) with a bit rate of 2400 bps. Besides, ACARS adopts amplitude modulation-minimum shift keying (AM-MSK) composite modulation, and its carrier frequency operates around 131.55 MHz.

The default format of the transmission packet in ACARS is depicted in Figure 2, which is made up of three parts: Preamble, Message, and End Identifier. The Preamble consists of Pre-key, Bit sync, and Character Sync. The Message consists of Start of Header (SOH), Text, and Suffix. The End Identifier consists of Block Check Sequence (BCS) and BCS Suffix. Table 1 summarizes the detailed character structure of the transmission packet.

2.2. Signal Preprocessing. In this subsection, we investigate which part of the captured signal is considered as valid. As depicted in Table 1, pre-key is of indefinite length consisting of all binary “ones”. Its role concludes with receiver AGC settling, transmitter power output stabilization, and local oscillator synchronization [26]. The bit sync, character sync, and the start of heading specify their respective character formats. These four components, namely Pre-key, Bit Sync, Character Sync, and the Start of Header are the most distinguishing part of the received signal. Furthermore, they conform to the standard message format and are not affected by the different transmitted content. Therefore, these three components best represent individual characteristics, and the other components are discarded.

Next, we discuss how to locate and intercept the valid signal using the synchronization sequences, including the Bit Sync and Character Sync. Figure 3 depicts the preprocessing framework proposed in this paper, and Figure 4 shows the output waveforms of several important steps.

As depicted in Figure 4(a), the received ACARS signal $r(t)$ is given by,

$$r(t) = R(t) \cos(2\pi f_c t + \varphi(t)) + n(t), \quad (1)$$

where $R(t)$, $\varphi(t)$, f_c and $n(t)$ is the envelope, phase shift,

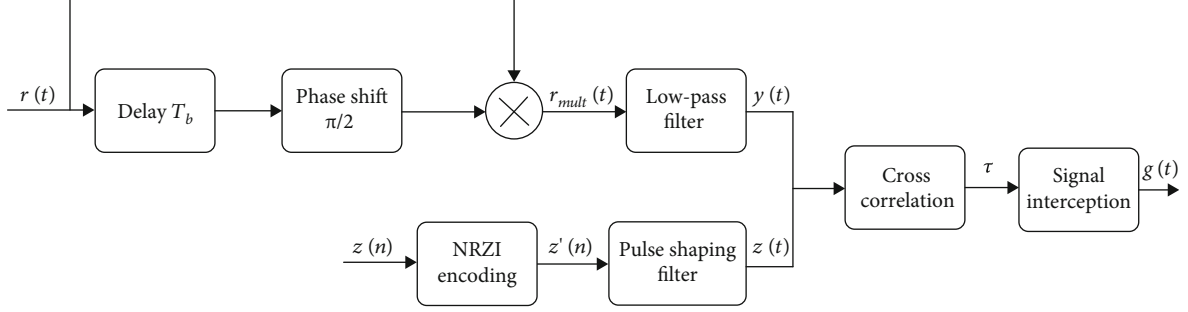


FIGURE 3: Framework of the preprocessing for the ACARS signal.

carrier frequency, and additive white Gaussian noise (AWGN), respectively.

In order to obtain accurate timing information, we adopt a method based on 1-bit differential noncoherent demodulation [27], which does not require precise carrier recovery and has a simple structure.

The delay-multiply signal is obtained with delay of T_b and shifted-phase of $\pi/2$,

$$r_{mult}(t) = R(t) \cos [2\pi f_c t + \varphi(t)] R(t - T_b) \cos [2\pi f_c (t - T_b) + \varphi(t - T_b)] + n'(t). \quad (2)$$

After low pass filtering, the high-frequency components are removed. We can obtain,

$$y(t) = \frac{1}{2} R(t) R(t - T_b) \sin [2\pi f_c T_b + \varphi(t) - \varphi(t - T_b)] + n''(t), \quad (3)$$

where $T_b = 1/2400$ s is the bit width, and $n''(t)$ denotes the noise component. According to the characteristics of MSK modulated signal, we have $f_c T_b = k$, $k = 1, 2, 3, \dots$, then $y(t)$ can be rewritten as,

$$y(t) = \frac{1}{2} R(t) R(t - T_b) \sin [\varphi(t) - \varphi(t - T_b)] + n''(t). \quad (4)$$

Since $R(t)$ and $R(t - T_b)$ is always positive, $y(t)$ has the same polarity as $\varphi(t) - \varphi(t - T_b)$. Therefore, when the transmission bit is 1, $y(t)$ is positive; when the transmission bit is 0, $y(t)$ is negative. To estimate starting time τ of the synchronization sequence, we do not perform sampling and decision. Instead, we correlate the ideal synchronization waveform $z(t)$ and the received signal $y(t)$. As depicted in Table 1, the synchronization sequence $z(n)$ consists of 4 characters, which is generated through ASCII encoding by a bit sequence as, 00101011001010100001011000010110. After NRZI encoding for $z(n)$, we obtain $z'(n)$ as,

$$z'(n) = -111 - 1 - 1111 - 111 - 1 - 111 - 11 - 111 - 1 - 11 - 11 - 11 - 11 - 1 - 1111 - 1. \quad (5)$$

Thereafter, the continuous training signal is given by,

$$z(t) = \sum_{n=1}^{32} z'(n) \cdot \left(\delta \left(t - \frac{n}{R_b} \right) * g(t) \right), \quad (6)$$

where $*$ is the convolution operation, and $g(t)$ is the pulse shaping filter, which is given by,

$$g(t) = \begin{cases} 1 & t \in (0, T_b] \\ 0 & \text{otherwise} \end{cases}. \quad (7)$$

The cross-correlation between $z(t)$ and $y(t)$ is given by,

$$corr(\tau) = \int_0^{T_z} z^*(t) y(t - \tau) dt, \tau \in (0, T_z], \quad (8)$$

where T_z is the time span of $z(t)$. As depicted in Figure 4(b), the starting of the synchronization sequence τ is given by

$$\tau = \arg \max_{\tau} corr(\tau). \quad (9)$$

Once τ is determined, we can locate the valid signal in $r(t)$. Marked by τ in the received ACARS signal $r(t)$, signals of lengths $T_p = 85$ ms and $T_q = 40T_b = 16.67$ ms are intercepted at the forward and backward directions, respectively, which is shown in Figure 4(c). In fact, in order to ensure that there are no redundant sampling points during signal segmentation in the subsequent operation, 1.83 ms at the end of the signal is discarded. Thus, the duration of the valid signal $v(t)$ is $T_v = 99.84$ ms, as shown in Figure 4(d). Note that the objects described above are continuous signals, but the captured signals are processed in a discrete form. We collect signals at the sampling rate of 400 kHz in the experiments. Therefore, the signal of T_v is discretized into 39936 sampling points. For the sake of presentation, we use the raw and original data to represent the ACARS signals before and after the signal interception.

The original signal is then divided into 39 nonoverlapping segments and each segment contains 1024 complex sampling points. The segments are randomly grouped into training dataset and testing dataset by the ratios of 80% and 20%. It should be noted that all the segments in one sample belong to the same dataset. For the training dataset, the segments are labeled with the true emitter category, and then the order of them is randomly shuffled.

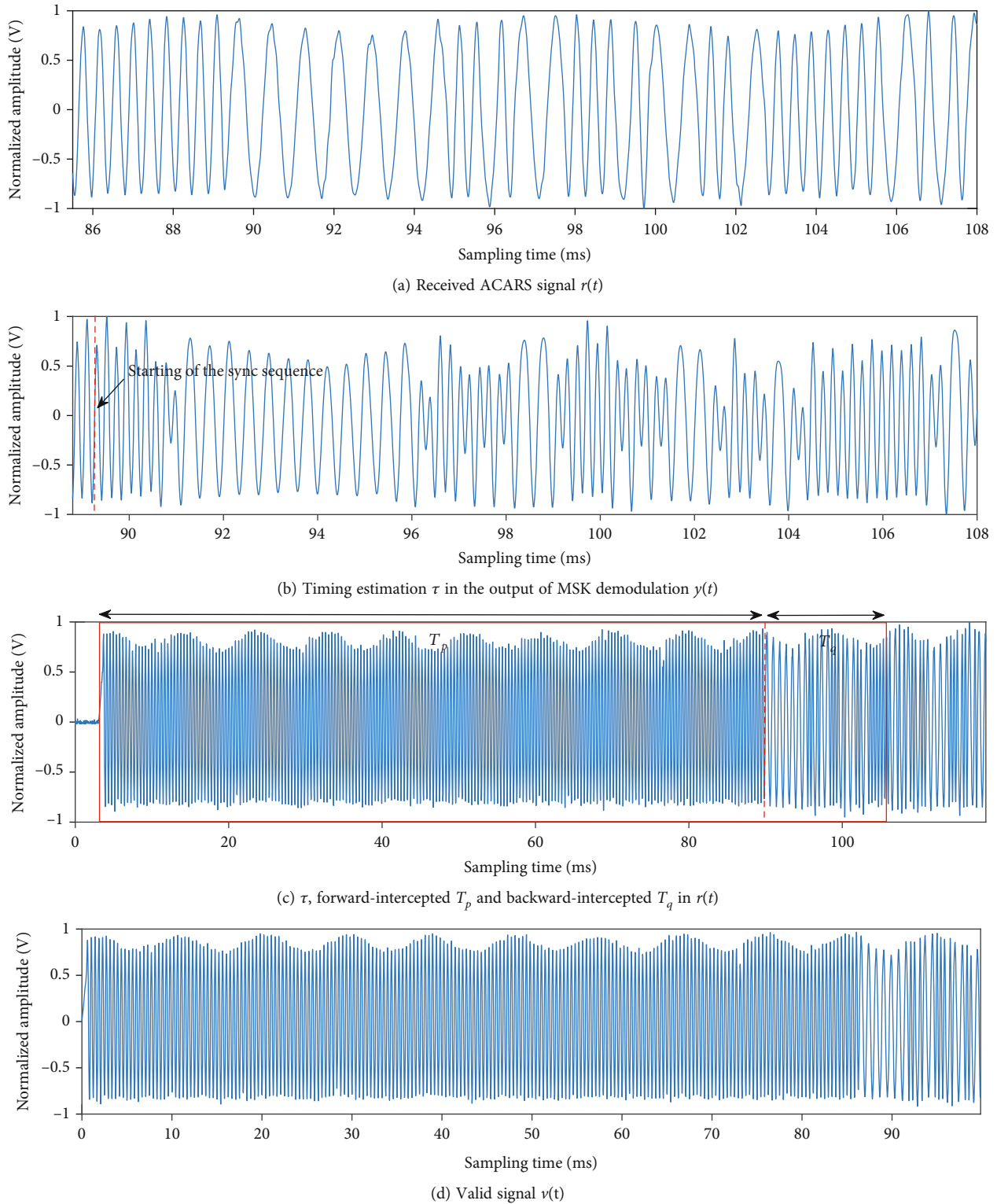


FIGURE 4: Waveform diagrams of several key steps in signal preprocessing.

3. The Hybrid DNN Model

This section describes the deep learning architecture and its main components of our algorithm. In order to extract features more effectively and improve identification perfor-

mance, DRSN, Bi-LSTM, and AM are integrated into a hybrid DNN model. As depicted in Figure 5, the proposed architecture consists of four essential blocks: local feature learning block, global feature learning block, attention block, and identification block. At first, DRSN is used to extract

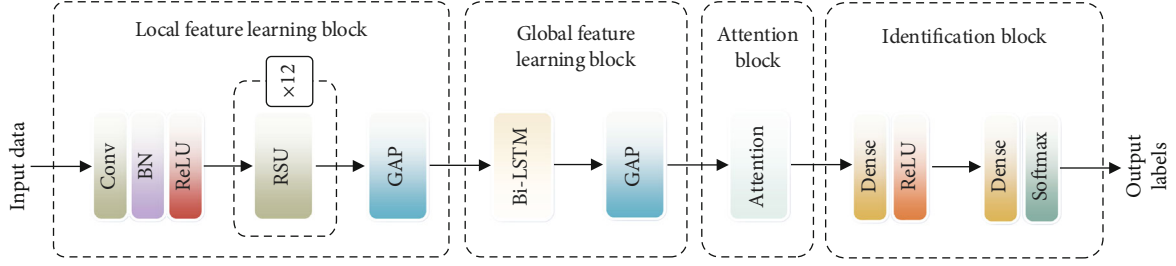
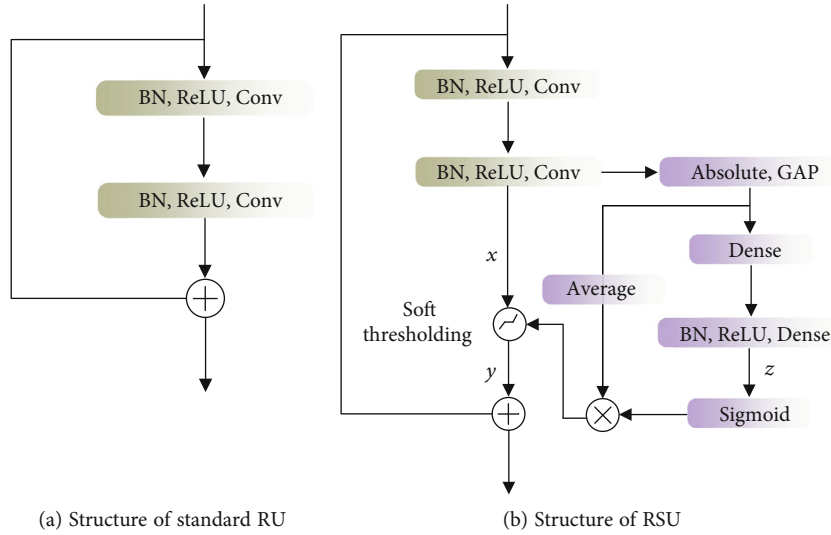


FIGURE 5: Architecture of the proposed network.



(a) Structure of standard RU

(b) Structure of RSU

FIGURE 6: Comparison on two structures of residual unit.

TABLE 2: Local feature learning network parameters.

Unit (number)	Number/size/stride of kernels	Output dimension
Input	—	1024 × 2
Conv	4/5 × 1/1	4 × 1024 × 2
RSU	4/5 × 1/2	4 × 512 × 2
RSU (×3)	4/5 × 1/1	4 × 512 × 2
RSU	8/3 × 1/2	8 × 256 × 2
RSU (×3)	8/3 × 1/1	8 × 256 × 2
RSU	16/3 × 1/2	16 × 128 × 2
RSU (×3)	16/3 × 1/1	16 × 128 × 2
GAP	—	16

local features from the time series segments in the local feature learning block. These local features are then transferred in sequence to the Bi-LSTM layer to learn global features. In the attention block, AM assigns various attention scores, accentuating the influence of the more significant element of the feature map, and aids in making more correct determinations. Finally, we stack the dense and output layer in the identification block to perform the final identification. Each block of the proposed DNN model is demonstrated in detail below.

3.1. Local Feature Learning Block. In this block, we hope to obtain deep and invariant local features. The deep mining of features can be realized through the Deep Residual Network (DRN), which is composed of the stacked standard residual units (RUs). As shown in Figure 6(a), a standard RU consists of two batch normalization layers, two Rectified Linear Units (ReLUs) of activation layer and two convolutional layers. The input and output of RU is connected via shortcut, so as to solve the degradation problem in deep network. Noise interference will introduce variance to the features. In order to suppress the interference, the RUs in the DRN are replaced with the residual shrinkage units (RSUs), thus forming the residual shrinkage network (DRSN) [28]. Based on RU, soft thresholding is introduced as a nonlinear activation layer to eliminate noise-related features, which is calculated as,

$$y = \begin{cases} x - \kappa & x \geq \kappa \\ 0 & -\kappa \leq x < \kappa \\ x + \kappa & x < -\kappa \end{cases} \quad (10)$$

where x is the input feature map, y is the output feature map, and κ is the threshold, i.e., a positive parameter. Unlike ReLU setting the negative features to zero, soft thresholding

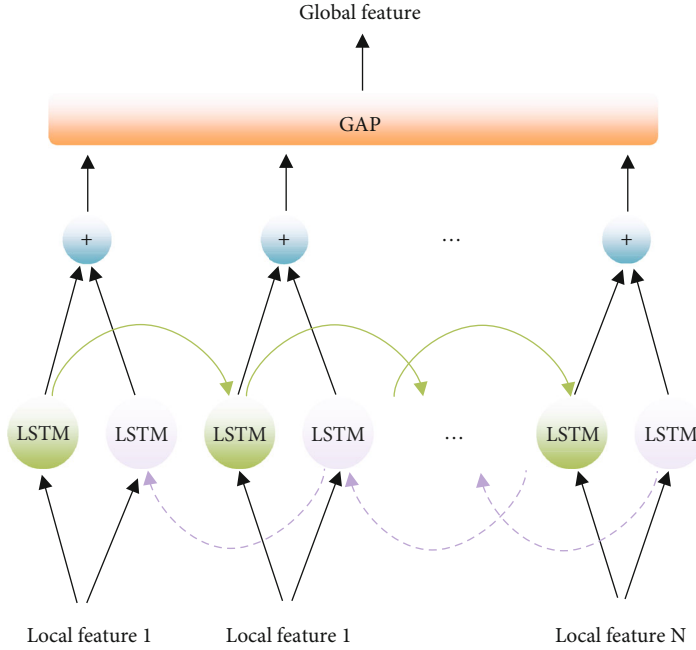


FIGURE 7: Schematic diagram of global feature extraction block.

sets the near-zero features to zeros to retain the useful negative features [28].

An additional branch of adaptive threshold calculation is added in RSU, as shown in the violet part of Figure 6(b). First, we use a global average pooling (GAP) layer to compress the absolute values of x into a one-dimensional (1D) vector, which is fed into a two-layer dense network to obtain the intermediate variable z [29]. z is then scaled to the range of (0, 1) using the sigmoid function, which is given by,

$$\alpha = \frac{1}{1 + e^{-z}}. \quad (11)$$

After that, the scaling parameter α is multiplied by the average value of $|x|$ to obtain the threshold. This is motivated by the fact that the threshold needs to be positive and cannot be too large. Thus, the threshold used in the RSU is given by,

$$\kappa = \alpha \cdot \text{average}_{i,j,c} |x_{i,j,c}|, \quad (12)$$

where i , j , and c are the indexes of width, height, and channel of x , respectively. In this way, the threshold is controlled in a reasonable range with respect to the input feature map.

In the local feature block, we stack several RSUs to achieve the following objectives: (a) extract deeper-level features, which aids in better representation of the input data; (b) gradually eliminate the noise-related features layer-by-layer. The number of RSU stackings is 12, the rationality of which is discussed later. Since the input of the Bi-LSTM layer must be a 1D array, the output of the local feature block is flattened into 1D data by the global average pooling (GAP) layer. The DRSN parameters are shown in Table 2.

3.2. Global Feature Learning Block. As the original inputs and the learned local feature maps represent the time course of electromagnetic activity of ACARS emitters, a RNN-based structure can be used to learn from the input along the time sequence in a parameter-sharing manner and memorize the context through their internal states [30]. An improved variant of RNN is LSTM, whose advantage is that it solves the problems of long-term memory and gradient disappearance in RNN while remaining computational cheap. In this paper, we adopt the structure of Bi-LSTM which has a forward and a backward LSTM layer. The forward one can process the past data information, whereas the reverse one can obtain the future data information. As shown in Figure 7, the local feature vectors are propagated into the Bi-LSTM layer in sequence, and the outputs are summed into a local-focused global feature vector, which encapsulates features from the context of the current step in both forward and backward directions. Both LSTM layers have 128 units, so the length of each local-focused global feature vector is 128. Finally, we use GAP to obtain a single output vector for the identification block.

3.3. Attention Block. Different kinds of information related to the emitter individual have different influence on the identification results. AM selectively focuses on some more influential information, so as to boost the expected information. The essence of AM is a mapping from a query to a sequence of key-value pairs, as depicted in Figure 8. The calculation of AM involves the following three stages.

At the first stage, the preliminary attention score is given by,

$$s_t = \tanh(W_h h_t + b_h), \quad (13)$$

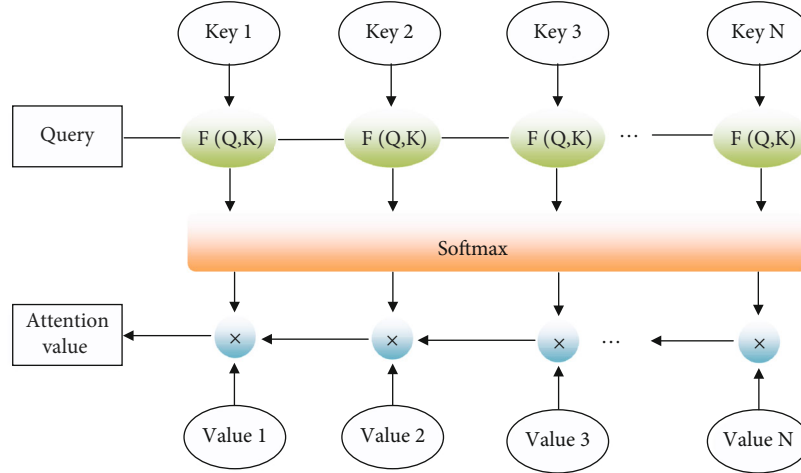


FIGURE 8: Schematic diagram of attention block.

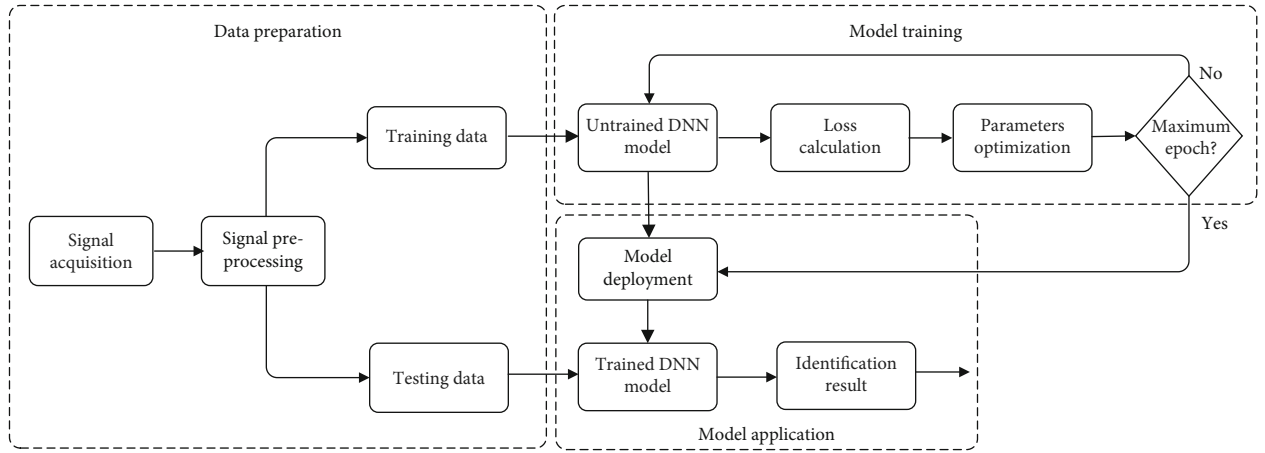


FIGURE 9: Flow chart of the proposed algorithm based on the DNN model.

where W_h , b_h are the weight and bias of AM, respectively, and h_t is the input vector. Then, the score is normalized using the softmax function,

$$a_t = \frac{\exp(s_t)}{\sum_t \exp(s_t)}, \quad (14)$$

regarding the score coefficient, the final attention score is obtained by weighted summation as shown,

$$c = \sum_t a_t q_t. \quad (15)$$

The above process shows that AM determines the most significant information by allocating higher scores to the feature map [31]. Thus, it has a positive optimization impact on our DNN model, and thus improves identification accuracy.

3.4. Identification Block. The dense layers map the distributed feature representation to the sample tag space via non-linear transformations. We have two dense layers in the final

block, and the first one has 128 neurons, while the second one has 7 neurons (corresponding to 7 categories). The first activation function uses ReLU to accelerate the back-propagation of gradients. The second activation function uses softmax to predict the probability distribution over the 7 categories.

4. Implementation of the Proposed SEI Algorithm

4.1. Overall Procedure of the Proposed Algorithm. In this paper, our proposed SEI algorithm contains three main steps: data preparation, model training, and model application, shown in Figure 9. In data preparation, the ACARS signals are collected in the out-field of the Xi 'an Xianyang airport emitted by seven civil aircraft, whose registration numbers are B1867, B30ER, B3229, B5180, B6469, B6695, and B9936. 1250 samples are collected from each aircraft. Subsequently, all samples are pre-processed in the same manner as section 2 described. The inputs are divided into training and testing data, which have a size of 1000 samples

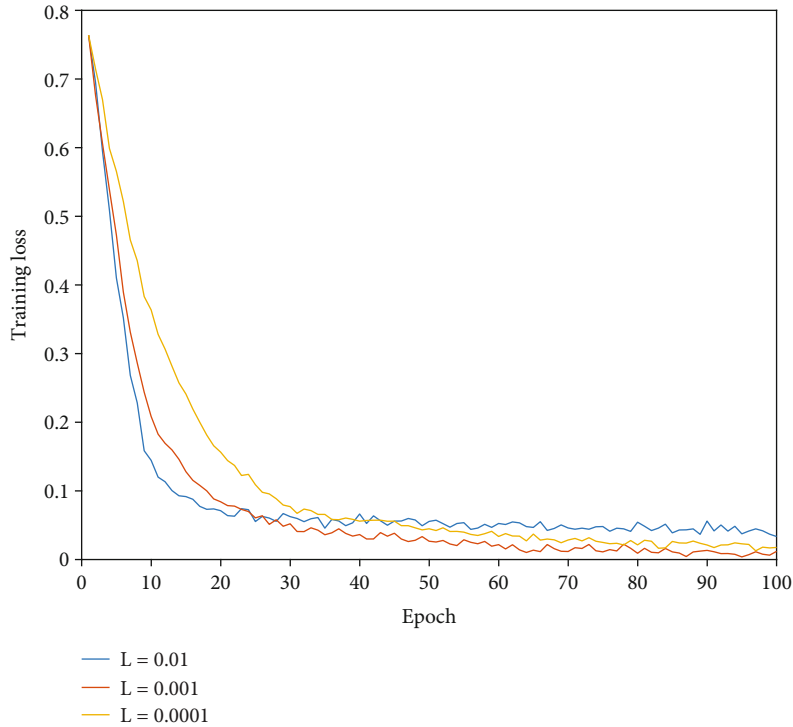


FIGURE 10: Training loss curve concerning different learning rate.

and 250 samples, respectively. We set the mini-batch size of the input data to 30. In model training, all aircraft that require certification must be registered offline. First, we construct the proposed DNN model with the determined initial parameters, and the training data from authorized entities are fed into the network to achieve forward propagation. We use cross-entropy as the loss function and the Adam algorithm as the optimizer. This training process will continue until the maximum epoch 200 is achieved. In model application, the online authentication system realizes the identification of the testing data through the trained DNN and finally outputs the probability distribution of the predicted emitter category. Through these three steps, SEI is implemented using the proposed algorithm.

The overall algorithm is run on a Linux machine with Nvidia K80 GPU, Intel Xeon W2155 CPU and 64GB RAM. The signal acquisition is implemented by TI ADC32RF82 RF-sampling wideband receiver and the preprocessing of the captured signals is performed on Matlab 2020b. The framework of the DNN model is constructed in Keras 2.0.8 with Tensorflow 1.7 backend.

4.2. Hyperparameters Selection of the DNN Model. The adjustments of model parameters are data-driven, but hyperparameters need to be selected manually. Here, we select several hyperparameters for balancing the performance and computational cost through a few trials.

- (a) **Learning Rate:** Learning rate controls the speed at which the loss function descends along the gradient. As shown in Figure 10, the curves of training loss concerning various learning rates exhibit significant

differences. We have the minimum training loss and the fastest convergence speed when we set the learning rate to 0.001. Thus, our DNN model selects the learning rate of 0.001 in the training process

- (b) **Hyperparameters of Local Features Learning Block:** As mentioned in Section 3, DRSN is designed to address the problem of performance degradation as the network deepens. Generally speaking, the deeper the residual network, the better the performance of feature learning. However, since an RSU occupies many computing resources, increasing the residual network depth will lead to the burden of computational complexity. Table 3 shows the identification accuracy and computational complexity concerning various numbers of RSUs. Note that the identification accuracy described here and below is for testing data. We can see that compared with image processing which requires dozens or even hundreds of layers of RSUs, feature learning for the signal can achieve good performance without stacking so many RSUs. The performance of feature learning hardly increases when the number of RUs is more than 12, but the computational complexity increases by 0.83 MFLOPs. Thus, we select 12 RSUs as the main body of the local feature learning block
- (c) **Hyperparameters of Global Features Learning Block:** We use dropout regularization in LSTM to prevent

TABLE 3: Identification accuracy and computational complexity concerning various hyperparameter settings for local feature learning.

Number of RSUs	Identification accuracy	Computational complexity (MFLOPs)
4	0.937	0.83
8	0.968	1.66
12	0.980	2.50
16	0.982	3.33
20	0.982	4.16

TABLE 4: Identification accuracy concerning various hyperparameter combination settings for global feature learning.

No. of LSTM units	Dropout rate	Recurrent dropout rate	Identification accuracy (%)
128	0.4	0.2	0.973
128	0.5	0.1	0.978
128	0.5	0.2	0.980
64	0.5	0.2	0.962
256	0.5	0.2	0.967

TABLE 5: Average time cost with raw and original ACARS signals as inputs of the DNN model.

Input ACARS signals	Raw	Original
Average training time (s)	48.2	14.7
Average testing time (s)	0.56	0.15

model over-fitting. Therefore, the hyperparameters closely related to model performance in the global feature learning block include LSTM unit number, dropout rate, and recurrent dropout rate. Table 4 shows the identification accuracy of various combinations of the above hyperparameters. We obtain the best performance when we have LSTM 128 units, 0.5 dropout rate, and 0.2 recurrent dropout rate

4.3. Complexity Analysis of the DNN Model. Next, we analyze the complexity of the proposed DNN model from time and space dimensions. The time complexity can be calculated by the Floating-Point Operations (FLOPs), and our model has 12.6MFLOPs; the space complexity can be calculated by the total weight parameters of the model, and our model has 4.7Mparas. Compared with the classical deep ResNet-50 network for image processing, which has 410 MFLOPs and 25.5Mparas, our model has low complexity. This is because we use the time series of signals as input, thus reducing the dimension of features and the complexity of the FC layer is greatly reduced due to fewer classification categories.

We then analyze the average time cost by dividing the total training and testing time by the dataset size. It can be seen from Table 5 that using original ACARS signals as inputs costs much more time than those using raw ACARS signals as inputs. The size of the original signal is several

times smaller than that of the raw signal due to the pre-processing. Thus, there are fewer segments in a sample that corresponds to the raw signal, which greatly reduces the time overhead.

5. Experiments and Discussions

5.1. Basic Identification Results. First, we conducted RF fingerprint registration for all 7 aircrafts, that is, training the collected samples through the network. The confusion matrix is shown in Figure 11(a), from which we can see that our algorithm is highly discriminative for the authorized aircraft. Next, we treated the aircraft with registration number B9936 as an unauthorized entity; namely, the corresponding samples are tested directly without training. The confusion matrix is shown in Figure 11(b), from which we can see that the probability distribution of the identification results of the unauthorized entity is scattered. Thus, we can set a threshold on the diagonal of the confusion matrix, such as 0.6, to distinguish between the authorized and unauthorized entities. We define the identification accuracy of the SEI algorithm as the average of the correct identification probability of all registered categories. In the case of Figure 11(a), the identification accuracy is 0.980, while in the case of Figure 11(b), the identification accuracy is 0.977. Generally speaking, the higher the proportion of registered entities in all categories, the higher the identification accuracy. Samples of unauthorized entities are untrained, leading to possible confusion with authorized entities. The identification accuracy involved below corresponds to the case where all seven aircraft have been registered.

Then, we evaluate the effect of valid signal interception in preprocessing. As Section 3 highlighted, we use the terms “raw ACARS signals” and “original ACARS signals” to represent the signals before and after the interception. As shown in Figure 12, the use of the original data shows better identification accuracy than that of the raw data. This is because we intercept the most representative part of the signal that best characterizes the individual information of each emitter in the preprocessing; on the other hand, the remainder of the raw data may contain differences in the message content, thus affecting the features learned by the network. According to the discussion in Section 4(c), the preprocessing makes our algorithm more competitive in identification accuracy and computational complexity.

5.2. Comparison with Other State-of-the-Art Algorithms. As our algorithm is based on the end-to-end learning, we select three DL algorithms which learn deep features directly from time series signals, namely CNN [20], LSTM [22, 32], and ResNet-50-1D [24]. Besides, we also use three manual feature-based machine learning algorithms for comparison, namely Bispectrum-CNN [17], HHT-DRN [18], and DCFT-CNN [19]. The identification results for various algorithms are shown in Figure 13. It can be observed that DL algorithms have better performance than ML algorithms, which indicates that the deep features are more reliable than the manual features. This may be due to the loss of information in the process of signal transformation. Moreover, manual feature extraction as an

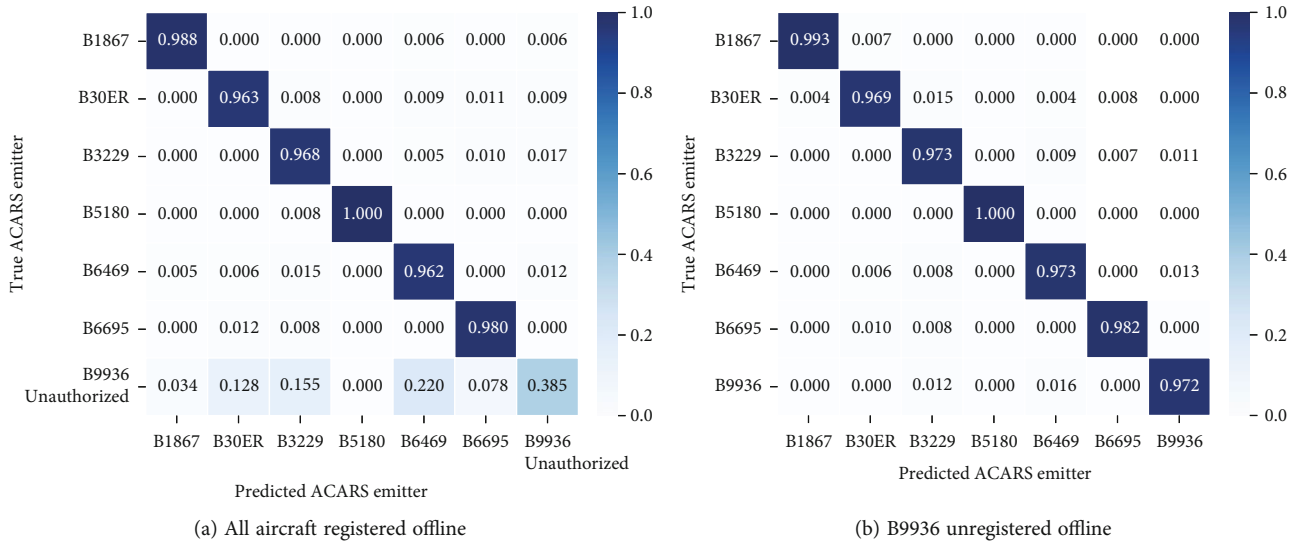


FIGURE 11: Confusion matrix for the SEI of ACARS authentication (The color depth of the color bar on the right of the picture represents the identification probability. The darker the color, the greater the identification probability).

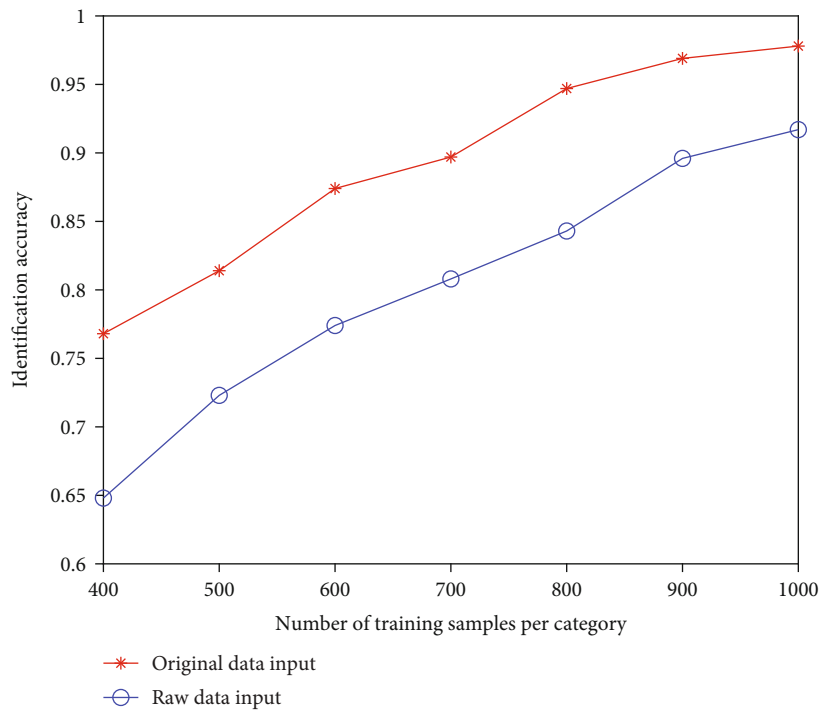


FIGURE 12: Identification results with original data and raw data as inputs.

additional operation will lead to a sharp rise in computational complexity, which is unfavorable to the practical application. We find that the Bispectrum-CNN performs the worst. For the local feature, the learning ability of the ResNet is stronger than that of the CNN, as the shortcut connections help the gradients propagate. The RNN-based structure, namely, LSTM in this paper, does well in characterizing temporal behavior but is poor at dealing with very long sequences. The length of the sequence is compressed after the local feature learning block, which

enhances the effectiveness and efficiency of RNN-based structure to learn global features. Among all DL algorithms, our DNN model based on DRSN and Bi-LSTM has the highest identification accuracy, which indicates that our hybrid deep architecture is competitive.

In addition, we also consider the role of AM in the proposed DNN model. The result shows that the absence of AM will lead to a decline in identification accuracy, which indicates that AM indeed enhances feature learning. Moreover, our pre-processing does not distinguish the

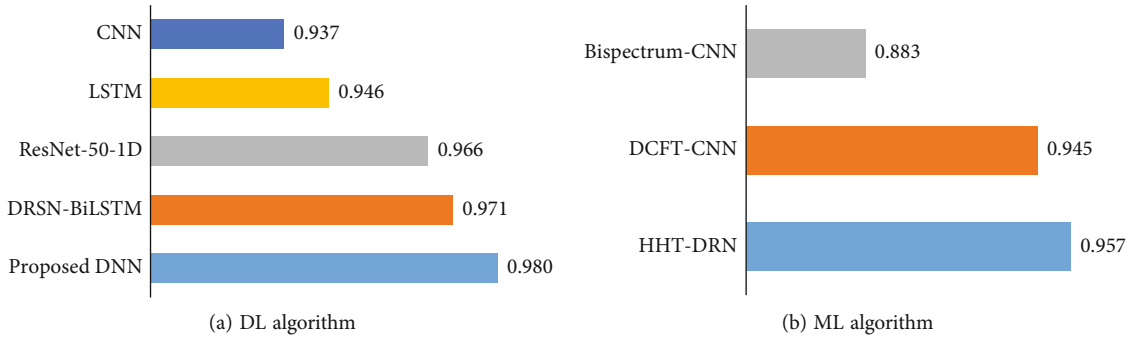


FIGURE 13: Identification accuracy of various state-of-the-art algorithms.

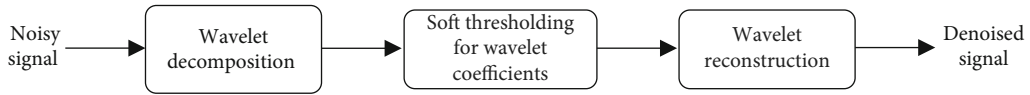


FIGURE 14: Wavelet denoising in the preprocessing.

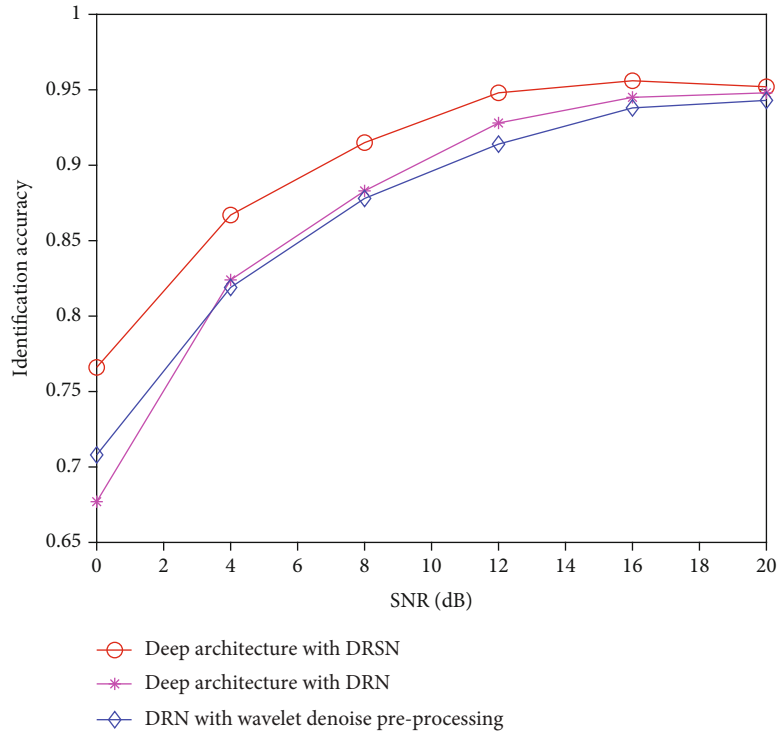


FIGURE 15: Performance on the noise sensitivity test under different deep architecture.

transient signal from the steady-state signal, AM aids to give different attention to the different signal periods to improve the accuracy of identification.

5.3. Noise Sensitivity Test. Next, we judge the identification performance of our algorithm under various signal qualities. Due to the limitation of acquisition conditions, we adopt the method of superimposing noise on the baseband signals. AWGN is injected with signal-to-noise ratio (SNR) ranging from 5 dB to 20 dB. It should be noted that the original sig-

nal is regarded as the unnoisy signal, so the actual SNRs are more miniature than the settings. For comparison, we use the DRN composed of standard RU and the wavelet noise preprocessing. The wavelet denoising process is depicted in Figure 14. The ‘db4’ wavelet is used to decompose the noise signal in the 4th order. We then use soft thresholding to filter the wavelet coefficients and perform inverse wavelet transformation to reconstruct the target signal [33].

Figure 15 compares the identification performance of ACARS signals for different deep architecture at various

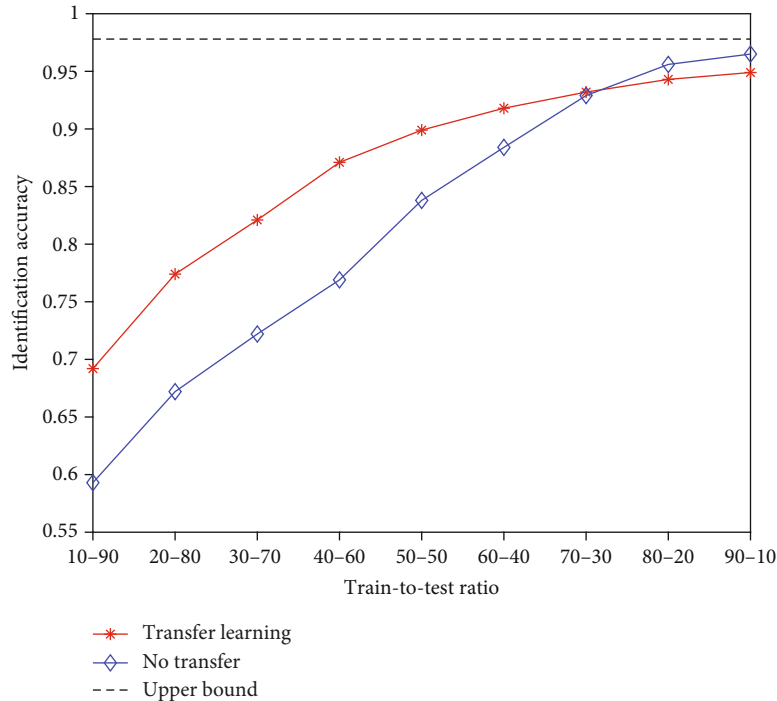


FIGURE 16: Performance on the robustness test for new data.

noise levels. It can be seen that compared with DRN, DRSN can enhance the resistance of the algorithm to noise disturbance, which is attributed to the stackings of several RSUs. Besides, the algorithm with denoise preprocessing performs with that without denoise preprocessing. The result suggests that denoising is not required in the preprocessing since part of the subtle characteristics representing individual differences may be removed simultaneously.

5.4. Robustness Test for New Data. At last, a new signal acquisition process was carried out a month after the first acquisition to test the robustness of our algorithm. In this experiment, we collected 1000 samples from each aircraft and divide training data and testing data according to a variable range. There are two strategies for the training process of the new data: transfer learning or no transfer. For transfer learning, the parameters are frozen in the local feature learning block, global feature learning block, and attention block, and the new data is used to re-train the dense layers in the identification block. This process can be called fine-tuning [34]. For the no transfer method, the new data is directly used to train the DNN model. In addition, the experiment uses the identification performance of a sufficient training set (containing 2000 samples) as the upper bound.

In Figure 16, we compare the two training strategies concerning the ratio of the sample number of train-to-test. It can be concluded that the performance will decrease with the shrinkage of the size of the training dataset, but using the transfer training strategy can significantly reduce the need for the number of training samples when the train-to-test ratio is less than “60-40”. Since feature extraction is the most time-consuming part of the training process, transfer learning can also accelerate training [34]. However, as the

number of training samples increases gradually, transfer learning even inhibits the optimization of the model, leading to the decline of the identification accuracy. Therefore, we recommend using 500-600 samples for transfer learning, which can conduct the training process of new data in a relatively short time and achieve an identification accuracy of 0.90 or so, which is close to the upper bound.

6. Conclusions

This paper proposed a novel SEI algorithm based on a hybrid DNN for ACARS authentication. The deep architecture combined DRSN, Bi-LSTM, and AM so that the hybrid network has a strong ability for feature learning and focusing. First, we preprocessed the captured signal to intercept the valid part according to the ACARS protocol. Then, the inputs were propagated into the hybrid DNN to obtain the probability distribution of the predicted emitter category. We introduced soft thresholding in DRSN to enhance the robustness against noise interference and adopted the transfer learning strategy to train new data in a low-cost manner. The hyperparameters of the model were determined through various trials. Finally, we performed a series of experiments under the condition of real-world signal acquisition. The results verify the rationality of the design and show considerable advantages of our algorithm in terms of accuracy and efficiency. The superior performance of the SEI algorithm shows its tremendous potential for practical application in ACARS authentication, thus providing a reliable guarantee for aviation information security. However, our signal acquisition process is all done on the ground, and the dataset is limited in size. In future work, we will

investigate the effect of transmission channels and consider applying large-scale datasets for validation.

Data Availability

The experimental data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest regarding this work.

Acknowledgments

This research was supported in part by the Natural Science Foundation of Shaanxi Province under Grant 2021JM-220, and in part by the Aeronautical Science Foundation of China under Grant ASFC-202055096001.

References

- [1] A. Roy, "Secure aircraft communications addressing and reporting system (ACARS)," in *20th DASC. 20th Digital Avionics Systems Conference (Cat. No.01CH37219)*, vol. 2, pp. 7A2/1–7A2/11, Daytona Beach, FL, USA, 2001.
- [2] M. Smith, M. Strohmeier, V. Lenders, and I. Martinovic, "On the security and privacy of ACARS," *Integrated Communications Navigation and Surveillance (ICNS)*, vol. 2016, pp. 1–27, 2016.
- [3] C. Breteau, S. Guigui, P. Berthier, and J. M. Fernandez, "On the security of aeronautical datalink communications: problems and solutions," in *2018 Integrated Communications, Navigation, Surveillance Conference (ICNS)*, Herndon, VA, USA, 2018.
- [4] C. Perner and C. Schmitt, "Security concept for unoccupied aerial systems," in *2020 AIAA/IEEE 39th Digital Avionics Systems Conference (DASC)*, pp. 1–8, San Antonio, TX, USA, 2020.
- [5] Aeronautical Radio, Incorporated, "ARINC 823 part 1 ACARS message security (AMS)," 2007, <https://www.sae.org/standards/content/arinc823p1/>.
- [6] H. Xu, X. Chen, F. Zhu, and P. Li, "A novel security authentication protocol based on physical unclonable function for RFID healthcare systems," *Wireless Communications and Mobile Computing*, vol. 2021, Article ID 8844178, 14 pages, 2021.
- [7] H. Zhou, L. Jiao, S. Zheng, L. Yang, W. Shen, and X. Yang, "Generative adversarial network-based electromagnetic signal classification: a semi-supervised learning framework," *China Communications*, vol. 17, no. 10, pp. 157–169, 2020.
- [8] N. Soltanieh, Y. Norouzi, Y. Yang, and N. C. Karmakar, "A review of radio frequency fingerprinting techniques," *IEEE Journal of Radio Frequency Identification*, vol. 4, no. 3, pp. 222–233, 2020.
- [9] L. Peng, A. Hu, J. Zhang, Y. Jiang, J. Yu, and Y. Yan, "Design of a hybrid RF fingerprint extraction and device classification scheme," *IEEE Internet of Things Journal*, vol. 6, no. 1, pp. 349–360, 2019.
- [10] L. J. Wong, W. C. Headley, and A. J. Michaels, "Specific emitter identification using convolutional neural network-based IQ imbalance estimators," *IEEE Access*, vol. 7, pp. 33544–33555, 2019.
- [11] C. Bertoncini, K. Rudd, B. Nousain, and M. Hinders, "Wavelet fingerprinting of radio-frequency identification (RFID) tags," *IEEE Transactions on Industrial Electronics*, vol. 59, no. 12, pp. 4843–4850, 2012.
- [12] G. Huang, Y. Yuan, X. Wang, and Z. Huang, "Specific emitter identification based on nonlinear dynamical characteristics," *Canadian Journal of Electrical and Computer Engineering*, vol. 39, no. 1, pp. 34–41, 2016.
- [13] D. Sun, Y. Li, Y. Xu, and J. Hu, "A novel method for specific emitter identification based on singular spectrum analysis," in *IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1–6, 2017.
- [14] A. Jagannath, J. Jagannath, and P. S. Kumar, "A comprehensive survey on radio frequency (RF) fingerprinting: traditional approaches, deep learning, and open challenges," 2022, <https://arxiv.org/abs/2201.00680>.
- [15] Z. Zhang, P. Cui, and W. Zhu, "Deep learning on graphs: a survey," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 1, pp. 249–270, 2022.
- [16] T. Young, D. Hazarika, S. Poria, and E. Cambria, "Recent trends in deep learning based natural language processing [review article]," *IEEE Computational Intelligence Magazine*, vol. 13, no. 3, pp. 55–75, 2018.
- [17] L. Ding, S. Wang, F. Wang, and W. Zhang, "Specific emitter identification via convolutional neural networks," *IEEE Communications Letters*, vol. 22, no. 12, pp. 2591–2594, 2018.
- [18] Y. Pan, S. Yang, H. Peng, T. Li, and W. Wang, "Specific emitter identification based on deep residual networks," *IEEE Access*, vol. 7, pp. 54425–54434, 2019.
- [19] L. Peng, J. Zhang, M. Liu, and A. Hu, "Deep learning based RF fingerprint identification using differential constellation trace figure," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 1, pp. 1091–1095, 2020.
- [20] K. Merchant, S. Revay, G. Stantchev, and B. Nousain, "Deep learning for RF device fingerprinting in cognitive communication networks," *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, no. 1, pp. 160–167, 2018.
- [21] Y. Wang, G. Gui, H. Gacanin, T. Ohtsuki, O. A. Dobre, and H. V. Poor, "An efficient specific emitter identification method based on complex-valued neural networks and network compression," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 8, pp. 2305–2317, 2021.
- [22] Q. Wu, C. Feres, D. Kuzmenko et al., "Deep learning based RF fingerprinting for device identification and wireless security," *Electronics Letters*, vol. 54, no. 24, pp. 1405–1407, 2018.
- [23] H. Zha, Q. Tian, and Y. Lin, "Real-world ADS-B signal recognition based on radio frequency fingerprinting," in *2020 IEEE 28th international conference on network protocols (ICNP)*, pp. 1–6, Madrid, Spain, 2020.
- [24] T. Jian, B. C. Rendon, E. Ojuba et al., "Deep learning for RF fingerprinting: a massive experimental study," *IEEE Internet of Things Magazine*, vol. 3, no. 1, pp. 50–57, 2020.
- [25] S. Chen, S. Zheng, L. Yang, and X. Yang, "Deep learning for large-scale real-world ACARS and ADS-B radio signal classification," *IEEE Access*, vol. 7, pp. 89256–89264, 2019.

- [26] Aeronautical Radio, Incorporated, “ARINC 618-8 air/ground character-oriented protocol specification,” 2016, <https://www.sae.org/standards/content/arinc618-8>.
- [27] Y. Qian, J. Qi, X. Kuai, G. Han, H. Sun, and S. Hong, “Specific emitter identification based on multi-level sparse representation in automatic identification system,” *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 2872–2884, 2021.
- [28] M. Zhao, S. Zhong, X. Fu, B. Tang, and M. Pecht, “Deep residual shrinkage networks for fault diagnosis,” *IEEE Transactions on Industrial Informatics*, vol. 16, no. 7, pp. 4681–4690, 2020.
- [29] Z. Wang, W. Yan, and T. Oates, “Time series classification from scratch with deep neural networks: a strong baseline,” in *2017 International Joint Conference on Neural Networks (IJCNN)*, pp. 1578–1585, Anchorage, AK, USA, 2017.
- [30] R. He, Y. Liu, K. Wang et al., “Automatic cardiac arrhythmia classification using combination of deep residual network and bidirectional LSTM,” *IEEE Access*, vol. 7, pp. 102119–102135, 2019.
- [31] P. Kavianpour, M. Kavianpour, E. Jahani, and A. Ramezani, “A CNN-BiLSTM model with attention mechanism for earthquake prediction,” 2021, <https://arxiv.org/abs/2112.13444>.
- [32] J. Ma, H. Liu, C. Peng, and T. Qiu, “Unauthorized broadcasting identification: a deep LSTM recurrent learning approach,” *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 9, pp. 5981–5983, 2020.
- [33] D. L. Donoho, “De-noising by soft-thresholding,” *IEEE Transactions on Information Theory*, vol. 41, no. 3, pp. 613–627, 1995.
- [34] F. Zhuang, Z. Qi, K. Duan et al., “A comprehensive survey on transfer learning,” *Proceedings of the IEEE*, vol. 109, no. 1, pp. 43–76, 2021.