IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received July 16, 2020, accepted July 29, 2020, date of publication August 7, 2020, date of current version August 27, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3014996

A Review About RNA–Protein-Binding Sites Prediction Based on Deep Learning

JIANRONG YAN AND MIN ZHU
College of Computer Science, Sichuan University, Chengdu 610065, China
Corresponding author: Min Zhu (zhumin@scu.edu.cn)

This work was supported by the National Science and Technology Major Project of the Ministry of Science and Technology of the People's Republic of China, from January 2018 to December 2020, under Grant 2018ZX10201002-002-004.

ABSTRACT RNA-binding proteins (RBPs) play crucial roles in gene regulation. The advent of high-throughput experimental methods, has generated a huge volume of experimentally verified binding sites of RNA-binding proteins and greatly advanced the genome-wide studies of RNA–protein interactions. Many computational approaches have been proposed, including deep learning models, which have achieved remarkable performance on the identification of RNA–protein binding affinities and sites. In this review, we discuss machine learning and deep learning approaches, mainly focusing on the prediction of RNA and proteins binding sites on RNAs by deep learning. Furthermore, we discuss the advantages and disadvantages of these approaches. The workflow of deep learning is also revealed. We recommend some promising future directions of deep learning models in the study of RBP-binding sites on RNAs, especially the embedding, generative adversarial net, and attention model. Extraction and visualization methods involving motif are illustrated. Finally, we summarize the previous studies, and then compare the performance on different dataset.

INDEX TERMS Binding site, deep learning, motif discovery, RNA-binding protein.

I. INTRODUCTION

RNA-binding proteins (RBPs) play important roles in various cellular processes, such as alternative splicing, RNA editing, and mRNA localization [1], [2]. Identifying the binding sites of RBPs on RNAs is also crucial for understanding the mechanism behind many biological processes. RBPs are involved in several stages of post-transcriptional regulation. For example, HuR binds to target mRNA to enhance its stability and translation [3], whereas, TIA-1 and TIAR inhibit mRNA translation [4]. In addition, the dysregulation of RBPs and the mutation of binding target may lead to abnormalities and diseases [5], including muscular atrophies and neurological disorders. Thus, decoding the overview of RBP binding sites can give deeper insights into many biological mechanisms [6].

The knowledge of RBPs is vitally important for multiple aspects of gene expression regulation. It has been demonstrated that RBPs take over 5–10% of the eukaryotic proteome [7] and its number exceeds 1000 [8]. Although some experimental methods such as in vitro EMSA [9] and in vivo fluorescence [10] are powerful tools in characterizing RNA-protein interactions before the development of high-throughput techniques, they are time-consuming and expensive, and not all interactions can be successfully identified. Recently, several high-throughput sequencing-based approaches, e.g., CLIP-seq [11], SELEX [12] and RNAcompete [13], have indeed advanced the genome-wide studies of RBP binding sites and binding affinities, but they are also time-consuming and costly. Moreover, false positives and false negatives remain in the collected data due to experimental noise and the limitations of current techniques, such as limited mapping of splice sites. To acquire the accuracy information of binding sites in a timely manner, a series of RBP have been developed by means of binding sites bioinformatics tools [14]–[16]. In view of this, the computational methods were focused in the current study.

Machine learning methods are widely used and successful methodologies to learn features and relationships from data. Indeed, some well-known algorithms of traditional machine learning have been applied in predicting of RNA binding proteins. Hilal *et al*. designed the RNAcontext [17], deriving

The associate editor coordinating the review of this manuscript and approving it for publication was Shuihua Wang.

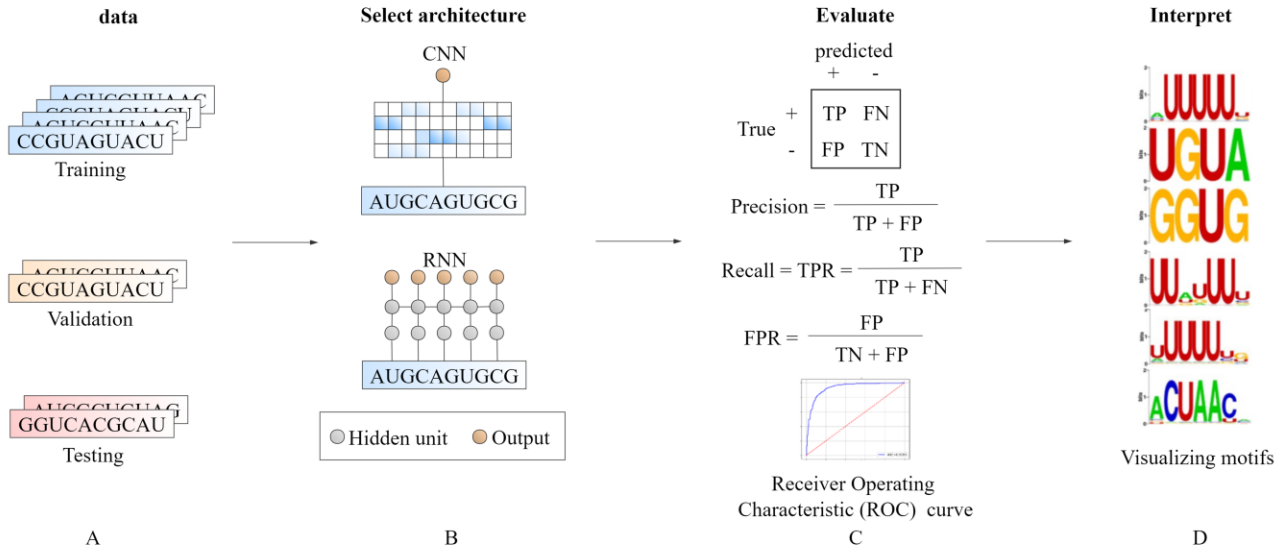

FIGURE 1. Deep learning workflow in RNA and protein binding sites prediction. (A). A dataset should be randomly split into training, validation and testing sets. Positive and negative samples should be balanced so that the predictor learns salient features rather than confounding factors. (B). The appropriate architecture is selected and trained in RNA protein binding sites prediction. For example, CNNs capture binding motifs, and RNNs capture long term information. (C). True positive (TP), false positive (FP), false negative (FN) and true negative (TN) rates are evaluated. When there are more negative than positive examples, precision and recall are often considered. Area Under Curve (AUC) is the area under the Receiver Operating Characteristic (ROC) curve and is also an important evaluation index. (D). The learned model is interpreted by visualizing motifs using sequence logo.

a position weight matrix (PWM) model and structural context preferences from RNAcompete [13] and CLIP data [18]. Orenstein *et al.* introduced the RCK [19] that utilizes the same input and optimization procedure as RNAcontext to infer model parameters. RCK could capture local preference better and achieve higher efficiency in the context of k-mer. Graphprot [18] was based on sequence- and structural-preference. The secondary structure of each RNA sequence was represented as combinatorial graphs in Graphprot. Li *et al*. [20] predicted protein-RNA binding residues using deep boosting-based approach via a total of 168 sequence features. For traditional machine learning, obtaining meaningful or task-related features is indispensable for performance, but features extracted require domain knowledge and designing manually by experts. However, deep learning has overcome those drawbacks by integrating the feature engineering step into a learning step. Instead of extracting features manually, deep learning obtains the informative representations by a self-taught manner [21]. Deep learning belongs to a class of machine learning, which can incorporate large-scale datasets, learn highly complex patterns, and incorporate existing knowledge. convolutional neural network (CNN) [22] was first applied to DeepBind [23] to model the mapping from sequence to binding strength and capture sequence motif automatically, which has attracted huge attention. Different from DeepBind, iDeepS [24] took structures into consideration for RBP binding specificity. It trained two individual CNNs and a long short-term memory network (LSTM) [25] for sequences and structures to capture binding sequence and structure motifs of RBPs. Considering the complementarity of multiple sources of data, e.g., sequences, structures, region type, Gene Ontology (GO) and clip-cobinding, Pan *et al.* proposed iDeep [26]. It integrated deep belief networks (DBNs) [27] and a CNN. Zhang *et al.* proposed deepnet-rbp [28], which incorporated sequence, secondary structure and tertiary structure features using DBNs. In contrast to current CNN methods, GraphProt2 [29] encoded input sequences as graphs, allowing the addition of base pair information in the form of graph edges and supporting variable length input. Till now, stacks of shallow models and combinations of different models are used frequently.

In this review, we summarize the recent progress of RBP binding sites, focusing on deep learning methods. We generalize available datasets that have been validated by biological experiments to build training datasets in section II. Importantly, we expand the basic theory in detail behind the different deep learning model in section III, such as CNN-LSTM, embedding, generative adversarial net (GAN), attention model, etc. In section IV, we introduce the applications of deep learning and machine learning in the prediction of RBP binding sites. We also show how to extract and evaluate motifs based on deep learning methods. In addition, we encode the data and answer questions that may arise during training in section V, including overfitting, hyperparameter adjustment, etc. Finally, we discuss the challenges and potential defects of the RBP binding sites prediction method based on deep learning. Deep learning workflow in RNA protein binding sites prediction is shown in the figure 1.

II. PROBLEM FORMULATION AND DATASETS

A. PROBLEM FORMULATION

RBP, is a class of ribonucleoprotein complex. When RBP binds to single or double stranded RNA, it is called interaction of RNA and protein. Generally, when the minimum distance between specific amino acid residues on protein and specific base on RNA is less than 3Å, it is considered that protein and RNA are bounded [30]. There also exists an interaction between RNA and protein. In this study, RBP-binding sites prediction is a classification problem. Formulating the prediction of RBP-binding sites is divided into two categories, based on whether binding occurs and how many sites binding to RBP. Whether or not binding occurs can be defined as a binary classification problem. For binary classification, the positive sequence is labeled as 1 and the negative sequence is labeled as 0. Here, a specific model is trained for each RBP. How many sites binding to RBP can be defined as a multi-classification problem, and the label is the type of RBP. In multi-classification, the number of labels is the sum of all the specific RBP types plus the non-binding sites, with only one general model to make prediction unlike the specific model. Here, non-binding sites are such sites that do not bind to all RBP in multi-classification.

B. DATASETS

At present, the frequently used datasets for identifying RBP binding sites are RNAcompete dataset [31], RBP-24 [18], RBP-31 [32], and RBP-67 [33] dataset.

RNAcompete-derived datasets consist of the measured binding preferences of 9 RBPs named HuR, Vts1p, PTB, FUSIP1, U1A, SF2/ASF, SLM2, RBM4 and YB1. In each RNAcompete experiment, the bindings of one protein to around 240000 short synthetic RNAs (30-40 nucleotides long) are measured, covering 24 different eukaryotes in total. The dataset includes 244 experiments, each of which contains the binding strength between a single protein and more than 240,000 RNA sequences. RNAcompete assesses an RBP's binding affinity for each sequence in an RNA pool. The estimate of affinity is based on the relative enrichment of that RNA sequence in the bound fraction versus the total RNA pool (as measured by transformed microarray intensity ratios). RNA pool is divided into two separate sets, Set A and Set B. A specific protein binds to a set of predesigned oligomers and measures binding by hybridizing with complementary probes on the microarray. This dataset is used by many models, such as GraphProt [18], RCK [19], Deepbind [23], RNAContext [17], DLPRB [34], etc.

RBP-24 dataset covers 24 experiments of 21 RBPs, in which 23 datasets are attained from doRiNA [35], and the remaining one which measured the PTB binding sites by HITS-CLIP is attained from [36]. The RBP binding positive sites are identified by the CLIP-based experiments, and they can be downloaded from doRiNA. The nonbinding negative sites are derived by shuffling the coordinates of binding sites within all genes with at least one binding site using bedtools shuffle [37]. In general, 80% of the training data are selected as training set, the remaining 20% as validation set. The independent testing set is derived directly from testing set. The dataset is available at http://www.bioinf.uni-freiburg.de/Software/GraphProt.

In RBP-31 dataset, the CLIP-seq data consists of 19 proteins with 31 experiments. Each nucleotide in the interaction site cluster derived from CLIP-seq is considered a binding site. In order to reduce redundancy, further random sampling of positive binding sites with the highest number of cDNA, there is no continuous site in the genome. Finally, only one of the sites with the highest cDNA count is selected as the positive sample. The negative site is extracted from any of 31 experiments that are not identified as interacting. In the experiment, there are 4000 cross-linking sites for training, 1000 samples for model optimization and validation, and another 1000 samples for independent testing. The dataset is available at https://github.com/mstrazar/ionmf.

RBP-67 dataset is constructed from RNAcommender [33]. Proteins interact RNA with 72226 UTRs (untranslated regions in mRNAs) for a total of 502178 interactions from the AURA 2 database [38]. The AURA 2 database includes a manually curated and comprehensive catalogue of experimentally determined interactions between human RBPs and UTRs (untranslated regions in mRNAs), including 67 distinct RBPs. Distinct RBP has different number of binding sites. Positive samples can be directly extracted from experimentally generated binding sites. The nonbinding negative sites are derived by shuffling the coordinates of binding sites within all genes with at least one binding site using bedtools shuffle. Pan *et al.* [39] also investigated that how many RNAs had multiple binding proteins, and found that 74.7% of RNAs had at least two binding proteins.

The subsequences with peak expression level in CLIP-seq serve as positive samples, and without binding sites of region in CLIP-seq belong to negative samples. Therefore, the label of samples is determined by experiments whether they interact with the corresponding RBP. In addition, RNA sequence needs to de-redundant by CD-HIT [40]. Although many datasets have been proposed, in most studies only part of datasets for prediction of RBP.

III. DEEP LEARNING ALGORITHMS

Deep learning has become a field with many practical applications and active research topics, which has been applied to image processing [41], speech recognition [42], translation, etc. A deep learning model typically contains a number of representation layers, all of which are automatically learned from the training data. Learning means finding a set of weights for all layers so that the network can correctly match each RNA sequence to corresponding label. In this section, we will introduce four typical deep learning models, namely CNN [22], recurrent Neural Network (RNN) [43], DBN [27] and graph convolutional networks (GCN). In addition, we will introduce stacked deep neural network model for predicting RBP binding sites, such as CNN-LSTM, Embedding, GAN, attention model, etc.

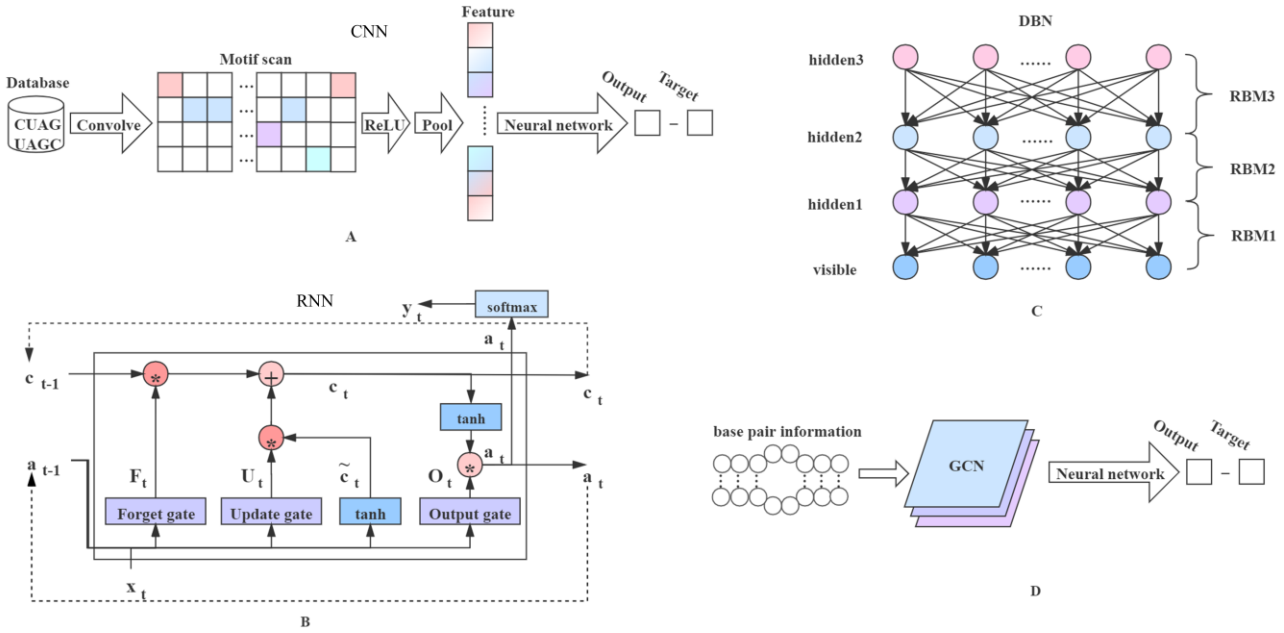

FIGURE 2. (A) Convolutional neural network is used to process the sequence, including Convolution layer, ReLU activation, and pooling layer. In general, softmax is also required in the end of model. (B). The framework of LSTM is shown with three gates. (C). It is the framework of DBN, which consists of RBM. (D). Sequence and base pair information are input to GCN for discrimination.

A. CONVENTIONAL DEEP LEARNING MODELS

1) CONVOLUTIONAL NEURAL NETWORK

CNN is a kind of neural network specially processing data with grid structure. Convolutional networks perform well in many applications, such as sequential data [44] and image data [45]. We first encode RNA sequences into one-hot encoding showing the presence of nucleotide A, C, G, and U. Then the one-hot encode matrix is fed into a CNN, which involves convolution, activation, and pooling operations. one-hot encoding will be introduced in section V. The CNN layer preserves the spatial information and output feature maps for subsequent processing. The advantage of CNN is that it no longer separates feature extraction and model learning into two independent steps like traditional statistical learning algorithms. It adopts a data-driven approach to learn the feature and classification model from the original input simultaneously, reducing the potential mismatch effect between feature extraction and learning classification model. The CNN model has been widely applied to the prediction of RNA binding proteins of DNA or RNA.

As is shown in the figure 2.A, in forward propagation, let each filter slide over the width and height of the input data, and then compute the inner product of the entire filter and the input data at any point. A two-dimensional activation diagram is generated, and each spatial position on the activation diagram represents the response of the original sequence to the filter. On each convolutional layer, there will be a set of filters, with number of 16 or 102 according to the previous studies [26]. At present, CNN is used because the filter has strong interpretability. The number of filters is related to the type of motif. Pooling operations include maximum pooling and average pooling. Pooling operation aims at reducing the dimension of matrix, which can retain the most representative elements in the matrix after the convolution operation and speed up the matrix operation. Generally, the input data is two-dimensional and represents the global RNA sequence information. In addition, Pan and Shen [46] also used k-mer to treat multiple subsequences in global sequence as RGB channels in an image, which is three-dimensional.

2) RECURRENT NEURAL NETWORK

RNN is widely used in NLP fields, such as automatic translation [47], emotion analysis [48], and human-computer dialogue. It could be used to process sequence data of any length. Timestep is a very important concept in RNN. At different timestep, memories are stored and flowed in hidden cells, and each hidden cell has an output. On the basis of the different propagation direction, bidirectional RNN emerged, which enables the network to adjust the current state according to the past and future state. Because of vanishing gradient and explosion gradient problems, it is difficult to obtain long-term dependence for RNN training. In order to solve the problem of long-term dependence, LSTM [43], gated recurrent unit (GRU), and other relevant variants are proposed to improve RNN. The details of LSTM are shown in the figure 2.B. LSTM cell is given in the following formulas.

$$\tilde{c}_t = \tanh(W_{ct} * a_{t-1} + W_{ct} * x_t + b_c) \quad (1)$$

$$U_t = \sigma(W_{ut} * a_{t-1} + W_{ut} * x_t + b_u) \quad (2)$$

$$O_t = \sigma(W_{ot} * a_{t-1} + W_{ot} * x_t + b_o) \quad (3)$$

$$F_t = \sigma(W_{ft} * a_{t-1} + W_{ft} * x_t + b_f) \quad (4)$$

$$c_t = U_t * \tilde{c}_t + F_t * c_{t-1} \quad (5)$$

$$a_t = O_t * c_t \quad (6)$$

where σ is the Sigmoid function, and tanh is a function to push the values to be between -1 and 1. $\tilde{c}_t$, U_t, O_t, and F_t represent outputs of the memory gate, update gate, forget gate, output gate, respectively. t represents timestep. W_{ct}, W_{ut}, W_{ot}, and W_{ft} represent weights in the t timestep. b_c, b_u, b_o and b_f are biases. U_t updates the state at timestep t, and F_t removes the state to be forgotten. Next, c_t and a_t are transferred to the next unit, as is shown in figure 2.B. x_t is input data. It is worthwhile to note that the initial values with $c_t = 0$ and $a_t = 0$. Three gates are designed to enhance the ability of LSTM to capture long-term dependence. However, RNN and variants are less used to predict RNA protein binding sites, because RNN has poor interpretability and cannot process excessively long sequences. Since the length of RNA sequences exceeds 500, the long-term dependence problem cannot be well solved. Therefore, it is not appropriate to deal with RNN directly. CNN is utilized to reduce the dimension and then concatenate RNN so as to solve this problem.

3) DEEP BELIEF NETWORK

DBN [27] is another deep learning algorithm to learn high-level features from massive data, which is also a popular choice for constructing the computational models in RBP prediction [26] recently. DBN is a probabilistic generation model composed of multiple layers of neurons. The visible layer is used to input training data, while the hidden layers are used to extract features. So hidden layer neurons also called feature detector. The components of DBN are Restricted Boltzmann Machines (RBM) [26]. The training process of DBN is carried out layer by layer, as is shown in the figure 2.C. In each layer, data vectors are used to infer the hidden layer, which is then treated as the data vector of the next layer. In fact, each RBM can be also used as a separate cluster. RBM has only two layers of neurons. The first layer, called the visible layer, consists of visible units for input to training data. Another layer is called the hidden layer, accordingly, and is made up of hidden units that act as feature detectors. It's worth noting that many studies have shown that the CNN and DBN have their own advantages owing to different deep learning architectures. For example, CNN is more appropriate for sequential data, and DBN prefers numeric input. It prompts us to consider how to combine the advantages of CNN and DBN to better predict the RBP binding sites and find the sequence motifs.

4) GRAPH CONVOLUTIONAL NETWORK

GCN [49] performs a convolution on a graph rather than on an image composed of pixels. Graph is composed of several nodes and the edge connecting two nodes, which is used to describe the relationship between different nodes. All nodes connected with this node are its neighbors. GraphProt2 [29] provides predictions for the entire site and profile for the input sequence, that is, a score for the entire sequence or a single score for each nucleotide position in the sequence. In addition, it uses GCN and contains base pair information, as is shown in the figure 2.D. The length of the input sequence can be variable, which makes the method more flexible, as well as the use of variable-sized windows in profile prediction.

Like traditional CNN, convolution operation in Euclidean space extracts the features of pixels with a fixed size learnable convolution kernel. In the Euclidean space represented by images, the number of neighbors of node is fixed. However, in the non-Euclidean space, the number of neighbors of node is not fixed. Because the neighbor nodes in the graph are not fixed, traditional convolution kernel can not be directly used to extract the features of the nodes in the graph. The real difficulty is that the number of neighbor nodes is not fixed. Therefore, we need to transform the non-Euclidean space into Euclidean space, and find a convolution kernel that can deal with the variable length neighbor nodes to extract features on the graph. The essence of graph convolution is to find the learnable convolution kernel for graphs. There are two kinds of graph convolution networks, one is based on spatial domain, the other is based on frequency domain.

B. THE VARIANTS OF DEEP LEARNING MODELS

1) CNN-LSTM

The model of CNN-LSTM is the most popular for processing RBP sequences data. In this model, CNN is utilized to obtain sequence features for RBPs, and RNN is used to extract high-level context representation of sequence features. CNN processes each input global sequence, it is not sensitive to the sequence of time steps, which is different from RNN. Of course, many convolutional and pooling layers can be stacked on top of each other in order to identify longer-term patterns, so that the upper layer can observe longer sequences in the original input. Because the RNA sequence data is relatively simple, much deep network structure is easy to cause overfitting. One way to combine the speed and lightness of the convolutional neural network with the order sensitivity of RNN is to use the convolutional neural network as a pre-processing step in front of RNN, especially for very long sequences that RNN cannot handle directly. Convolutional neural networks can convert long input sequences into shorter sequences composed of higher features by down-sampling [50], [51], as is shown in the figure 3.A. These sequences after down-sampling, are composed of extracted features, then become inputs to RNN in model. This is the most common model in many studies, such as DLPRB [34], iDeepS [24].

2) EMBEDDING

Inspired by the success of word embedding in natural language processing [52], it is also increasingly used in bioinformatics [53]. Word embedding learns a distributed representation for words, thereby reducing the dimension of word space. Word embedding is context independent static

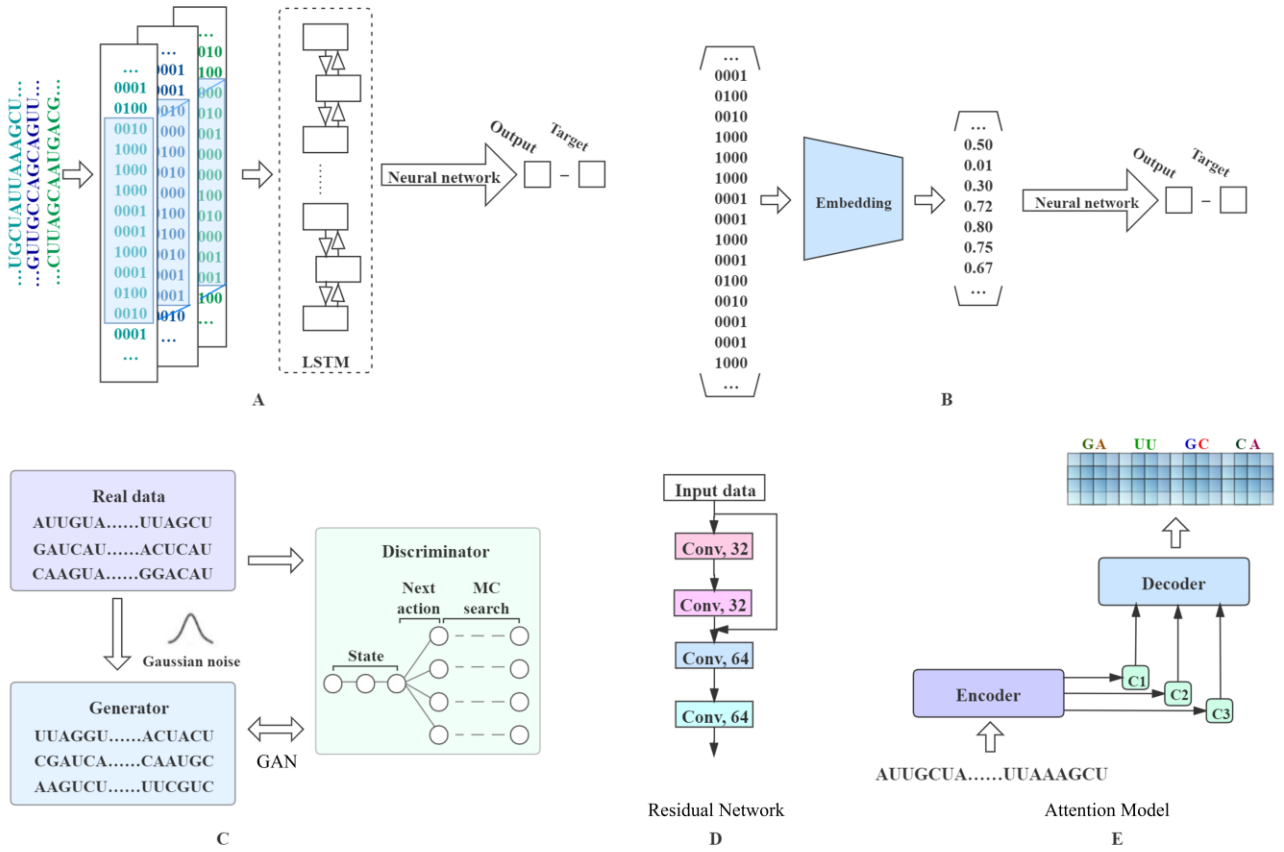

FIGURE 3. (A) The sequences are fed into CNN to extract features. Then, long and short-term memory training is carried out through LSTM. (B). The high-dimensional sequence is reduced by word embedding, and then input into neural network for prediction. (C). The GAN have two neural network models, the discriminator and the generator, competing against each other during learning. The discriminator aims to distinguish the synthetic from the real data, while the generator is trained to confuse the discriminator by generating high quality synthetic data. (D). Sequences are processed using jump connection by residual network. (E). The attention network could be in encoder and decoder. These attention weights are used to construct the content vector C1, C2, and C3, which is passed to the decoder as input. At each decoding position, the content vector is the weighted sum of all the hidden states of the encoder and their corresponding attention weights.

vector, including statistical tools such as word2vec [54], glove [55], etc. Asgari and Mofrad proposed BioVectors [56] using word embedding technology based on n-gram for biological sequences, e.g., DNA, RNA, protein. BioVectors can describe the basic patterns of biological sequences in the sense of biochemistry and biophysics.

In this review, we consider each k-mer as a word and each sequence as a sentence, and learn the distributed representation of k-mers using skip-gram. We map each subsequence of k-mer into a high-dimensional vector and assemble all high-dimensional vectors into subsequence space as mentioned section V. The subsequence space of k-mer is 4^k by one-hot encoding, and each encoding corresponds to one subsequence. A window of length k slides over a global sequence of length L. When the step size is s, the RNA sequence of length L has (L-k +1)/s number of words. The embedding layer compresses k-mers of the input sequences into their distributed representations. Where the distributed representation of k-mers are learned from genome-wide UTR sequences using the word2vec or glove algorithm, which is an orthogonal matrix essentially. The high-dimensional one-hot matrix is embedded into a much lower dimensional vector continuous space, as shown in figure 3.B. Each word is mapped to a vector in the real field, called a word vector. In natural language processing, word vector distance proximity can indicate semantic similarity. However, words in RNA sentences cannot express semantics after dimensionality reduction, there is no evidence indicates that two subsequences are similar. It is inappropriate to use a current word embedding algorithm. More meaningfully, we can construct an orthogonal matrix algorithm for RNA subsequences specifically. Orthogonal matrix is generated by subsequence alignment, which is more suitable for RNA word embedding. It will be a meaningful study in the future.

3) GENERATIVE ADVERSARIAL NET

Sequence generation plays a major role in natural language processing, which is necessary to many applications such as machine translation [57], image captioning [58], and dialogue systems [59]. The main idea behinds GANs proposed by

Goodfellow *et al.* [60] is to have two neural network models, the discriminator and the generator, competing against each other during learning, as shown in figure 3.C. The discriminator aims to distinguish the synthetic from the real data, while the generator is trained to confuse the discriminator by generating high quality synthetic data. GANs achieve great performance in computer vision tasks such as image synthesis [61]. Their successes are mainly attributed to training the discriminator to estimate the statistical properties of the continuous real-valued data, e.g., pixel values. However, the GANs have difficulties in dealing with discrete data. In natural languages processing, the text sequences [62] are evaluated as the discrete tokens whose values are non-differentiable. Therefore, the optimization of GANs is challenging. During learning, the policy gradient technique [63] is adopted to overcome the non-differentiable problem. At present, studies [64], [65] have demonstrated that GAN can synthesize text sequences.

We can construct an adversarial prediction model, which makes network stronger. In addition, prediction using deep learning models generally requires large-scale training data. The studies showed that the corresponding AUC was lower for those with less RBPs. To solve this problem, we may generate more RNA sequences with similar features through GAN. Synthetic RNAs have similar features relative to real sequences, and synthetic RNAs can better assist in prediction. The training set consists of synthetic RNA and real RNA, which are used to train a better model for prediction.

4) RESIDUAL NETWORK

The residual network [66] won the competition of image classification and object recognition in 2015. The residual network is constructed by residual blocks. The residual network is easy to optimize in RBP prediction and can improve the accuracy by increasing the depth. The internal residual block uses jump connection as shown in the figure 3.D, which alleviates the gradient disappearance problem caused by increasing depth in the depth neural network. iDeepE was constructed by residual network.

5) ATTENTION MODEL

In recent years, attention model has been widely used in various fields of deep learning, including image processing, speech recognition and natural language processing [51]. Generally, the attention model includes encoder and decoder. There have many studies, including Seq2Seq [67], Transformer [68], Bert [69], etc.

Pan *et al.* proposed iDeepA [70], which used an attention-based convolutional neural network model. Attention module in the network learn the attention weight automatically, which can capture encoder hidden state (i.e., candidate state) and decoder hidden state (i.e., query state). These attention weights are used to construct the content vector C1, C2, and C3, as shown in figure 3.E, which is passed to the decoder as input. At each decoding position, the content vector is the weighted sum of all the hidden states of the encoder and their corresponding attention weights. There are many variants of attention model. In hard attention, only one position weight is set to 1 at a time, and the rest is set to 0. Here, only one position is focused every time. However, the soft attention takes care of all positions each time, different position weights are different. It is significantly important innovation for both of them to combine, or transfer. In RBP prediction, different positions contribute different weights to the prediction results. In the section V, it is found that the probability of motif is different in different position.

IV. APPLICATION OF DEEP LEARNING IN THE PREDICTION OF RBP BINDING SITES

Owing to advances in high-throughput technologies, a deluge of RBP data has been obtained in recent years. Some successful applications of deep learning in RNA and protein binding fields are reviewed in this section. In addition, we also summarize the extraction and evaluation of motif.

A. APPLICATION OF TRADITIONAL MACHINE LEARNING-BASED METHODS

Structure annotation profiles and RBP affinity were applied to RNAcontext [17] together with a set of sequences for the given RBP. Clearly, residue preferences were inferred as a PWM. The relative structural preferences were inferred according to different structural contexts. Finally, RNAcontext implemented AUC of 0.82 on the RBP-24, and AUC of 0.43 on the RNAcompete dataset in vitro binding prediction. Orenstein *et al.* introduced RCK [19], improving RNAcontext by a k-mer sequence- and structure-based binding model with local structure preferences. In particular, the input data and optimization procedure of RCK was the same as RNAcontext to infer model parameters. RCK in a k-mer based context could capture local preferences, and implement 0.46 of AUC on the RNAcompete dataset in vitro binding prediction.

In GraphProt [18], highly probable secondary structures were derived using RNAshapes [71] in the context of each potential target site. The RNA secondary structure of each RBP-bound context jointing sequence information was represented as a hyper-graph. In hyper-graph, secondary structure of nucleotide on the type of substructures was annotated, including stem, hairpin loop, internal loop, multi-loop, bulge loop, and external region. Features of sequence were extracted from the hypergraphs using efficient graph kernels. It's AUC was 0.89 on the RBP-24, and 0.82 on the RBP-31 dataset. However, it cannot extract the structure motif. GraphProt applied support vector machines (SVM) to graphs. it sped more time about 7 days for each of the 255 proteins. Oli [72] was developed using SVM approach and own AUC 0.77 on the RBP-31, based on tetranucleotides as features. In addition, there were two extensions as comparison: OliMo, which added protein-specific binding motifs, and OliMoSS, which also added secondary structure information [73]. The performance of predictions with different feature were compared with each other. The result of Oli suggested original sequence information of feature can be

TABLE 1. Machine learning-based methods for predicting RNA-protein binding sites.

Method	Feature	Model	Motif	Source
GraphProt	Sequences, structures	SVM	Sequences, structures	http://www.bioinf.uni-freiburg.de/Software/GraphProt
RNAContext	Sequences, structures	Probabilistic models	Sequences, structures	http://www.cs.toronto.edu/ ~hilal/rnacontext
iONMF	Sequences, structures, region type, GO, clip-cobinding	Orthogonal matrix factorization	Sequences	https://github.com/mstrazar/ iONMF
Oli	Sequences	SVM	None	None
RNAcommender	Sequences	Matrix factorization	None	http://rnacommender.disi.unitn.it.
RCK	Sequences, structures	Probabilistic models	Sequences, structures	http://rck.csail.mit.edu
deepboost	Sequences	Boosting	Sequences	http://github.com/dongfanghong/deepboost

fully distinguished to predict specific RNA-protein interactions. In iONMF [32], k-mer sequence, secondary structure, CLIP co-binding, GO information, and region type were integrated to predict RBP binding sites, using orthogonality-regularized nonnegative matrix factorization method and achieving AUC of 0.85 on the RBP-31. It used multiple data sources of RBPs to perform better. RNAcommender [33] was introduced to discover genome-wide recommendation of protein targets. It utilized variant of matrix factorization based collaborative filtering skillfully, and removed the original orthogonality constraints. RNAcommender obtained data from high-throughput experiments as train data, unexplored RBPs without known targets as test data. RNAcommender used a recommender system [74] essentially, which used the known interaction information to predict the unknown. Li *et al.* [20] predicted protein-RNA binding residues using deep boosting-based approach via a total of 168 sequence features based on k-mer. Details of these methods are shown in TABLE 1.

B. VARIOUS APPLICATION BASED DEEP LEARNING

1) APPLICATION OF CNN AND LSTM

DeepBind [23] was proposed by Alipanahi *et al.*, and was the first method to implement predictions of RBP binding sites using sequences by deep learning. In addition, motifs of RNA feature can be extracted from CNN filters in DeepBind. It pushed the prediction of RBP binding sites a big step forward, which achieved AUC of 0.92 on the RBP-24, 0.85 on the RBP-31, and 0.41 on the RNAcompete dataset in vitro binding prediction.

Pan *et al.* developed iDeepS [24], which consisted of CNNs and a bidirectional LSTM. It identified the sequence and structure binding motifs simultaneously. Both of the sequence and structure binding motifs could be fully automatically captured by iDeepS. iDeepS not only implemented better performance than peer sequence-based methods on average (e.g., DeepBind), it also surpassed some approaches integrating multiple sources of hand-designed features (e.g., iONMF). As compared to GraphProt, which requires a complicated postprocessing step, iDeepS easily extracted learned parameters of the convolved filters to PWMs and accomplished identification of the sequence and structure motifs. As the input of encoding was easier and simple, iDeepS had a wider range of applications based on sequence. However, AUC with 0.86 on the RBP-31 was achieved lowly in ideepS, which may due to improper selection of optimization methods and training times. When processing RNA data, the shallow network layer often had a higher prediction accuracy. RNA data, as simple text data, has low dimension and little content of data expression. Therefore, the number of shallow network layers and fewer neurons is better. MSC-GRU [75] was proposed infer the binding motifs of RBPs. It integrated a multi-scale CNN layer and a bidirectional GRU layer to capture the local combination pattern, achieving AUC of 0.920 on the RBP-31. DLPRB [34] proposed by Ben-Bassat *et al.*, utilized two DNN architectures, CNN and RNN, respectively, with input features including sequences and structures. It inferred precise RNA-binding models from high-throughput in vitro data, and implemented the test in vivo. Feature coding of structure was a structure probabilities matrix predicted by RNAplfold [76]. Here, CNNs reach high performance owing to its ability of analyzing spatial information. DLPRB achieved an excellent high pearson correlation coefficient [77], and completed high average AUC on the eCLIP experiments and AUC of 0.63 on the RNAcompete dataset in vitro binding prediction.

Pan *et al.* introduced iDeepE [46]. It integrated global and local convolutional neural networks (CNNs) to predict RNA–protein binding sites. Complete RNA sequences were fed to global CNN, which padded into the same length according to the predefined longest sequence. Each RNA sequence was split into multiple overlapping fixed-length subsequences that looked like a channel just as RGB channel in images for the local CNN. iDeepE was evaluated with eight variants against three state-of-the-art methods, including CNN, CNN-LSTM and ResNet. The results Indicated that fusing the local and global ResNets led to better performance. Generally speaking, it had been confirmed that ResNet and CNN were powerful tools in prediction of RNA binding protein. ideepE achieves excellent performance with AUC of 0.93 on RBP-24 datasets.

2) APPLICATION OF DBN

A novel hybrid CNN and DBN were applied to iDeep [26] to predict the RBP binding sites and motifs on RNAs. It integrated multiple sources of data, e.g., sequence, structure, motif, CLIP co-binding, region type, through cross-domain knowledge at an abstraction level to enhance the prediction

ability. CNN and DBN are designed into framework of iDeep, managing sequence with CNN and the remaining four (i.e., structure, motif, CLIP co-binding, and region type) with DBN. Then iDeep used deep network of multimodal comprising DBNs and CNNs to integrate these extracted representations, achieving AUC of 0.90 on the RBP-31. The CNN was able to capture regulatory motifs, that were recurring patterns in RNA sequences with a biological function. The DBN learned high-level features regarded as a joint distribution determined by hidden variables for different inputs. Since multimodal deep learning could learn shared representation, integrated iDeep performed better than individual modality.

3) APPLICATION OF GCN

GraphProt2 [29] was presented by a computational RBP binding sites prediction method based on GCN. GraphProt2 encoded input sequences as graphs, allowing the addition of base pair information in the form of graph edges. It also could encode base-pair information (i.e., annotated connections between non-adjacent bases), which required a more flexible method to deal with these limitations while supporting other functions. In contrast to GraphProt, GraphProt2 provided an improved profile prediction mode, i.e., the calculation of position-wise prediction scores for the entire RNA sequence. GraphProt2 certainly provided state-of-the-art predictive performance, exceeding iDeepS.

4) APPLICATION OF ATTENTION LEARNING

To better characterize RBP binding sites, iDeepA [70] was introduced by an attention-based convolutional neural network model. It integrated CNNs and two levels of attentions, to predict binding or not from filters of CNNs only by RNA sequence. iDeepA extracted three levels of abstract features, including the output feature maps from the CNN and the outputs from two attention model. iDeepA and DeepBind had similar average AUC, with AUC of 0.92 on the RBP-24. However, iDeepA performed excellent with small training set on that DeepBind did not achieve high AUC. The results indicated attention mechanism can enhance the learning ability on small dataset and quickly focused on important feature. But identifying interpretable motifs still not be solved.

5) APPLICATION OF EMBEDDING

Since neural networks have difficulty handling high dimensional data, sequences are often converted to k-mers. Word emdedding about Word2vec was applied to iDeepV [78] to represent k-mers in a lower dimensional space, attaching 1-D CNN to predict the RBP binding sites. It performed faster than other structure-profile based methods, such as GraphProt, deepnet-rbp and iDeepS, in that iDeepV only required sequences. iDeepV yielded average AUC with 0.85 on the RBP-31. deepRKE [79] similar to iDeepV used an unsupervised shallow two-layer neural network to automatically learn the distributed representation of k-mer by considering its neighbor context. Compared with the traditional k-mers method, the distributed representation can effectively detect the potential relationship and similarity between k-mers. The distributed representations of sequences and secondary structures were fed into CNN and bidirectional long short-term memory networks (BLSTM) to achieve AUC of 0.934 on the RBP-24 and 0.873 on the RBP-31. However, iDeepV and deepRKE can not extract sequence and structure motif by using word embedding.

6) APPLICATION OF SPLINE TRANSFORMATION

In general, innovation of prediction focuses on input features or model variants, but CONCISE [80] was different. It introduced the Spline transformation, which was integrated into a neural network similar to ReLU of function. It applied scalar features into neural networks efficiently, such as distances. Spline transformation was based on smooth penalized splines, and can be applied to the context of each network layer. CONCISE achieved superior performance with AUC of 0.92 on RBP-31 datasets. One limitation of spline transformation was that scale of the input features keeps indispensable and had to be selected in advance, because spline knots were set uniformly across the whole range of feature values. Therefore, pre-processing studies must be conducted to determine the appropriate scale that best fits the current problem.

7) MULTI-LABEL PREDICTION

At present, all methods trained specific models for every RBP. In other words, each model could only predict RNA targets for one RBP, which led to the relationship among different RBPs being totally ignored. For instance, different RBPs can be associated to predict RNA-protein interaction, in virtue of share of similar binding domains. In different databases, it was different for construction rules to negative samples of each RBP. In order to reduce the impact of negative sample quality, RBPs were utilized as labels directly. Existing studies demonstrated that RNAs can bind multiple proteins and 74.7% of RNAs have at least two binding proteins. iDeepM [39] was proposed. This task can be defined as a multi-classification problem. The iDeepM model was the classic CNN-LSTM, achieving AUC of 0.87 on RBP-67 datasets.

DeepRiPe [81] was proposed to characterize RBP binding preferences, which also was a multi-category task and multimodal DNN model similar to iDeepM. DeepRiPe used a modular structure to learn information features from DNA sequence and transcript region types, because many RBPs tended to bind to specific regions of transcripts. As we know, this allowed the model to use shared information between tasks, which was critical for focusing the model on the unique characteristics of the RBP binding site. By predicting how many RBPs were reacted, we could obtain biological insights in the case of the mechanism of protein RNA-binding. RBPs as labels could indeed alleviate the instability of negative samples, but it brought about an imbalance in the number of labels. In the RBP-specific model, as a binary classification task, negative samples were created to be comparable to

TABLE 2. Deep learning-based methods for predicting RNA-protein binding sites.

Method	Feature	Model	Motif	Source
DeepBind	Sequences	CNN	Sequences	http://tools.genes.toronto.edu/ deepbind
iDeepS	Sequences, structures	CNN-LSTM	Sequences, structures	https://github.com/xypan1232/ iDeepS
iDeepE	Sequences	ResNet	Sequences	https://github.com/xypan1232/ iDeepE
MSCGRU	Sequences	Embedding-CNN-RNN	None	None
DLPRB	Sequences, structures	CNN/RNN	Sequences, structures	https://github.com/ilanbb/dlprb
iDeep	Sequences, structures, region type Motif, clip-cobinding	CNN-DBN	Sequences	https://github.com/xypan1232/ iDeep
GraphProt2	Sequences, structures	GCN	None	None
iDeepA	Sequences	CNN-attention	None	https://github.com/xypan1232/ iDeepA
iDeepV	Sequences	Embedding-CNN	None	https://github.com/xypan1232/ iDeepV
deepRKE	Sequences, structures	Embedding-CNN-RNN	None	None
CONCISE	Sequences, structures, region type motif, clip-cobinding, relative distance to landmarks	Spline transformation	None	https://github.com/gagneurlab/ concise
iDeepM	Sequences	CNN-LSTM	None	https://github.com/xypan1232/ iDeepM
DeepRiPe	Sequences, 3'UTR, 5'UTR, CDS, or intron region	CNN	sequences	https://github.com/ohlerlab/DeepRiPe
deepnet-rbp	Sequences, structures	DBN	Sequences, structures	https://github.com/thucombio/ deepnet-rbp

the number of positive samples. However, for multi-category task, the samples of each RBP were taken as positive samples, and the corresponding numbers were far apart. For example, in RBP-24, there were 1197 number of positive samples in ALKBH5, while ELAVL1 PAR-CLIP (C) had 125202 number of positive samples. The number of RBPs was extremely unbalanced, which may affect the predictions.

8) PREDICTION BY RNA SEQUENCE, SECONDARY AND TERTIARY STRUCTURE JOINTLY

Deepnet-rbp [28] was developed to integrate the RNA sequence, secondary and tertiary structural profiles, and constructed a unified representation to extract the hidden structural features of RBP targets. It introduced RNA tertiary structural profiles for the first time and a new method to construct the RNA tertiary structural profiles. Multimodal DBN was applied to deepnet-rbp, a generative model in nature, which could generate sequence and structure motifs directly, achieving AUC of 0.90 on the RBP-24. TABLE 2 shows the features, models, sources and different kinds of motifs that can be obtained. The detail of those methods is shown in TABLE 2. In these studies, CNN is the most common method according to TABLE 2. It may because CNN is good at processing sequential data.

C. PERFORMANCE COMPARISON

To gauge the performance of every method, we summarize the performance of studies on the comprehensive dataset of RNAcompete dataset, RBP-24 dataset, and RBP-31 dataset from every study. In RNAcompete dataset, each experiment contained the binding intensities between a single protein and more than 240000 RNA sequences. Here, models were trained on sequences from set A and predicted the intensities on set B. There are two reasons that may affect the accuracy of the in vitro model for predicting in vivo binding. First of all, it is known that there is noise in vivo and deviation in experiment. In addition, the accuracy of RNA structure prediction in vivo is lower than that in vitro, so the learned structure preference may not improve the binding prediction. The performance of four methods are collected as follows, including RNAcontext, RCK, DeepBind, and DLPRB. The result of AUC is shown as figure 4.A. In addition, the result on the RBP-24 and RBP-31 is shown as figure 4.B and figure 4.C. RBP-24 and RBP-31 have training set and independent testing set.

D. EXTRACTING AND VISUALIZING BINDING MOTIFS

1) EXTRACTING BINDING MOTIFS

Motif is based on mathematical statistical models in biology [82]. It is a locally conserved region in a sequence, or a short sequence pattern common to a set of sequences. Motif is the probability that bases (or secondary structures) appear in every positions of the short sequence, with 7 bp [13]. Motif is visualized by sequence logo [83]. In this model mentioned in this review, motifs can be extracted by using convolutional neural network mostly. After several iterations, the weight of each residue is recorded in the position of convolution kernel. Since some operations in the network may lead to the deletion of some residue information, the extraction length of motif in network is set to 1.5 times length of the original motif, which is determined to be 10 bp [26]. We suppose an RNA sequence with a fixed 519 nucleotides. Sequence data is convoluted by the filter parameters from the previous trained model and activated by ReLU. The filter slide over the input data as shown in the figure 5, and then compute the inner product of the filter and the input data at any position in RNA sequence. At each position, an inner product is generated, also known as an intermediate value A_i. There are the *threshold* set for the filter, which is 0.5 times of the maximum A_i. When A_i exceeds the *threshold*, the short sequence or structure can be considered valid motif. When short sequence length is 10 bp, each RNA sequence with 519 nucleotides can be divided into short sequences with number of 498.

$$threshold = 0.5 * \max_{i \in [0,497]} A_i \tag{7}$$

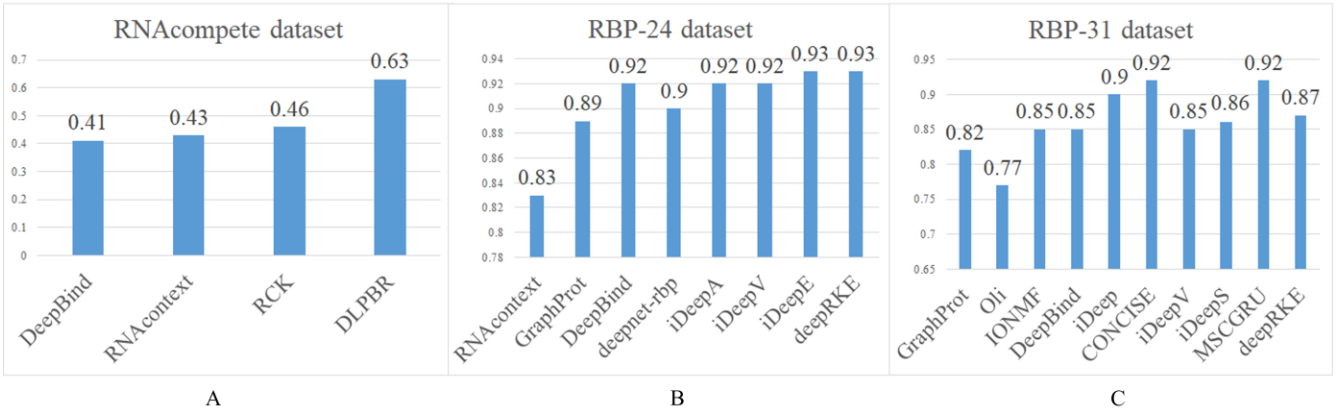

FIGURE 4. (A) shows AUC on RNAcompete dataset, (B) shows AUC on RBP-24 dataset, (C) shows AUC on RBP-31 dataset. The Y-axis of (A), (B), and (C) represents the value of AUC. The model used in the research with higher AUC is also recognized as an advanced model in the field of image processing, such as CNN, residual network.

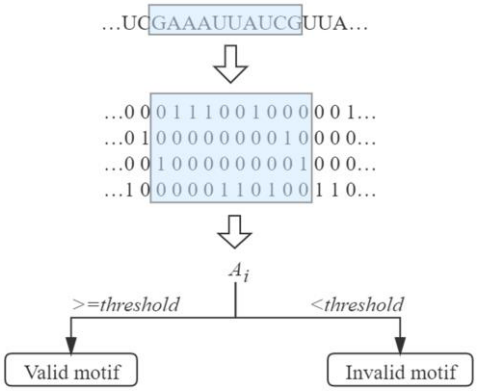

FIGURE 5. The filter of trained convolutional neural network slide over the input data, and then compute the inner product of the filter and the input data also called A_i at any position in RNA sequence. There are the threshold set for the filter, which is 0.5 times of the maximum A_i. When A_i exceeds the threshold, the short sequence or structure can be considered valid motif. Otherwise, it is considered invalid motif.

where i represents the first nucleotide position of the short sequence in RNA sequence. Previously, some sequences are filled with 'N', so short sequences containing 'N' are eliminated. We preserve short sequence containing only residues of A, U, G, and C. since the motif length is 7 bp [13] generally, only the first 7 bp of valid motif are intercepted. Usually, there will be a group of filters, operates as described above, so we can get a group of motifs. A group of motifs is used for separate analysis. We can also splice motifs together to get a set of short sequences, which can be used to extract common features [84].

2) VISUALIZING BINDING MOTIFS

We can perform sequence alignment on a set of short sequence using tools, such as WebLogo 3 [83], [85], MEME [14], and motifStack [86]. Simultaneously, sequence identity is counted at each position in the sequence. The sequence logo plots the statistical results of residues appearing at each position in a graphical way. The accumulation of residues at each position can reflect the consistency of residues at that position. The currently identified motifs related to RBPs can be obtained from the CISBP-RNA database [13]. The size of each base pattern is proportional to the frequency of bases appearing at this position. For each position, the height of the vertical axis of each base pattern is calculated as follows.

$$Height = f_i * R \tag{8}$$

where i is residue of A, C, G, and U. f_i represents the frequency of occurrence of base, and R is calculated as follows.

$$R = \log_2(m) - (entropy + e_n) \tag{9}$$

$$entropy = -\sum_i f_i * \log_2(f_i) \tag{10}$$

$$e_n = (m-1)/(2 * \ln 2 * n) \tag{11}$$

where *entropy* denotes the total entropy of the position, and m denotes the number of base species. For the protein, m is equal to 20. For the RNA studied in this paper, m is equal to 4. e_n is small sample test modification, where n represents the number of sequences.

E. VERFICATION AND EVALUATION METHOD OF MOTIFS

Facts proved that filter of CNN is applied to automatically capture known sequence motifs and structure motifs. In order to verify the detected motifs, they are compared with the known motifs from CISBP-RNA database [13] and literature. How to evaluate the quality of extracted motif? Here are some evaluation indicators, such as E values and P values, etc. Tomtom [85] introduced a statistical standard E values of similarity between motifs. The accuracy of Tomtom's E values was demonstrated to be effectiveness in finding similar motifs. The P value is used to estimate the enrichment score using AME [87] in the MEME suite [14] by scanning the predicted motif for the input sequence and the corresponding

shuffle sequence. The P value is calculated by the cumulative density function estimated from the target database. The minimum P value of these offset P values is used to calculate the match of the overall P value between the query motif and the target motif. If the P value is independent, it is called the motif P value.

V. RBP PREDICTION PROCESS AND ATTENTION PROBLEMS IN TRAINING

A. ENCODING RNA SEQUENCES AND STRUCTURES

The linear arrangement of four bases in RNA constitutes the primary structure of RNA. Due to the existence of base pairing (A-U, G-C), RNA secondary structure is formed by single strand refolding. Based on the spatial constraints of geometry and steric effect, the tertiary structure of RNA is formed, i.e., the position of RNA atoms in three-dimensional space, involving covalent bond, hydrogen bond, electrostatic force, van der Waals force, etc. The quaternary structure of RNA is the interaction between RNA and other small molecules. In this paper, the research theme is quaternary structure. At present, it is difficult to obtain the tertiary structure and its prediction method is not mature. On the contrary, the prediction results of the secondary structure level have been generally accepted. Therefore, in this section, we introduce the encoding of sequence and secondary structure.

1) SEQUENCES ENCODING

One-hot encoding is the basic method for converting RNA sequence into vectors. Therefore, we must vectorize the RNA sequence into binary matrix. During data preprocessing, all RNA sequences need to be cut to a fixed length. If the sequence exceeds the fixed length, it will be trimmed. If the sequence length is insufficient, use 'N' padding to the fixed length. As is shown in the figure 6.A, each site could be represented by a four states vector corresponding to bases of A, C, G, and U. Here, an RNA sequence named $S = (s_1, s_2, \ldots, s_i, \ldots, s_n)$ with n nucleotides and sequence motif length with m, the binary matrix S for RNA sequence is represented by a $4*n$ dimensional column vector as follows.

$$S_{ij} = \begin{cases} 0.25, & if\ s_{i-m+1} =' N'\ or\ i < m\ or\ i > (n-m) \\ 1, & if\ s_{i-m+1}\ is\ (A, U, C, G) \\ 0, & \text{otherwise} \end{cases} \quad (12)$$

where i is the index of the nucleotide, j is the index of the column corresponding to A, C, G, U.

RNA sequences are usually represented by k-mer encoding. In RNA sequences, k-mer is often nonrandom, and exists certain rules. The k-mer contains k nucleotides, which is a subsequence of sequence. The frequency information of k-mer is often used in k-mer coding. The subsequence space contains all possible motif of length k. Each possible motif is arranged by selecting k elements repeatedly from A, C, G, and U. We map each subsequence of k-mer into a high-dimensional vector, as is shown in the figure 6.B, assuming k is 3. Therefore, the subsequence space size is 4^k. The frequency of each subsequence in the sequence

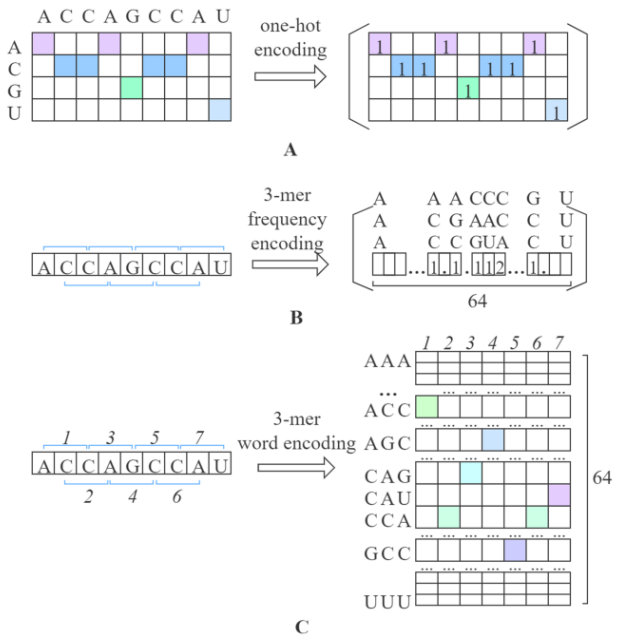

FIGURE 6. (A) RNA sequences are converted into vectors by one-hot encoding. Because there are four bases, the matrix is 4-dimensional. (B). k-mer frequency encoding is 64-dimensional, where k is set to 3. (C). k-mer word encoding is constructed by slide window over on the sequence, where k is set to 3.

is calculated. The dimension is determined by k. Here, we call it k-mer frequency encoding.

In addition, k-mer can be regarded as a word and RNA sequence as a sentence. Split the sentence into words and convert each word into a vector. The properties of k-mer can reflect the relationship between the functions and different sequence regions. Put all the k-mers in order without calculating the frequency, as is shown in the figure 6.C. The dimension of this method is mainly determined by the length of RNA sequence, so we call it k-mer word encoding.

2) STRUCTURES ENCODING

In general, annotation of RNA secondary structure could be reflected by the pairing state or pairing probability at each site. At present, prediction methods for RNA secondary structure include SFOLD [88], RNAshapes [71], RNAplfold [76], SCFG [89], etc. The structural information was introduced according to the following annotation, such as stems (S), multiloops (M), hairpins (H), internal loops (I), dangling end (T) and dangling start (F). In the same way as the sequence information, the structural sequence containing 'SMHITF' is encoded one-hot or k-mer. Given an RNA sequence $S = (s_1, s_2, \ldots, s_i, \ldots, s_n)$ with n nucleotides and motif length with m, we can obtain secondary structure matrix $T = (t_1, t_2, \ldots, t_i, \ldots, t_n)$ from prediction methods, details as follows.

$$T_{ij} = \begin{cases} 0.16, & if\ i < m\ or\ i > (n-m) \\ 1, & if\ t_{i-m+1}\ is\ (S, M, H, I, T, F) \\ 0, & \text{otherwise} \end{cases} \quad (13)$$

where i is the index of the nucleotide, j is the index of the column corresponding to S, M, H, I, T, and F.

B. VALIDATION METRICS

For balanced classification (each category has the same probability), the main evaluation criteria are Accuracy, Precision, Recall and f-score, which can be easily calculated by the confusion matrix. In addition, there is another commonly used indicator, ROC Curve, which quantifies the performance of a classifier by calculating the value of the AUC. Most of the models mentioned in this paper belong to two balanced classification. For the imbalance problem, Accuracy and Recall can be used. For sorting problems or multi-label classifications, the mean average precision could be used. They are calculated by the following formulas.

$$Precision = \frac{TP}{TP + FP} \tag{14}$$

$$Recall = \frac{TP}{TP + FN} \tag{15}$$

where P is the number of positives, and N is the number of negatives. TP is the number of true positives. TN is the number of true negatives. FP is the number of false positives. FN is the number of false negatives. Precision and Recall two evaluation criteria are not functionally related. Neither of them can reflect the effect of the classifier alone. Therefore, the harmonic average F-score of the two is used as follows.

$$F - score = \frac{2}{\frac{1}{Precision} + \frac{1}{Recall}} \tag{16}$$

In addition, Accuracy is utilized to measure the overall ability of the model sometimes to distinguish between positive and negative samples. The formula is as follows.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{17}$$

The statistics of confusion matrix needs to set a threshold to convert probability into binary, which is usually set manually. In this paper, we generally choose 0.5 as the threshold, that is, the probability of classification results is more than 0.5, which is considered as a positive sample, otherwise it is a negative sample. Setting different threshold can produce different confusion matrix and get different evaluation scores. Therefore, in order to avoid the interference caused by artificial threshold setting, ROC curve and AUC are used as evaluation indexes. ROC curve is generated based on the real category and prediction probability of samples. Specifically, ROC curve is a curve of true positive rate (TPR) to false positive rate (FPR). Here, X axis is TPR, Y axis is FPR. TPR and FPR are calculated as follows.

$$TPR = Recall = \frac{TP}{TP + FN} \tag{18}$$

$$FPR = \frac{FP}{FP + TN} \tag{19}$$

where AUC is the area under the ROC curve. The higher AUC value is, the better performance is.

C. OVERFITTING, MODEL REGULARIZATION AND ADJUSTING SUPER PARAMETERS

Focusing on evaluation of model is to divide the data into three parts, including training set, verification set and test set. We can train the model on the training data and evaluate the model on the verification data. It is necessary to determine how to measure current progress in training. Currently, there are three common evaluation methods, leave-out cross validation, K-fold cross validation, and repeated K-fold verification.

In addition, it is also important to take into consideration the overfitting of model and adjustment of super parameters. The ideal model is just on the boundary between underfitting and overfitting. At the beginning of training, optimization and generalization are related: the smaller the loss on the training set, the smaller the loss on the test dataset, and the model is underfitting. It can still be improved here. The network has not yet modeled all the relevant patterns in the training data. However, after certain iterations, the generalization of training data is no longer improved. During this process, the evaluation index is unchanged first, and then it begins to get worse, that is, the model begins to overfit. At this time, the model begins to learn the pattern only related to training data. However, this pattern is wrong or irrelevant for the new data.

The more training data, the better the generalization ability. But more data is not available usually, another solution is to adjust or restrict the amount of information that the model allows to store. The easiest way to prevent overfitting is to reduce the model size, such as the number of layers and the number of units per layer. Intuitively, a model with more parameters has a larger memory capacity. Find a compromise between large capacity and insufficient capacity. A simple model can avoid overfitting, and refers to a model with a smaller entropy of parameter value distribution or a model with smaller parameters.

Another common method to reduce over fitting is to limit the model weight to a smaller value, so as to reduce the complexity of the model. At this time, the distribution of weights is more regular, called weight regularization. The implementation method is to add the cost associated with a larger weight value to the loss function. This cost has two forms: L1 regularization and L2 regularization. L1 regularization means that the added cost is proportional to the absolute value of the weight coefficient. L2 regularization means that the added cost is proportional to the square of the weight coefficient, and also called weight decay. Dropout [90] is also one of the most effective and commonly used regularization methods. To use dropout for a layer is to randomly discard some output features of the layer (i.e., set to 0) during the training. The dropout ratio is the proportion of features set to 0, usually in the range of 0.2 to 0.5. In addition, you can try different hyperparameters, such as the learning rate or batch size, to find the best configuration. Adjustment of hyperparameters is time-consuming. We constantly adjust model training, evaluate on validation data, and adjust the

model again. Finally, it is optional to do feature engineering repeatedly to add new features or delete features without information.

VI. DISCUSSION

In this paper, we review the development of deep learning techniques and some of the-state-of-art applications in this field. Firstly, we introduce the related datasets. Next, we describe development of deep learning and recent studies that have exploited deep learning models. Finally, we conclude this article by summarizing research trends and suggesting directions for further improvements. Although deep learning algorithm has improved the performance of classification and become a promising approach, there are still significant challenges for its applications in biology and medicine data analysis.

A. APPROPRIATE EVALUATION CRITERIA FOR UNBALANCED DATA

RBPs data are usually unbalanced because the process of data acquisition is usually complex and expensive. In fact, the number of negative samples in RBP is much larger than that of positive samples. The standard performance criteria used in the data training process are often biased toward the majority class. Therefore, we have to balance the positive and negative samples in training. This will lose a lot of negative sample information because the number of negative samples is much larger than positive samples, leading to only a part of the negative data being sampled. We can also train all RBPs to only one model, and the number of each RBP is not equal and balanced. When assessing the model building on the imbalance data, more metrics have to be taken into consideration. It may bury over accuracy or recall if consider only ROC curve.

B. PENDING DATA QUALITY

According to RBP specific model, the model fails in some RBPs where other existing tools also have low AUC values. Although high-throughput experiments have been developed in advance, the collected data still suffer from the false-positive and false-negative problems due to the experimental noises and current limitations. High false positive rates and false negative rates may lead to poor prediction. Deep learning model needs to be trained and optimized through a large number of datasets with high quality. Thus, more studies are required to further improve the data quality.

C. COMBINING MORE DEEP LEARNING MODELS INTO PREDICTION OF RBPS

With the development of deep learning, more and more deep learning models are proposed. Residual networks are also used for RBP prediction due to their great performance. For example, object detection [91] is divided into three steps, including classification, object detection and semantic segmentation. Studies have shown that 75% RNAs react with at least two or more RBPS. We take the type of RBP as the label, so the prediction problem is a multi-label problem. When predicting RBP, the corresponding learned motif is used as a feature for RBP classification in the stage of object detection. The image perceiving features is like the RNA sequence perceiving motif. Ideally, motif can be detected directly in RNA sequence in the stage of semantic segmentation. It will be an interesting study. At present, there are many trained migration models for us to use. Ideal configuration of hyperparameters depends on the data and application through grid search. At this point, we can migrate the trained model, which can effectively explore different layers or values, while keeping all other layers or super parameters in the model unchanged. Deep learning likes a ''black box''. In the future work, we tend to investigate interpretability of models, which is a hot research topic in deep learning and also future direction for RBP prediction.

REFERENCES

[1] J. D. Keene, ''RNA regulons: Coordination of post-transcriptional events,'' *Nature Rev. Genet.*, vol. 8, no. 7, pp. 533–543, Jul. 2007.
[2] A. R. Buxbaum, G. Haimovich, and R. H. Singer, ''In the right place at the right time: Visualizing and understanding mRNA localization,'' *Nature Rev. Mol. Cell Biol.*, vol. 16, no. 2, pp. 95–109, Feb. 2015.
[3] I. L. de Silanes, N. Olmo, J. Turnay, G. G. de Buitrago, P. Pérez-Ramos, A. Guzmán-Aránguez, M. García-Díez, E. Lecona, M. Gorospe, and M. A. Lizarbe, ''Acquisition of resistance to butyrate enhances survival after stress and induces malignancy of human colon carcinoma cells,'' *Cancer Res.*, vol. 64, no. 13, pp. 4593–4600, Jul. 2004.
[4] H. S. Kim, M. C. J. Wilce, Y. M. K. Yoga, N. R. Pendini, M. J. Gunzburg, N. P. Cowieson, G. M. Wilson, B. R. G. Williams, M. Gorospe, andJ. A. Wilce, ''Different modes of interaction by TIAR and HuR with target RNA and DNA,'' *Nucleic Acids Res.*, vol. 39, no. 3, pp. 1117–1130, Feb. 2011.
[5] M. M. Scotti and M. S. Swanson, ''RNA mis-splicing in disease,'' *Nature Rev. Genet.*, vol. 17, no. 1, pp. 19–32, Jan. 2016.
[6] X. Pan, Y. Yang, C. Xia, A. H. Mirza, and H. Shen, ''Recent methodology progress of deep learning for RNA–protein interaction prediction,'' *Wiley Interdiscipl. Rev., RNA*, vol. 10, no. 6, Nov. 2019, Art. no. e1544.
[7] A. Castello, B. Fischer, K. Eichelbaum, R. Horos, B. M. Beckmann, C. Strein, N. E. Davey, D. T. Humphreys, T. Preiss, L. M. Steinmetz, J. Krijgsveld, and M. W. Hentze, ''Insights into RNA biology from an atlas of mammalian mRNA-binding proteins,'' *Cell*, vol. 149, no. 6, pp. 1393–1406, Jun. 2012.
[8] M. W. Hentze, A. Castello, T. Schwarzl, and T. Preiss, ''A brave new world of RNA-binding proteins,'' *Nature Rev. Mol. Cell Biol.*, vol. 19, no. 5, pp. 327–341, May 2018.
[9] A. E. Dahlberg, C. W. Dingman, and A. C. Peacock, ''Electrophoretic characterization of bacterial polyribosomes in agarose-acrylamide composite gels,'' *J. Mol. Biol.*, vol. 41, no. 1, pp. 139–147, Apr. 1969.
[10] J. Czworkowski, O. W. Odom, and B. Hardesty, ''Fluorescence study of the topology of messenger RNA bound to the 30S ribosomal subunit of escherichia coli,'' *Biochemistry*, vol. 30, no. 19, pp. 4821–4830, May 1991.
[11] J. König, K. Zarnack, G. Rot, T. Curk, M. Kayikci, B. Zupan, D. J. Turner, N. M. Luscombe, and J. Ule, ''ICLIP reveals the function of hnRNP particles in splicing at individual nucleotide resolution,'' *Nature Struct. Mol. Biol.*, vol. 17, no. 7, pp. 909–915, Jul. 2010.
[12] A. D. Ellington and J. W. Szostak, ''*In vitro* selection of RNA molecules that bind specific ligands,'' *Nature*, vol. 346, no. 6287, pp. 818–822, Aug. 1990.
[13] D. Ray *et al.*, ''A compendium of RNA-binding motifs for decoding gene regulation,'' *Nature*, vol. 499, no. 7457, pp. 172–177, Jul. 2013.
[14] T. L. Bailey, M. Boden, F. A. Buske, M. Frith, C. E. Grant, L. Clementi, J. Ren, W. W. Li, and W. S. Noble, ''MEME SUITE: Tools for motif discovery and searching,'' *Nucleic Acids Res.*, vol. 37, pp. W202–W208, Jul. 2009.
[15] P. H. Reyes-Herrera and E. Ficarra, ''Computational methods for CLIP-seq data processing,'' (in Eng), *Bioinf. Biol. Insights*, vol. 8, pp. 199–207, Jan. 2014.

[16] B. C. Foat, A. V. Morozov, and H. J. Bussemaker, "Statistical mechanical modeling of genome-wide transcription factor occupancy data by MatrixREDUCE," *Bioinformatics*, vol. 22, no. 14, pp. e141–e149, Jul. 2006.

[17] H. Kazan, D. Ray, E. T. Chan, T. R. Hughes, and Q. Morris, "RNAcontext: A new method for learning the sequence and structure binding preferences of RNA-binding proteins," *PLoS Comput. Biol.*, vol. 6, no. 7, Jul. 2010, Art. no. e1000832.

[18] D. Maticzka, S. J. Lange, F. Costa, and R. Backofen, "GraphProt: Modeling binding preferences of RNA-binding proteins," *Genome Biol.*, vol. 15, no. 1, p. R17, 2014.

[19] O. Yaron, Y. Wang, and B. Bonnie, "RCK: Accurate and efficient inference of sequence- and structure-based protein-RNA binding models from RNAcompete data," *Bioinformatics*, vol. 32, no. 12, pp. i351–i359, 2016.

[20] S. Li, F. Dong, Y. Wu, S. Zhang, C. Zhang, X. Liu, T. Jiang, and J. Zeng, "A deep boosting based approach for capturing the sequence binding preferences of RNA-binding proteins from high-throughput CLIP-seq data," *Nucleic Acids Res.*, vol. 45, no. 14, p. e129, Aug. 2017.

[21] Y. Bengio, "Learning deep architectures for AI," *Found. Trends Mach. Learn.*, vol. 2, no. 1, pp. 1–127, 2009.

[22] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[23] B. Alipanahi, A. Delong, M. T. Weirauch, and B. J. Frey, "Predicting the sequence specificities of DNA- and RNA-binding proteins by deep learning," *Nature Biotechnol.*, vol. 33, no. 8, pp. 831–838, Aug. 2015.

[24] X. Pan, P. Rijnbeek, J. Yan, and H.-B. Shen, "Prediction of RNA-protein sequence and structure binding preferences using deep convolutional and recurrent neural networks," *BMC Genomics*, vol. 19, no. 1, p. 511, Dec. 2018.

[25] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with LSTM," *Neural Comput.*, vol. 12, no. 10, pp. 2451–2471, Oct. 2000.

[26] X. Pan and H.-B. Shen, "RNA-protein binding motifs mining with a new hybrid deep learning based cross-domain knowledge integration approach," *BMC Bioinf.*, vol. 18, no. 1, p. 136, Dec. 2017.

[27] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng, "Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations," in *Proc. 26th Annu. Int. Conf. Mach. Learn. ICML*, 2009, pp. 609–616, doi: 10.1145/1553374.1553453.

[28] S. Zhang, J. Zhou, H. Hu, H. Gong, L. Chen, C. Cheng, and J. Zeng, "A deep learning framework for modeling structural features of RNA-binding protein targets," *Nucleic Acids Res.*, vol. 44, no. 4, p. e32, Feb. 2016.

[29] M. Uhl, V. D. Tran, F. Heyl, and R. Backofen, "GraphProt2: A novel deep learning-based method for predicting binding sites of RNA-binding proteins," *BioRxiv*, 2019, Art. no. 850024, doi: 10.1101/850024.

[30] J. L. Thorvaldsen, J. R. Weaver, and M. S. Bartolomei, "A YY1 bridge for X inactivation," *Cell*, vol. 146, no. 1, pp. 1–13, 2011.

[31] D. Ray, H. Kazan, E. T. Chan, L. P. Castillo, S. Chaudhry, S. Talukder, B. J. Blencowe, Q. Morris, and T. R. Hughes, "Rapid and systematic analysis of the RNA recognition specificities of RNA-binding proteins," *Nature Biotechnol.*, vol. 27, no. 7, pp. 667–670, Jul. 2009.

[32] M. Stražar, M. Žitnik, B. Zupan, J. Ule, and T. Curk, "Orthogonal matrix factorization enables integrative analysis of multiple RNA binding proteins," *Bioinformatics*, vol. 32, no. 10, pp. 1527–1535, May 2016.

[33] G. Corrado, T. Tebaldi, F. Costa, P. Frasconi, and A. Passerini, "RNAcommender: Genome-wide recommendation of RNA-protein interactions," *Bioinformatics*, vol. 32, no. 23, pp. 3627–3634, 2016.

[34] I. Ben-Bassat, B. Chor, and Y. Orenstein, "A deep neural network approach for learning intrinsic protein-RNA binding preferences," *Bioinformatics*, vol. 34, no. 17, pp. i638–i646, Sep. 2018.

[35] G. Anders, S. D. Mackowiak, M. Jens, J. Maaskola, A. Kuntzagk, N. Rajewsky, M. Landthaler, and C. Dieterich, "DoRiNA: A database of RNA interactions in post-transcriptional regulation," *Nucleic Acids Res.*, vol. 40, no. D1, pp. D180–D186, Jan. 2012.

[36] Y. Xue, Y. Zhou, T. Wu, T. Zhu, X. Ji, Y.-S. Kwon, C. Zhang, G. Yeo, D. L. Black, H. Sun, X.-D. Fu, and Y. Zhang, "Genome-wide analysis of PTB-RNA interactions reveals a strategy used by the general splicing repressor to modulate exon inclusion or skipping," *Mol. Cell*, vol. 36, no. 6, pp. 996–1006, Dec. 2009.

[37] A. R. Quinlan and I. M. Hall, "BEDTools: A flexible suite of utilities for comparing genomic features," *Bioinformatics*, vol. 26, no. 6, pp. 841–842, Mar. 2010.

[38] E. Dassi, A. Re, S. Leo, T. Tebaldi, L. Pasini, D. Peroni, and A. Quattrone, "AURA 2: Empowering discovery of post-transcriptional networks," *Translation*, vol. 2, no. 1, Jan. 2014, Art. no. e27738.

[39] X. Pan, Y.-X. Fan, J. Jia, and H.-B. Shen, "Identifying RNA-binding proteins using multi-label deep learning," *Sci. China Inf. Sci.*, vol. 62, no. 1, pp. 213–215, Jan. 2019.

[40] Y. Huang, B. Niu, Y. Gao, L. Fu, and W. Li, "CD-HIT Suite: A Web server for clustering and comparing biological sequences," *Bioinformatics*, vol. 26, no. 5, pp. 680–682, 2010.

[41] C. Zhu, Y. Chen, Y. Zhang, S. Liu, and G. Li, "ResGAN: A low-level image processing network to restore original quality of JPEG compressed images," in *Proc. Data Compress. Conf. (DCC)*, Mar. 2019, p. 616.

[42] A. Rosenberg, Y. Zhang, B. Ramabhadran, Y. Jia, P. Moreno, Y. Wu, and Z. Wu, "Speech recognition with augmented synthesized speech," in *Proc. IEEE Autom. Speech Recognit. Understand. Workshop (ASRU)*, Dec. 2019, pp. 996–1002.

[43] M. Lukoševičius and H. Jaeger, "Reservoir computing approaches to recurrent neural network training," *Comput. Sci. Rev.*, vol. 3, pp. 127–149, Aug. 2009.

[44] G.-P. Liao, W. Gao, G.-J. Yang, and M.-F. Guo, "Hydroelectric generating unit fault diagnosis using 1-D convolutional neural network and gated recurrent unit in small hydro," *IEEE Sensors J.*, vol. 19, no. 20, pp. 9352–9363, Oct. 2019.

[45] H.-C. Shin, H. R. Roth, M. Gao, L. Lu, Z. Xu, I. Nogues, J. Yao, D. Mollura, and R. M. Summers, "Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning," *IEEE Trans. Med. Imag.*, vol. 35, no. 5, pp. 1285–1298, May 2016.

[46] X. Pan and H.-B. Shen, "Predicting RNA–protein binding sites and motifs through combining local and global deep convolutional neural networks," *Bioinformatics*, vol. 34, no. 20, pp. 3427–3436, Oct. 2018.

[47] S. Karita, N. Chen, T. Hayashi, T. Hori, H. Inaguma, Z. Jiang, M. Someki, N. E. Y. Soplin, R. Yamamoto, X. Wang, S. Watanabe, T. Yoshimura, and W. Zhang, "A comparative study on transformer vs RNN in speech applications," in *Proc. IEEE Autom. Speech Recognit. Understand. Workshop (ASRU)*, Dec. 2019, pp. 449–456.

[48] D. Li and J. Qian, "Text sentiment analysis based on long short-term memory," in *Proc. 1st IEEE Int. Conf. Comput. Commun. Internet (ICCCI)*, Oct. 2016, pp. 471–475.

[49] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2016, *arXiv:1609.02907*. [Online]. Available: http://arxiv.org/abs/1609.02907

[50] X. Guo, H. Zhang, H. Yang, L. Xu, and Z. Ye, "A single attention-based combination of CNN and RNN for relation classification," *IEEE Access*, vol. 7, pp. 12467–12475, 2019.

[51] Q. Du, W. Gu, L. Zhang, and S.-L. Huang, "Attention-based LSTM-CNNs for time-series classification," in *Proc. 16th ACM Conf. Embedded Networked Sensor Syst.*, Nov. 2018, pp. 410–411, doi: 10.1145/3274783.3275208.

[52] Z. S. Ritu, N. Nowshin, M. M. H. Nahid, and S. Ismail, "Performance analysis of different word embedding models on bangla language," in *Proc. Int. Conf. Bangla Speech Lang. Process. (ICBSLP)*, Sep. 2018, pp. 1–5.

[53] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013, *arXiv:1301.3781*. [Online]. Available: https://arxiv.org/abs/1301.3781

[54] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin, "A neural probabilistic language model," *J. Mach. Learn. Res.*, vol. 3, pp. 1137–1155, Feb. 2003.

[55] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 1532–1543.

[56] E. Asgari and M. R. K. Mofrad, "Continuous distributed representation of biological sequences for deep proteomics and genomics," *PLoS ONE*, vol. 10, no. 11, Nov. 2015, Art. no. e0141287.

[57] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," 2014, *arXiv:1409.0473*. [Online]. Available: https://arxiv.org/abs/1409.0473

[58] Z. Zheng, X. Yang, Z. Yu, L. Zheng, Y. Yang, and J. Kautz, "Joint discriminative and generative learning for person re-identification," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 2138–2147.

[59] N. T. Ngo, T. N. Nguyen, and T. H. Nguyen, "Learning to select important context words for event detection," in *Advances in Knowledge Discovery and Data Mining*. Cham, Switzerland: Springer, 2020, pp. 756–768.

[60] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A, Courville, and Y. Bengio, "Generative adversarial nets," presented at the Neural Inf. Process. Syst., Aug. 2014.
[61] M. Zhao, X. Liu, H. Liu, and K. K. L. Wong, "Super-resolution of cardiac magnetic resonance images using Laplacian pyramid based on generative adversarial networks," *Computerized Med. Imag. Graph.*, vol. 80, Mar. 2020, Art. no. 101698.
[62] S. Ghannay, Y. Estève, and N. Camelin, "A study of continuous space word and sentence representations applied to ASR error detection," *Speech Commun.*, vol. 120, pp. 31–41, Jun. 2020.
[63] R. S. Sutton, D. A. McAllester, S. P. Singh, and Y. Mansour, "Policy gradient methods for reinforcement learning with function approximation," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 12, 2000, pp. 1057–1063.
[64] K. Lin, D. Li, X. He, Z. Zhang, and M. T. Sun, "Adversarial Ranking for Language Generation," presented at the 31st Conf. Neural Inf. Process. Syst. (NIPS), 2017.
[65] L. Yu, W. Zhang, J. Wang, and Y. Yu, "SeqGAN: Sequence Generative Adversarial Nets with Policy Gradient," presented at the Proc. 21st AAAI Conf. Artif. Intell. (AAAI), 2017. [Online]. Available: https://www.aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14344/14489
[66] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
[67] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to Sequence Learning with Neural Networks," presented at the Adv. Neural Inf. Process. Syst. (NIPS), 2014.
[68] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," presented at the Adv. Neural Inf. Process. Syst., Dec. 2017.
[69] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," presented at the Proc. Conf. North American Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol., Nov. 2018.
[70] X. Pan and J. Yan, "Attention based convolutional neural network for predicting RNA-protein binding sites," 2017, *arXiv:1712.02270*. [Online]. Available: http://arxiv.org/abs/1712.02270
[71] P. Steffen, B. Voss, M. Rehmsmeier, J. Reeder, and R. Giegerich, "RNAshapes: An integrated RNA analysis package based on abstract shapes," *Bioinformatics*, vol. 22, no. 4, pp. 500–503, Feb. 2006.
[72] C. M. Livi and E. Blanzieri, "Protein-specific prediction of mRNA binding using RNA sequences, binding motifs and predicted secondary structures," *BMC Bioinf.*, vol. 15, no. 1, pp. 1–11, Dec. 2014.
[73] I. L. Hofacker, W. Fontana, P. F. Stadler, L. S. Bonhoeffer, M. Tacker, and P. Schuster, "Fast folding and comparison of RNA secondary structures," *Monatshefte Für Chem. Chem. Monthly*, vol. 125, no. 2, pp. 167–188, Feb. 1994.
[74] M. de Gemmis, P. Lops, C. Musto, F. Narducci, and G. Semeraro, "Semantics-aware content-based recommender systems," in *Recommender Systems Handbook*, F. Ricci, L. Rokach, and B. Shapira, Eds. Boston, MA, USA: Springer, 2015, pp. 119–159.
[75] Z. Shen, S.-P. Deng, and D.-S. Huang, "RNA-protein binding sites prediction via multi scale convolutional gated recurrent unit networks," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, early access, Apr. 19, 2019, doi: 10.1109/TCBB.2019.2910513.
[76] R. Lorenz, S. H. Bernhart, C. Höner zu Siederdissen, H. Tafer, C. Flamm, P. F. Stadler, and I. L. Hofacker, "ViennaRNA package 2.0," *Algorithms Mol. Biol.*, vol. 6, no. 1, p. 26, Dec. 2011.
[77] S.-K. Hong, M.-S. Gil, and Y.-S. Moon, "Secure computation of Pearson correlation coefficients for high-quality data analytics," in *Database Systems for Advanced Applications*. Cham, Switzerland: Springer, 2018, pp. 89–98.
[78] X. Pan and H.-B. Shen, "Learning distributed representations of RNA sequences and its application for predicting RNA-protein binding sites with a convolutional neural network," *Neurocomputing*, vol. 305, pp. 51–58, Aug. 2018.
[79] L. Deng, Y. Liu, Y. Shi, and H. Liu, "A deep neural network approach using distributed representations of RNA sequence and structure for identifying binding site of RNA-binding proteins," in *Proc. IEEE Int. Conf. Bioinf. Biomed. (BIBM)*, Nov. 2019, pp. 12–17.
[80] Ž. Avsec, M. Barekatain, J. Cheng, and J. Gagneur, "Modeling positional effects of regulatory sequences with spline transformations increases prediction accuracy of deep neural networks," *Bioinformatics*, vol. 34, no. 8, pp. 1261–1269, Apr. 2018.
[81] M. Ghanbari and U. Ohler, "Deep neural networks for interpreting RNA-binding protein target preferences," *Genome Res.*, vol. 30, no. 2, pp. 214–226, Feb. 2020.
[82] J. Chen, B. J. Aronow, and A. G. Jegga, "Disease candidate gene identification and prioritization using protein interaction networks," *BMC Bioinf.*, vol. 10, no. 1, p. 73, 2009.
[83] T. D. Schneider and S. R. Michael, "Sequence logos: A new way to display consensus sequences," *Nucleic Acids Res.*, vol. 18, no. 20, pp. 6097–6100, 1990.
[84] S. Gupta, J. A. Stamatoyannopoulos, T. L. Bailey, and W. Noble, "Quantifying similarity between motifs," *Genome Biol.*, vol. 8, no. 2, p. R24, 2007.
[85] G. E. Crooks, G. Hon, J. M. Chandonia, and S. E. Brenner, "WebLogo: A sequence logo generator," *Genome Res.*, vol. 14, no. 6, pp. 1188–1190, May 2004.
[86] J. Ou and L. J. Zhu, "Plot stacked logos for single or multiple DNA, RNA and amino acid sequence," *R Package Version*, vol. 18, no. 20, pp. 6097–6100, 2015.
[87] R. C. McLeay and T. L. Bailey, "Motif enrichment analysis: A unified framework and an evaluation on ChIP data," *BMC Bioinf.*, vol. 11, no. 1, p. 165, Dec. 2010.
[88] Y. Ding and C. E. Lawrence, "A statistical sampling algorithm for RNA secondary structure prediction," *Nucleic Acids Res.*, no. 24, p. 24, 2003.
[89] Y. Zhu, Z. Xie, Y. Li, M. Zhu, and Y.-P.-P. Chen, "Research on folding diversity in statistical learning methods for RNA secondary structure prediction," *Int. J. Biol. Sci.*, vol. 14, no. 8, pp. 872–882, 2018.
[90] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.
[91] G. Xu, X. Zhu, and N. Tapper, "Using convolutional neural networks incorporating hierarchical active learning for target-searching in large-scale remote sensing images," *Int. J. Remote Sens.*, vol. 41, no. 11, pp. 4057–4079, Jun. 2020.

JIANRONG YAN was born in November 1993. She is currently pursuing the master's degree with the College of Computer Science, Sichuan University. Her research interests include RNA binding protein sites prediction and deep learning.

MIN ZHU received the Ph.D. degree in applied mathematics from Sichuan University, in 2004. Since 1996, she has been with the College of Computer Science, Sichuan University. She was a Visiting Scholar with Case Western Reserve University, USA, from 2013 to 2014. She is currently a Professor with the College of Computer Science, Sichuan University. She has published more than 100 academic articles. Her current research interests include bioinformatics, information visualization, and visual analysis. She is a member of the CCF Bioinformatics Special Committee.