Deep Learning Based Identification Method for Signal-Level Wireless Protocol

JIE REN, SHUAI LI, YIHAO SONG, AND MING LI
Institute of Telecommunication and Navigation Satellites, China Academy of Space Technology, Beijing 100094, China
Innovation Center of Satellite Communication System, CNSA, Beijing 100094, China
Corresponding author: Jie Ren (rj278007@126.com)

ABSTRACT Electromagnetic spectrum surveillance is the basis of space environment monitoring and management, while protocol identification is one of the most important methods. Protocol identification has been applied in many fields, such as target recognition, anomaly detection, security management and information countermeasure. In the past decade, a lot of deep learning based protocol identification methods have been proposed, based on the data-level protocol. However, the data needs to be demodulated and decoded, depending on the prior information of the target system. In this paper, we focus on signal-level protocol, and introduce the deep learning based identification method. We first develop a multi-dimensional convolutional neural network based method for feature extraction of signal-level protocol. Then, a spatial pooling convolutional neural network is introduced for variable length protocol identification. At last, an unsupervised training network is proposed for feature extraction of unlabelled signal-level protocol. The proposed methods are evaluated by using the practical wireless protocols, and the effectiveness is verified by simulation results.

INDEX TERMS Protocol identification, signal-level protocol, convolutional neural network, spatial pyramid pooling, correlation coefficient loss function.

I. INTRODUCTION
Electromagnetic spectrum surveillance is important for electronic warfare, which is used to discover electromagnetic signals and identify target parameters. Protocol is the foundation of wireless communication system, and the identification of protocol type is an important research in electromagnetic spectrum surveillance [1]. Protocol defines the transmission data format, data content and information timing of the wireless system, that is closely related to its data structure, transmission system, data type and application type [2]. Due to its importance, the protocol identification based electromagnetic spectrum surveillance technology has important research and application value.

With the rapid development of wireless communication technology, the electromagnetic environment is becoming more complex, which is mainly reflected in the rapid increase of wireless communication system types, the fast growth in the number of wireless communication terminals, and the expansion of wireless communication data. For the system type, a complex space environment with multiple wireless systems has been formed, which includes 5G, GNSS, satellite communication systems, etc [3]. For the terminal, the number of global IP network equipment in 2018 is 18.4 billion, and the number is expected to 29.3 billion in 2023 [4]. For wireless communication data, the mobile traffic in China is reported to exceed 221 billion GB in 2021 [5]. Due to the trend of multi-system, multi-object and big data in wireless communication environment, the wireless protocol scenario is facing the trend of protocol system diversification, protocol big data and protocol non-cooperation. Therefore, it is of great significance to improve the generality and applicability of protocol identification method to meet the requirements for accurate protocol identification in complex wireless protocol scenarios.

Due to the particularity of the research object, wireless protocol identification is different from traditional protocol identification. First, traditional protocol identification...
methods are usually based on application layer protocols of wired communication system, while wireless protocol identification focuses on the application layer, link layer or physical layer protocols. In [6], the authors studied the protocol identification method for spatial link layer protocol. Second, the traditional protocol identification methods are based on the data-level protocol [7]. However, the data-level protocol needs to be demodulated and decoded after receiving the wireless signal, which is difficult when lacking of prior information [8]. Furthermore, feature extraction for data-level protocol is rather resource-expensive and does not work on encrypted system, which has been widely used in wireless communication system. In that case, the identification of signal-level protocol is an important characteristic of wireless protocol identification. Some researches for signal-level protocols are applied for the identification of different wireless systems. However, the protocols within the same wireless system exhibit a high degree of consistency in signal-level features because of the same physical layer parameters are used. Therefore, we mainly focus on the identification method for signal-level protocol.

A. RELATED WORK
Protocol identification refers to distinguish a specific type of protocol from other types by the various logical relationships and protocol features. Generally, protocol identification consists of two parts: feature extraction and protocol identification. For feature extraction, the current research focuses on the data-level protocol features, while the features can be divided into fixed features, payload features, behavior features and statistical features. Fixed features are used for protocols with fixed port number, such as port 80 for HTTP, port 25 for SMTP, port 20 for FTP and port 110 for Pop3 [9]. Payload features refer to the specific character strings or binary sequences contained in a data packet or data stream of a specific protocol [10]. Behavior features refer to the observable data behaviors or user behaviors in communication system [11]. Statistical features are extracted by statistical analysis from protocol data stream [12].

According to the type of features, the traditional protocol identification methods can be divided into four types. Fixed features based method uses fixed features for protocol identification, such as port number [9]. However, most protocols are not registered in Internet Assigned Numbers Authority (IANA) any more. In addition, the use of dynamic ports and port concealment technologies makes it difficult to identify these protocols effectively. Payload features based method is based on the payload feature library of data packets or data streams, such as Deep Packet Inspection (DPI) and Deep Flow Inspection (DFI) [13]. Behavior features based method is implemented by behavior analysis, such as communication data behaviors or user behaviors. In [14], the authors extracted the application behaviors for classifying P2P traffic, such as external IP addresses, the number of sent and received data streams, and the number of data packets, then a decision tree was used to realize P2P protocol identification. Statistical features based method is based on the statistical features of the protocol data, and machine learning is used to realize the protocol identification [15]. In [16], the bayesian neural network was used for protocol classification. The authors in [17] proposed the hidden Markov based method to identify the encrypted protocol by using the statistical features of TCP packet size, arrival time, direction of arrival, etc. In [18], the authors proposed a support vector machine (SVM) based method for protocol identification. The above methods have been verified in some specific scenarios.

B. MAIN CONTRIBUTIONS
In the past decades, protocol identification has been widely used in the fields of target recognition, anomaly detection, security management and information countermeasure [19]. The performance of identification is mainly determined by feature extraction. According to the properties of protocol, the features can be divided into two types, data-level features and signal-level features. The data-level refers to the bitstream protocol after demodulation, decoding and despreading, while the signal-level refers to the sampled data without demodulation, decoding and despreading. For data-level protocol, the features are extracted by data statistical analysis or data packet analysis, such as bitstream length, number of data packets, average packet size, stream length difference, packet size difference, data composition, etc. In [20], the authors analyzed the properties of the network protocol streams and summarized 248 optional bitstream features. For signal-level protocol, the features used for identification are usually the parameters of the system, such as frequency, modulation type, channel coding type, frame format, signal high order statistics, etc. These features are commonly applied for wireless protocol identification of different wireless systems. Besides, most of the traditional protocol identification methods are based on supervised learning.

Generally, the traditional protocol identification methods rely on prior information and expert knowledge [21]. However, due to the trend of multi-system, multi-target and big data in wireless environment, the existing protocol identification technology, especially the feature extraction, is facing severe challenges [22]. First, the traditional feature extraction methods are only suitable for some certain systems. When the object system changes, the feature extraction needs to be redesigned based on prior information of new system. Second, due to the reliance on prior information and the wireless protocol big data, feature extraction and feature library construction will consume a lot resources. Besides, The diversity of feature attributes makes it difficult to determine the optimal feature selection. Third, the existing protocol identification methods are suitable for data-level protocols, but data acquisition requires demodulation and decoding after receiving wireless signals. Especially for encrypted system, the traditional protocol identification methods exhibit the unsatisfied performance since the encrypted traffic is no longer in plain text and hard to extract effective data-level features. The current research on signal-level protocol identification is used
for multiple wireless systems identification. Last, protocol big data makes it difficult for data labelling, which leads to the ineffectiveness of supervised learning based protocol identification method. Thus, the traditional protocol identification is always restricted for its prior reliance, resources consumption and feasibility limitation.

In recent years, the development of deep learning promotes the innovations in many fields, such as image processing and natural language processing [23]. As a representation learning method, deep learning is suitable for machine to automatically discover the representations from the structured data [24]. Since requiring very few prior, deep learning can be used for automatic feature extraction and object identification [25]. Inspired by deep learning, we aim to provide a novel deep learning based method for accurate protocol identification. Specifically, the proposed method in this paper is suitable for signal-level protocol. Since signal-level data is used, the method proposed in this paper can be extended to encrypted signals.

In brief, the main contributions of this paper are summarized as follows:

1) We investigate the signal-level protocol identification problem and propose a multi-dimensional convolutional neural network (4LCNN) based method for feature extraction of signal-level protocol.
2) We develop a spatial pooling convolutional neural network (CNN-SP) for variable length protocol identification.
3) Considering the unlabelled dataset scenario, we propose an unsupervised training based convolutional neural network (CAEs) for feature extraction, by maximizing the signal correlation between original protocol signal and reconstructed signal.

C. ORGANIZATION OF THIS PAPER

The rest of this paper is organized as follows. In Section II, a multi-dimensional CNN method is proposed for signal-level protocol identification. After that, a CNN-SP is presented for variable length protocol identification in Section III. Then, an unsupervised training based CAEs is proposed for protocol identification. Simulation results are presented in Section V. Finally, conclusions are drawn in Section VI.

II. MULTI-DIMENSIONAL CNN FOR SIGNAL-LEVEL PROTOCOL IDENTIFICATION

Feature extraction is the key to protocol identification, and the effectiveness of selected features determines the performance of protocol identification. The traditional data-level feature extraction is based on the data-level protocol, while the current research on signal-level protocol feature extraction is used to identify the protocols of different wireless systems. For different systems, protocols can be realized by the features in time domain, frequency domain, or power domain. However, due to the consistency of these features in the certain system, it is difficult to identify the protocols within the same system by these features. Therefore, we consider a method to achieve efficient protocol feature extraction without demodulation and decoding, that is, the signal-level protocol feature extraction.

A. STRUCTURE

Considering the data structure in wireless communication system, bitstream is used to form radio frame in a certain format, which is then used to generate transmission signals. For data-level, the protocol is formed in a certain format, such as pilot, transport format combination indicator (TFCI), feedback information (FBI), transmit power control (TPC), etc [26]. In addition, there is a certain difference on protocol structure of different wireless communication systems. The protocols of the same communication system have similarity in structure, but the protocols of different applications have distinguishable structural features. For signal-level, the protocol is first generated by structural bitstream data, then the signal-level data is generated by spreading, modulation, etc. Similar to the hierarchical structure of image, in which high-level features are the representations of low-level features, the protocol can be regarded as a coalition of a series of structured patterns, both in data-level and signal-level. Considering the advantages for structured features representation, we introduce the deep learning model, convolutional neural network, for the task of feature extraction. In this paper, we focus on signal-level protocol.

Convolutional neural network is a deep learning network for processing data with known topology, such as time series data with one-dimensional grid and image data with two-dimensional grid [23]. Compared to other classification methods, it uses convolution instead of general matrix multiplication with little preprocessing. Due to low reliance on prior information and expert knowledge, CNN is competent for automatic feature extraction, which has been proved in practical applications of image and video [27].

In this section, we focus on the protocol identification method for signal-level protocol. Since high-order modulation is used in modern wireless communication system, such as QPSK and QAM, the received signal can be represented as in-phase and quadrature branches. In that case, the signal-level protocol can be regarded as a two-dimensional data. Based on the signal-level protocol, this paper develops a multi-dimensional convolutional neural network for signal-level protocol identification. The structure of the model is shown in figure 1. It consists of four convolutional layers, four pooling layers, and two fully connected layers. For convolutional layers, different kernels are used to detect and extract different kinds of local structural features in the feature maps of previous layer, and weight sharing is used to make the feature extraction insensitive to positions. It means that we only need to save a few parameters, which reduces the memory requirement of the network. Furthermore, convolution simplifies the operation of forward propagation, and leads to the computational complexity reduction and efficiency improvement. The pooling layer is used to make the
representation approximately invariant to small translations of the input and reduce the size of the output [28]. The fully connected layer is used to construct the feature vector, which is used for protocol identification.

B. ALGORITHM

In order to demonstrate our method, the used notations and their explanations are shown in table 1.

Assume that there is a protocol sample \((x', y') \in D_S\), where \(x'\) is the signal-level protocol data with dimension \([2, L]\) and \(y'\) is the type of protocol in one-hot format. \(L\) is the length of the protocol data. First, we feed forward to get the prediction result of input \(x'\). For the first convolutional layer, we have that

\[
[u_i^c]^{(1)} = f_{\text{conv-2D}}([x']^c, W_c^{(1)}) + b_c^{(1)},
\]
\[
[a_i^c]^{(1)} = f_{\text{activation}}([u_i^c])^c, m = 1, \ldots, K^{(1)},
\]

where \(W_c^{(1)}\) is the parameter of the \(c\)-th convolution kernel in the first convolutional layer. Since the input is described as the I/Q branches data, two-dimensional convolution is used for the first convolutional layer. That is, \(W_c^{(1)}\) is a two-dimensional convolution kernel. \(b_c^{(1)}\) is the bias of feature map \(c\), and \(K^{(1)}\) is the number of convolution kernels in the first convolutional layer. \(f_{\text{conv-2D}}\) represents the two-dimensional convolution operation. The output of convolution is \([u_i^c]^{(1)}, f_{\text{activation}}\) represents the ReLU activation operation, and the output is \([a_i^c]^{(1)}\). Due to the dimension of input \(x'\) and convolution kernel \(W_c^{(1)}\), the output \([a_i^c]^{(1)}\) is a one-dimensional feature map.

After the convolutional layer, it is followed by a pooling layer, with the purpose of reducing the size of output feature map. Max pooling is used to select the maximum value from the cluster in the prior layer with the same size of pooling window, and expressed as

\[
[a_i^c]^{(2)} = f_{\text{maxpooling}}([a_i^c]^{(1)} + \beta_c^{(2)} + b_c^{(2)}),
\]

where \(\beta_c^{(2)}\) and \(b_c^{(2)}\) are the weight and bias of the pooling layer. \([a_i^c]^{(2)}\) is the output, and the size is smaller than that of the input feature map \([a_i^c]^{(1)}\).

In order to efficiently extract protocol features with appropriate dimension and reduce prediction error, the multi-layer convolution framework is used to extract signal-level protocol features. Thus, in the second convolutional layer, we have that

\[
[u_i^d]^{(3)} = f_{\text{conv-td}}([a_i^c]^{(2)}, W_d^{(3)}) + b_d^{(3)}
\]
\[
[a_i^d]^{(3)} = f_{\text{activation}}([u_i^d])^d, m = 1, \ldots, K^{(2)},
\]

where \(W_d^{(3)}\) is the weight of the \(d\)-th convolution kernel, and \(b_d^{(3)}\) is the corresponding bias. Since the input of second convolutional layer, denoted as \([a_i^c]^{(2)}\), is a one-dimensional feature map, one-dimensional convolution is used in the second and the following convolutional layers. That is, \(W_d^{(3)}\) is a one-dimensional convolution kernel. \(K^{(2)}\) is the number of kernels in the second convolutional layer. Similarly, the second pooling layer can be expressed as:

\[
[a_i^d]^{(4)} = f_{\text{pooling}}([a_i^d]^{(3)} + \beta_d^{(4)} + b_d^{(4)}),
\]

where \(\beta_d^{(4)}\) and \(b_d^{(4)}\) are the weight and bias of the pooling layer. \([a_i^d]^{(4)}\) is the output. Similarly, the output of the third and fourth convolutional layers and the pooling layers are denoted as \([a_i^c]^{(5)}, [a_i^c]^{(6)}, [a_i^c]^{(7)}, [a_i^c]^{(8)}\) respectively, where the subscripts \(e\) and \(f\) represent the feature map of the corresponding layers.

After that, two fully connected layers are used to construct the protocol feature vector. In the first fully connected layer, there is

\[
[u_i^o]^{(9)} = f([a_i^f]^{(8)}, W^{(9)}) + b^{(9)}
\]
\[
[a_i^o]^{(9)} = f_{\text{activation}}([u_i^o])^{(9)}
\]

where \(W^{(9)}\) is the weight to connect the nodes in the \(l = 9\) layer with all the nodes of feature maps in the \(l = 8\) layer. \(b^{(9)}\) is the bias and \([a_i^o]^{(9)}\) is the output of the fully connected layer. Similarly, the output of the second fully connected layer is expressed as \([a_i^o]^{(10)}\). Finally, the output of the classifier is

\[
o_i^* = g([a_i^o]^{(10)} \cdot W + b),
\]

where \(o_i^*\) is the prediction result of input \(x'\). \(g(\cdot)\) is the softmax classification function. Therefore, the feature vector of protocol \(x'\) is denoted as \(h' = [a_i^o]^{(10)}\).

After feedforward, cross entropy loss function is applied to evaluate the deviation between the real type and prediction, given by

\[
J_{\text{loss}} = - \sum_{i=1}^{k} y_i^* \log o_i^*.
\]
When back propagation to train the network, we compute the derivatives with respect to the weights and biases of each layer. For the fully connected layer, there is

$$
\delta^{(l)} = \nabla^{(l+1)^T} \cdot \delta^{(l+1)} \circ f_{\text{activation}}'([a_i^{(l)}])
$$

$$
\frac{\partial J}{\partial b^{(l)}} = \delta^{(l)}
$$

$$
\frac{\partial J}{\partial W^{(l)}} = [a_i^{(l-1)}(\delta^{(l)})]^T,
$$

where $\delta$ represents the “errors” when propagating backwards through the network. $W^{(l+1)}$ is the weight of the $(l+1)$ layer, and $[a_i^{(l)}]$ is the input before activation. $f'($) is the derivative of activation function. $\circ$ represents the element-wise multiplication of two tensors with same size.

For the pooling layer, if followed by fully connected layer, then the “errors” can be calculated as the way for fully connected layer. Otherwise, for the “errors” of layer $l=2,4,6$ that followed by convolutional layer, there is

$$
\delta_n^{(l)} = f''_{\text{activation}}([a_i^{(l)}]) \circ f_{\text{conv},1D}^{(l)}(\delta_n^{(l+1)}, W_n^{(l+1)})
$$

$$
\frac{\partial J}{\partial m(l)} = \sum_k (\delta_m^{(l)})_k
$$

$$
\frac{\partial J}{\partial m(l)} = \sum_k (\delta_m^{(l)} \circ f_{\text{down}}([a_i^{(l-1)}]))
$$

where $f_{\text{down}}$ represents a sub-sampling function.

For convolutional layer, there is

$$
\delta_n^{(l)} = f_{\text{conv},1D}^{(l)}(\delta_n^{(l+1)} \circ f_{\text{up}}(\delta_m^{(l+1)})))
$$

$$
\frac{\partial J}{\partial m(l)} = \sum_k (\delta_m^{(l)}_k)
$$

$$
\frac{\partial J}{\partial m(l)} = \sum_k (\delta_m^{(l)} \circ (\delta^{(l-1)}_k))
$$

where $f_{\text{up}}$ is the up-sampling operation. $(\delta_m^{(l)}_k)$ is the “errors” of the $m$-th feature map in layer $l$, while $k$ represents the element. $(\delta^{(l-1)}_k)$ is the data patch that is multiplied by $W_m^{(l)}$ during convolution in order to compute the element at $k$ in the output feature map.

### III. VARIABLE LENGTH PROTOCOL IDENTIFICATION

In traditional protocol identification methods, data length is generally used as one of the features for identification. However, since most protocols have variable lengths and overlap with different protocol length ranges, the length characteristic is indistinguishable in protocol identification. Because of the uncertainty in data collection and segmentation, the protocol to be identified is usually in variable length. Therefore, we develop a new structure CNN for variable length protocol identification.

#### A. STRUCTURE

Protocol identification is divided into two parts: feature extraction and protocol identification. The variable length protocol only affects the construction of feature extraction. Traditionally, the variable length protocol is converted to fixed length feature vector when using hand-design identification methods, or converted to fixed length data by truncating or padding when using deep learning based identification methods. In the previous section, the proposed 4LCNN consists of convolutional layers, pooling layers, and fully connected layers. According to the structure of 4LCNN, convolutional and pooling layers have no restriction on the dimension of the input data and can be used to extract features from variable length protocol. However, the fully connected layer needs to determine the network structure in advance, including the input and output dimension. In 4LCNN, max pooling is used to reduce the size of feature map, expressed as

$$
f_{\text{max pooling}} = \max(p_i), i = s, \ldots, s+k-1
$$

where $s$ is the starting position of the pooling window and $k$ is the size of pooling window. $p_i$ is the input feature map of pooling layer. Suppose that $x$ is the input signal-level protocol data with length $L$. After four convolutional and pooling layers, the dimension of the output is $[1, (L/\prod_{i=1}^{4} s^{(i)})@K^{(4)}]$, where $s^{(i)}$ is the kernel length of the $i$-th pooling layer and $K^{(4)}$ is the number of output feature maps. Since the input dimension of the fully connected layer is fixed, that is, $[1, (L/\prod_{i=1}^{4} s^{(i)})@K^{(4)}]$ is a fixed value. So the input protocol length $L$ is a fixed value. Due to the limitation of the fully connected layer on the dimension of the input data, the 4LCNN based method can be only used for the fixed length protocol identification.

In order to solve the problem of variable length protocol identification, this paper introduces the idea of spatial pooling for 4LCNN, which is used to convert the variable length input to a fixed length output [29]. The structure of CNN-SP based protocol identification method is shown in figure 2. Different from 4LCNN based method, CNN-SP consists of four convolutional layers, three max pooling layers, one spatial pooling layer, and two fully connected layers.

#### B. ALGORITHM

As shown in figure 2, the input of the CNN-SP model is denoted as $x$ with dimension $[2, L]$. After three convolutional
and pooling layers, the dimension of output feature map is $[1, (L / \prod_{l=1}^{3} s'_p)] @ K^{(3)}$, where $K^{(3)}$ is the number of convolution kernels in the third convolutional layer. For the fourth convolutional layer, the dimension of the output feature map is $[1, (L / \prod_{l=1}^{3} s'_p)] @ K^{(4)}$, where $K^{(4)}$ is the number of convolution kernels in the fourth convolutional layer. Different from the 4LCNN based method, spatial pooling layer is used to down-sample feature maps with different kernel sizes and generate the output with a fixed size. The kernel size is dynamically adjusted according to the dimension of the input data. In order to improve the robustness of feature extraction and the performance of protocol identification, multiple spatial sizes are generally used. In CNN-SP model, the dimension of each input feature map of the spatial pooling layer is $[1, (L / \prod_{l=1}^{3} s'_p)]$. Then, three spatial pooling kernels are used for down-sampling, while the output sizes are $[1, 8]$, $[1, 4]$, and $[1, 1]$ respectively. Thus, the variable length input with size $[1, (L / \prod_{l=1}^{3} s'_p)]$ can be converted to the fixed length output with size $[1, 13]$. Therefore, for the spatial pooling, we have

$$s_{sp} = \left[ \left( L / \prod_{l=1}^{3} s'_p \right) / n \right] = \text{ceil}(L / \prod_{l=1}^{3} s'_p) / n$$

$$k_{sp} = \left[ \left( L / \prod_{l=1}^{3} s'_p \right) / n \right] = \text{ceil}(L / \prod_{l=1}^{3} s'_p) / n$$

$$p_{sp} = s_{sp} * n - (L / \prod_{l=1}^{3} s'_p),$$

(12)

where $s_{sp}$ is the length of spatial pooling kernel for the output with dimension $[1, n]$, and $\left[-\right]$ is the ceil function. $k_{sp}$ is the sliding step of spatial pooling kernel. $p_{sp}$ is the data that needs to be filled at the end of the input data. By the proposed CNN-SP method, the input protocol with variable length can be converted to the feature map with fixed size $[1, 13] @ K^{(4)}$, and that can be used as the input of fully connected layer and applied for variable length protocol identification.

**IV. UNSUPERVISED CAE FOR PROTOCOL IDENTIFICATION**

The 4LCNN is a supervised training based method, which relies on the large scale labelled protocol dataset. However, it is difficult to collect sufficient and quality labelled dataset, because of the protocol big data and protocol non-cooperation. In this section, we combine the CNN with the autoencoder, and propose an unsupervised training based feature extraction method for unlabelled dataset.

**A. STRUCTURE**

Traditionally, the CNN based feature extraction is trained by the large scale labelled protocol dataset. The AE based feature extraction is trained by the unlabelled protocol dataset. The former has better feature representation performance, while the latter does not rely on the prior information of protocol labels. Combined the advantages of these two models, we develop an unsupervised CAEs model for unsupervised feature extraction. The structure of the CAEs is shown in figure 3.

The CAE unit consists of two parts, the first is feature extraction, which is constructed by convolutional and pooling layers to extract features from the input. The second part is signal reconstruction, which is constructed by decoder layers to reconstruct the input by the extracted features. The first CAE unit is trained by back propagation and the extracted features are used as the input of second CAE. Four CAEs are stacked and the feature extraction part is used to extract features from the input.

**B. ALGORITHM**

As shown in figure 3, the input of the CAEs is denoted as $x^i$. For the first feature extraction part, the input $x^i$ is processed by convolution and pooling, and the output is denoted as $[a^i]^{(2)}$. Successively, by four feature extraction layers, the extracted feature is expressed as $[a^i]^{(3)}$. Then, by four reconstruction layers, the features are decoded to reconstruct the input signal, denoted as $\hat{x}^i$. The target of CAEs is to make the reconstructed signal $\hat{x}^i$ as similar as possible to the original input signal $x^i$. Therefore, the correlation coefficient loss function is constructed for the training, which is expressed as follows

$$f_{loss} = - \frac{\mathbb{E}[(x^i - \mathbb{E}(x^i))(\hat{x}^i - \mathbb{E}(\hat{x}^i))]}{\mathbb{E}[x^i - \mathbb{E}(x^i)]^2 \mathbb{E}[\hat{x}^i - \mathbb{E}(\hat{x}^i)]^2}. \tag{13}$$

The target is to minimize the loss function by back propagation.

**V. SIMULATION RESULTS**

**A. DATASET**

The rapid development of deep learning technology is benefited from big data. Furthermore, the integration of deep learning and wireless communication technology also needs the support of big data. However, considering standard, copyright and security issue, there are few open source datasets for
wireless communication data, and some open source datasets are mainly generated by simulation. For example, O’Shea generated 24 kinds of wireless signals with different modulation types [30]. In addition, the current protocol datasets are mainly based on wired network protocol data streams, such as CAIDA dataset, UNIBS dataset, WIDE dataset, etc [31]. In our experiment, we collected the protocol data by the Software Defined Radio (SDR) platform USRP, which consists of the USRP X310 motherboard, UBX160 daughterboard and GNURadio software platform. Then, the signal-level protocol dataset was constructed. The data collection platform is shown in figure 4.

Based on the data collection platform, we first collected the protocol data based on the 802.11a standard system. Then, the data was labelled by wireshark and 7 types of protocol were selected to construct the data-level protocol dataset. Based on the physical layer parameters of 802.11a, we constructed an OFDM-based signal transmitter and converted the above data-level protocol dataset to the corresponding signal-level protocol dataset. The transmitter consists of coding, interleaving, modulation, IFFT, etc [32]. For coding and modulation, a $(133, 171)_8$ convolutional code with rate $\frac{1}{2}$ and QPSK modulation were applied, respectively. In order to improve the efficiency of data processing, an efficient data format provided by Google, TFRecord, was constructed for the training and testing dataset [33]. The dataset is shown in table 2. The program was developed on Tensorflow, and the training was accelerated by GPU.

### B. RESULTS AND ANALYSIS

#### 1) EVALUATION OF 4LCNN BASED SIGNAL-LEVEL PROTOCOL IDENTIFICATION METHOD

In this simulation, the proposed 4LCNN based signal-level protocol identification method is evaluated. It consists of four convolutional layers, four pooling layers, and two fully connected layers. The performance of the proposed 4LCNN is simulated on the signal-level protocol dataset.

![FIGURE 4. Data collection platform, which consists of the USRP X310 motherboard, UBX160 daughterboard and GNURadio software platform.](image)

![FIGURE 5. The performance of 4LCNN with different feature dimensions.](image)

The proper feature dimension is important for protocol identification. Insufficient features cannot represent the properties of signal-level protocol, that leads to inaccuracy for protocol identification. Conversely, too many features will increase the number of parameters and lead to the increase in computational complexity, but it is difficult to further improve the performance of protocol identification. Thus, we first learn the dimension of features. In this experiment, the dimension of the last fully connected layer is set to 5 to 60. Then, the number of parameters, runtime, loss, and accuracy of different dimensions are analyzed, and the feature dimension of 4LCNN is determined. The results are shown in figure 5. Note that, the accuracy score in this paper is calculated as the ratio between the number of correct predicted protocols to the total number of protocols, and the micro-precision, micro-recall, and accuracy share the same value.

Figure 5(a) shows the number of parameters with different output feature dimensions. As the feature dimension increases, the number of parameters increases rapidly, resulting in increased model training and optimization complexity. The training time of 4LCNN with different feature dimensions is shown in figure 5(b). It can be seen that the training time arises along with the increased feature dimension, and a large feature dimension leads to the sharp increased training time because of the large scale parameters. These two results reflect the complexity of 4LCNN. In figure 5(c), it shows the prediction errors of training and testing dataset when setting different feature dimensions. With the increase of feature dimension, the error decreases rapidly. When the feature dimension exceeds 25, the error curve flattens out, indicating the convergence of 4LCNN. Using softmax as a classifier, the performance of the 4LCNN is evaluated by identification accuracy, and the results are shown in figure 5(d). Based on the above experiments, the increased feature dimension within a certain range can improve the performance of 4LCNN, but excessively...
Increased feature dimension will not improve the identification performance. Therefore, trading off the identification performance against the computational complexity, the feature dimension of 4LCNN is set to 25. The parameters of 4LCNN are summarized in Table 3.

Table 3. The parameters of 4LCNN.

| Layer  | Feature maps | kernel size | sliding step |
|--------|--------------|-------------|--------------|
| Layer1 | Conv1        | 128         | 2*4          | 1            |
|        | Pool1        |             |              |              |
| Layer2 | Conv2        | 64          | 1*4          | 4            |
|        | Pool2        |             |              |              |
| Layer3 | Conv3        | 32          | 1*4          | 1            |
|        | Pool3        |             |              | 4            |
| Layer4 | Conv4        | 16          | 1*4          | 1            |
|        | Pool4        |             |              | 4            |
| Layer5: FC1 |               | dimension: 1024 |
| Layers: FC2 |               | dimension: 25  |
| Output(Softmax) | number of type: 7 |

After that, the proposed 4LCNN method is compared with other representative methods. The widely used method is the hand-design protocol identification, which depends on hand-engineered features with some statistical technologies and applies machine learning based classifier. Note that Moore summarized 248 classes of features for identification [20]. In this comparison, we select data-level features such as data composition, data packet size, length of the flow, and signal-level features such as high-order moment, high-order cumulant, and high-order cumulant spectrum to construct a 20-dimensional feature vector. Then, SVM is used for protocol identification [34]. Besides, another deep learning based protocol identification method is introduced for comparison. The authors used a stacked autoencoder model to learn generic protocol features [35]. In the comparison, we first train a 4-layer autoencoder(4LAEs) in a greedy layerwise fashion, and then the SVM is used as classifier.

First, the feature performance of different protocol identification methods is analyzed, including hand-design, 4LAEs and 4LCNN. In the experiment, the visually feature confusion matrix is used to analyze the protocol features obtained by the above three methods, while cosine is used to calculate the confusion matrix of features. The result is shown in Figure 6. It can be seen that the features extracted by 4LAEs and 4LCNN have better correlation for the same type of protocol than hand-design method, resulting in better performance for protocol identification. Due to the reliance on expert knowledge, the features selected by hand-design method is difficult to represent the characteristics of different types of protocols effectively, limiting the improvement of identification performance.

Then, the two deep learning based methods are evaluated, including 4LAEs and 4LCNN. The complexity and performance of the above methods when setting different feature dimensions are evaluated, and the results are shown in Figure 7. In Figure 7(a), the training time of 4LAEs and 4LCNN with different feature dimensions is demonstrated. Figure 7(b) shows the identification results of different feature dimensions. The accuracy arises along with the increased feature dimension, but leading to more training time. In addition, the 4LAEs with feature dimension of 30 and the 4LCNN with feature dimension of 25 are compared. According to the structure of 4LAEs and 4LCNN, the number of parameters are 4,976,702 and 136,937, respectively. Since the application of fully connected layers, the number of parameters in 4LAEs increases sharply due to the increased number of layers or feature dimension. In comparison, the 4LCNN utilizes the local connection and weight sharing to reduce the number of parameters. So it is an order of magnitude lower than the former. The reduction in the number of parameters also improves the efficiency of model training. The training time of the 4LCNN is reduced by 25%, compared with the 4LAEs. In terms of accuracy, the structured feature extraction method 4LCNN can achieve higher protocol identification accuracy.

At last, the 4LCNN based protocol identification method is compared with some state of art methods, including HD-SVM [34], HD-DT [36], DNN [37] and 4LAEs [35]. HD-SVM and HD-DT are two hand-design based methods, and the feature extractors of these two methods are designed by prior information and expert knowledge. So a 20-dimensional feature vector of signal-level protocol is...
constructed for the input of classifier, while the support vector machine and decision tree are applied respectively. The DNN based method takes the original signal-level protocol data as input and constructs the identification model by a multi-layer neural network. The 4LAEs consists of a four layer auto-encoder based feature extractor and a SVM based classifier. The results are shown in figure 8.

It can be seen that the identification accuracy of HD-SVM and HD-DT methods are worse than the other three methods, due to the reliance on expert knowledge to extract protocol features. The hand-design based feature extraction methods cannot guarantee the feature efficiency and the validity of the protocol identification results. DNN uses the signal-level sample data as features for training and improves the identification performance. However, due to the large dimension of input and multi-layer structure, the number of parameters is too large, that leads to the increase of training complexity. 4LAEs and 4LCNN first extract protocol features by learning model, and the features are used as the input of the classifier for protocol identification. The simulation results of the above methods are 97.68% and 98.42% respectively. These two methods depend on the learning based architecture, and use the model complexity and identification performance as the indicators to determine the model structure. The above two learning based methods can effectively improve the accuracy of protocol identification, compared with the traditional methods. In summary, the proposed 4LCNN method outperforms these four state of art protocol identification methods, and makes protocol identification more intelligent.

2) EVALUATION OF CNN-SP BASED METHOD
In this simulation, the proposed CNN-SP based variable length protocol identification method is evaluated. The structure of CNN-SP is shown in figure 2, which consists of four convolutional layers, three max pooling layers, one spatial pooling layer, and two fully connected layers. The CNN-SP takes the signal-level protocol with variable length as input, and extracts the structural features of the input data to construct the feature vector with fixed length. The dataset of the experiment is shown in table 4, where the training and testing dataset are denoted as “train_var_*.tfrecord” and “test_var_*.tfrecord” respectively, according to the length of protocol.

First, two different training modes are compared. The CNN-SP(Fix) mode first converts the samples to some fixed lengths of all possible values of the protocol length, and then trains the model by datasets of certain lengths. CNN-SP(Var) directly uses the variable length protocol, which retains the protocol length information. Compared with CNN-SP(Fix), it has a higher protocol identification accuracy, reaching 98.95%.

As can be seen from figure 9, since needs to convert the variable length protocols to a certain fixed length by padding or truncating, the CNN-SP(Fix) loses the length information of the protocol, that may leads to the performance degradation. The identification accuracy of CNN-SP(Fix) is 94.31%. Oppositely, CNN-SP(Var) directly trains the model by variable length protocol, which retains the protocol length information. Compared with CNN-SP(Fix), it has a higher protocol identification accuracy, reaching 98.95%.

Then, the CNN-SP is compared with some state of art protocol identification methods. Hand-design based method designs the feature extractor by prior information and expert knowledge, and constructs a 20-dimensional feature vector by extracting the data-level and signal-level features. Then, SVM is used for protocol identification [34]. 4LCNN trains the model by the fixed length dataset, which is achieved by padding the data in table 4 to the maximum protocol length. The result is shown in figure 10.

From figure 10, the identification accuracy of hand-design and 4LCNN are 92.68% and 98.42% respectively.
4LAEs and CAEs with L2 loss function is 97.68% and two loss functions are shown in figure11. The protocol loss functions are compared, including the L2 loss function constructed for unsupervised training. Besides, two different learning methods, which are used for automatic feature extraction method. Autoencoder is an unsupervised deep learning method, which is used for automatic feature extraction part and signal reconstruction part. The CAEs takes the signal-level protocol data as input and trains the model by minimizing the correlation coefficient loss function. Then, the extracted features are used to identify the protocols. The performance of the proposed CAEs is simulated on the signal-level protocol dataset.

In hand-design based method, the protocol length is one of the protocol features to construct feature vector. However, due to the variable length of these protocols and the overlapping range of length variations, the different protocols cannot be well differentiated. 4LCNN needs to convert the data to the fixed length, which may lead to the loss of length information and affect the identification accuracy. The CNN-SP directly uses the variable length protocols for training, and the accuracy is improved to 98.95%. Therefore, the effectiveness of our proposed method can be verified.

3) EVALUATION OF CAEs BASED METHOD

In this simulation, the proposed unsupervised CAEs protocol identification method is evaluated. The structure of CAEs is shown in figure 3, which consists of feature extraction part and signal reconstruction part. The CAEs takes the signal-level protocol data as input and trains the model by minimizing the correlation coefficient loss function. Then, the extracted features are used to identify the protocols. The performance of the proposed CAEs is simulated on the signal-level protocol dataset.

In order to evaluate the performance of proposed CAEs, we make a comparison with the state of art unsupervised feature extraction method. Autoencoder is an unsupervised deep learning method, which is used for automatic feature extraction. In the experiment, a four layer autoencoder(4LAEs) is constructed for unsupervised training. Besides, two different loss functions are compared, including the L2 loss function and the correlation coefficient loss function. The protocol identification results of two unsupervised training models and two loss functions are shown in figure 11.

From the above results, the identification accuracy of 4LAEs and CAEs with L2 loss function is 97.68% and 98.75% respectively, while the accuracy of 4LAEs and CAEs with correlation coefficient loss function is 98.36% and 99.12% respectively. It can be seen that CAEs has higher identification accuracy than the 4LAEs. Considering the loss function, the correlation coefficient loss function is more suitable for the feature extraction of signal-level data. Therefore, the effectiveness of our proposed method can be verified.

VI. CONCLUSION

Protocol identification is one of the important means of electromagnetic spectrum surveillance. In this paper, we mainly focus on signal-level protocol and propose the deep learning based signal-level identification methods. First, we proposed a multi-dimensional CNN based method for signal-level protocol identification. Then, a modified CNN-SP method was developed for variable length protocol identification. For unlabelled protocol feature extraction and identification, an unsupervised CAEs was constructed by minimizing the correlation coefficient loss function. The experimental results are presented, and the results show the effectiveness of the proposed methods for signal-level protocol identification.

REFERENCES

[1] W. Li, Z. Dou, and L. Qi, “Communication protocol classification based on LSTM and DBN,” IEEE Access, vol. 8, pp. 91818–91828, 2020.
[2] L. Zou, Z. Wang, J. Hu, Y. Liu, and X. Liu, “Communication-protocol-based analysis and synthesis of networked systems: Progress, prospects and challenges,” Int. J. Syst. Sci., vol. 52, no. 14, pp. 3013–3034, Oct. 2021.
[3] D. Tse and P. Viswanath, Fundamentals of Wireless Communication. Cambridge, U.K.: Cambridge Univ. Press, 2005.
[4] U. Cisco, “Cisco annual internet report (2018–2023) white paper,” Cisco, San Jose, CA, USA, White Paper C11-741400-01, 2020.
[5] The 49th Statistical Report on Internet Development in China, CNNIC, Beijing, China, 2022.
[6] X. Yao and X. Li, “Protocol recognition technology for the CCSDS space data link layer,” Aerosp. Electron. Warfare, vol. 28, no. 2, pp. 26–28, 2012.
[7] W. Wang, B. Bai, Y. Wang, X. Hei, and L. Zhang, “Bitstream protocol classification mechanism based on feature extraction,” in Proc. Int. Conf. Netw. Netw. Appl. (NINA), Oct. 2019, pp. 241–246.
[8] H. Su, Y.-D. Yao, and Z. Yang, “MAC protocol identification using support vector machines for cognitive radio networks,” IEEE Wireless Commun., vol. 21, no. 1, pp. 52–60, Feb. 2014.
[9] A. Callado, C. Kamienski, G. Szabo, B. P. Gero, J. Kelner, S. Fernandes, and D. Sadok, “A survey on internet traffic identification,” IEEE Commun. Surveys Tuts., vol. 11, no. 3, pp. 37–52, 3rd Quart., 2009.
[10] S. Sen, O. Spatscheck, and D. Wang, “Accurate, scalable in-network identification of P2P traffic using application signatures,” in Proc. 13th Conf. World Wide Web (WWW), 2004, pp. 512–521.
[11] T. Karagiannis, K. Papagiannaki, and M. Faloutsos, “BLINC: Multilevel traffic classification in the dark,” in Proc. Conf. Appl., Technol., Architectures, Protocols Comput. Commun., 2005, pp. 229–240.
