Chinese Journal of Electronics
Vol.32, No.4, July 2023

Deep Contextual Representation Learning for Identifying Essential Proteins via Integrating Multisource Protein Features

LI Weihua[1], LIU Wenyang[1], GUO Yanbu[2], WANG Bingyi[3], and QING Hua[2]

(1. *School of Information Science and Engineering, Yunnan University, Kunming 650500, China*)

(2. *College of Software Engineering, Zhengzhou University of Light Industry, Zhengzhou 450002, China*)

(3. *Institute of Highland Forest Science, Chinese Academy of Forestry, Kunming 650224, China*)

Abstract — Essential proteins with biological functions are necessary for the survival of organisms. Computational recognition methods of essential proteins can reduce the workload and provide candidate proteins for biologists. However, existing methods fail to efficiently identify essential proteins, and generally do not fully use amino acid sequence information to improve the performance of essential protein recognition. In this work, we propose an end-to-end deep contextual representation learning framework called DeepIEP to automatically learn biological discriminative features without prior knowledge based on protein network heterogeneous information. Specifically, the model attaches amino acid sequences as the attributes of each protein node in the protein interaction network, and then automatically learns topological features from protein interaction networks by graph embedding algorithms. Next, multi-scale convolutions and gated recurrent unit networks are used to extract contextual features from gene expression profiles. The extensive experiments confirm that our DeepIEP is an effective and efficient feature learning framework for identifying essential proteins and contextual features of protein sequences can improve the recognition performance of essential proteins.

Key words — Essential proteins, Protein interaction networks, Gene expression profile, Deep neural networks.

I. Introduction

Proteins are generally involved in the life activities of organisms and essential proteins are often found in protein complexes [1], [2]. Essential proteins are indispensable proteins for the survival and evolution of the organism cell [3], and their absence or dysfunction could cause some diseases and even lead to the inability of the body or organism to survive [4], [5]. From a practical perspective, essential proteins are closely related to pathogenic genes [6], thus the prediction of essential proteins is of great significance for the discovery of pathogenic genes. Therefore, accurately identifying essential proteins not only indicates the minimal requirements for the cell growth regulation mechanisms but also accelerates the discovery of disease genes and potential drug targets.

Biological experimental methods [3] could identify essential proteins accurately, such as RNA interference [7] and conditional knockouts [8], but however, they are time-consuming and expensive. Thus, developing computational algorithms is very important for identifying essential proteins. To date, many computational methods and centrality methods have been proposed successively to infer potential essential proteins. The research work of Jeong *et al.* [9] indicates that there is a positive correlation between the topological properties of proteins in protein-protein interaction (PPI) networks and protein essentiality. Subsequently, a lot of centrality methods were designed to identify essential proteins by the interconnectivity of proteins in PPI networks by topological features of PPI networks, such as degree centrality (DC) [10], betweenness centrality (BC) [11], closeness centrality (CC) [12], subgraph centrality (SC) [13], eigenvector centrality (EC) [14], and information centrality (IC) [15]. Gene expression profiles are useful

Manuscript Received Mar. 18, 2022; Accepted May 30, 2022. This work was supported by the National Natural Science Foundation of China (32060151, 62062066, 61762090) and the Doctor Scientific Research Fund of the Zhengzhou University of Light Industry (2021BSJJ032).

© 2023 Chinese Institute of Electronics. DOI:10.23919/cje.2022.00.053

to identify essential proteins [16], and proteins are some products of gene expressions. Moreover, the localization of proteins in cells is usually related to protein functions and most essential biological processes take place in certain subcellular localization [16]–[18], and the methods [19]–[21] were proposed to identify essential proteins by combining network topological features with different biological information.

With the rapid development of high-throughput sequencing techniques, many protein sequences and properties have been obtained, which make it possible to develop efficient machine learning algorithms for identifying essential proteins. Some machine learning-based recognition methods have been used for identifying essential proteins, such as local random walks [22], SVM [23], Naïve Bayes [19], and ensemble learning [24]. Although centrality and machine learning-based methods have obtained remarkable recognition results of essential proteins, they still have room for improvement. For centrality methods or machine learning-based algorithms, the biggest challenge is the feature representation of biological information [6], [16], such as network topology of the PPI network, gene expression, and subcellular localization. In centrality methods [10], [13], [25], researchers often designed the score function to represent the importance of each piece of biological information and then combined their functions into an equation to determine the essentiality of a protein. Although the methods have achieved good results, however, the centrality methods need a lot of prior knowledge to design the good score function, and does not characterize the comprehensive biological information.

In recent years, deep neural networks-based recognition models have achieved state-of-the-art performance for identifying essential proteins [6], [16], [26], [27], such as convolution neural networks (CNN), long short-term memory (LSTM), and multilayer perceptrons. The deep neural models are capable of fitting various signal data by the substantially increased depth and enlarged width of models, but they did not extract multi-scale contextual features. Besides, the effect of the amino acid sequence has been well studied in many protein-related prediction tasks, such as protein interactions [28], protein secondary structures [29], and essential proteins [16], [26]. Moreover, the amino acid sequence feature also affects the function and structure of the protein, and most proteins with the same function have relatively similar amino acid sequence features. Existing representation methods do not fully capture the high nonlinearity and preserve various proximities in both topological structure and node attributes of the PPI networks. Moreover, sophisticated hand-crafted features require massive prior knowledge, and they could not fully reflect intrinsic interaction patterns of protein sequences.

Oligomers of the length k (k-mer) are convenient and widely used as a feature encoding tool for modeling properties and functions of biological sequences [30]. The one-hot encoding method is a widely used feature representation tool [31], but feature vectors of k-mers are high-dimensional and sparse based on one-hot methods. Co-occurrence statistics of k-mers contain important information from protein sequences, and k-mer embedding can be used to represent contextual features of protein sequences. Besides, Node2Vec network embedding [32] has attracted much attention in scalable feature learning of node classification and link prediction for networks. However, protein nodes are often associated with rich attributes in protein biological systems, and deep attributed network embedding (DANE) [33] is an efficient feature learning algorithm that can capture the high nonlinearity and preserve various proximities in both topological structure and node attributes of networks. Thus, we adopt the DANE to automatically extract structure and attribute features based on amino acid sequence features and topological features of PPI networks and then map a protein into a continuous embedding simultaneously.

To tackle the above problems, we propose a deep neural model called DeepIEP to automatically learn biological features without prior knowledge based on graph embedding. Particularly, we used deep DANE [33] to map each protein of PPI networks into a low-dimensional embedding representation by fully leveraging the topological structure of PPI and node attribute information. Then, multi-scale convolutions and bidirectional gated recurrent unit networks (GRU) are applied to capture biological sequence dependency relationships in gene expression profile data. Briefly, the main contributions of this work are as follows:

1) A deep supervised learning framework is designed to automatically extract discriminative representation features based on three different types of biological information including PPI network, gene expression data, and protein sequence information from protein heterogeneous networks.

2) A deep attributed graph embedding is used to extract a low-dimensional graph embedding representation of each protein and can generate an informative representation of protein-protein interaction network topologies, where it combines topology and amino acid sequence information automatically.

3) Multi-scale convolutions are used to capture multiple spatial patterns from gene expression data and gated recurrent units are used to extract long-term dependency information. Next, the spatial patterns and

long-term dependency information are combined with graph embedding from the PPI network and protein sequences.

The remainder of this paper is organized as follows: Section II introduces related works. Section III explains the details of DeepIEP. Section IV presents the experiments and results. Section V gives the conclusion.

II. Related Works

The section summarizes related works based on computational algorithms for essential protein recognition.

1. Computational algorithm-based essential protein recognition

Existing computational methods mainly include centrality-based methods, traditional machine learning-based methods, and deep learning-based methods for identifying essential proteins. Centrality-based methods focus on topological features which are obtained from biological networks [8], [9], [11], [12], and the computation methods are sensitive to protein networks and missing data [19]. To improve the performance of essential protein recognition, centrality-based methods often integrate biological features into topological features. The biological features are related to sequence features obtained from genome or transcriptomics data, and functional features such as subcellular location or molecular functions. For example, Tang *et al.* [34] and Li *et al.* [35] generated features from gene expression profiles and fused them with network topology features to obtain the representation of essential proteins. Then, Li *et al.* [18] proposed a novel fusion method by integrating subcellular location, orthologous, and PPI, which is better than the above centrality methods. Zhu *et al.* [21] proposed an iterative model of multi-feature fusion to predict essential proteins by fusing biological and topological information of proteins, and Wang *et al.* [22] designed a novel method called RWAMVL to predict essential proteins based on the random walk and the adaptive multi-view label learning.

Traditional machine learning-based methods usually integrate multiple information for identifying essential proteins. For example, Plaimas *et al.* [36] integrated network topological features and gene expression information and used support vector machines (SVM) to identify essential proteins. Huang *et al.* [37] integrated network topology features and sequence information, and used SVM as classifiers. Zhong *et al.* [20] used gene expression information as biological attributes of proteins via different topological centrality methods, and then used random forest (RF), decision trees (DT), and SVM as the classifier. The methods integrate topological features and biological information to reduce the influence of noise data and topological networks and then improve the identification performance of essential proteins. However, they require prior knowledge and complex feature engineering. Besides, deep neural networks have a strong feature learning ability to model nonlinear relationships and can integrate various heterogeneous data, and they have been widely used in the field of biological information processing [38]. Recently, some deep learning-based methods have been proposed for the essential protein recognition task. For example, Zeng *et al.* [16] proposed a remarkable deep recurrent model called DeepEP-LSTM for identifying essential proteins based on CNN, long short-term memory (LSTM), and node2vec [32], and the successful performance further verified advantages of deep models for identifying essential proteins. What is more, deep neural networks require little prior knowledge and no laborious feature engineering, but the deep models do not fully exploit the information of the PPI networks.

2. Deep neural networks-based essential protein recognition

Due to the powerful representation ability, deep CNN-based models have been widely used in computer vision and computational biology [16], [27], [30], [39]–[41], and have obtained some great breakthroughs. However, the further improvement of their performance typically faces the following challenges: CNN-based models are hard to train and their intrinsic properties are sensitive to the input; due to the limited representation power of a single convolution, it largely neglects the multi-scale and long-term dependency patterns. The gradients of the loss function gradually decrease or even disappear after flowing through many layers, where the trainable parameters (i.e., the layers close to the input layer) cannot be optimized effectively.

Besides, with the powerful abilities to capture long-range dependencies, recurrent neural networks (RNN) have been widely applied in premature ventricular contraction (PVC) detection [42] and heart sound classification [43]. For example, Alkhodari *et al.* [43] developed a deep LSTM model to extract spatial and temporal features from the sound recordings and conduct the classification of heart sounds. Then, Wang *et al.* [42] designed an improved deep gated recurrent unit model by introducing a scale parameter into bidirectional GRU for PVC signals recognition, and the model can alleviate the problem of information redundancy. LSTM and GRU were two popular types of RNN, and they all used the gate mechanism to control how much the previous information is combined with the current input from the raw data. However, RNN-based deep models need more training time and computational power than convolutional operations during feature learning, and traditional RNN could not be computed in parallel for various tasks. Compared to LSTM, GRU could achieve comparable performance with less parameters. Therefore, we first exploit multi-scale convolutions to extract multiple spatial features of gene expression data in parallel. Then, bidirectional GRU is used to extract long-term dependencies from gene expression data based on spatial features of multi-scale convolutions.

III. Method

This section first introduces the architecture of DeepIEP, DANE, CGRU, and feature fusion components, respectively. Then, parameter learning is further introduced. Table 1 shows the main notations and descriptions.

1. Architecture of DeepIEP

As shown in Fig.1, the sequence of the protein is attached to the corresponding node of PPI networks as attribute information, and then the amino acids are fed into DeepIEP. The input of DeepIEP includes the PPI network, protein sequences, and gene expression profiles. The feature extraction part is composed of the DANE component and convolutional gated recurrent unit (CGRU) component. The deep attributed network embedding (DANE) component automatically learns

Table 1. The main symbols and descriptions

Symbols	Descriptions
H	The discriminative features extracted by DeepIEP
h_i	The hidden representation encoded by the encoder
H_{ST}	The spatial temporal interaction features
ζ	The loss function
H_{TF}	The topological information features
P_{ij}	The joint distribution between the nodes
Z	The attribute matrix of the protein
$\sigma(\cdot)$	A nonlinear function
E	The adjacency matrix of edges in the PPI network
Θ	All learnable parameters of DANE
x_i	The i-th input protein data

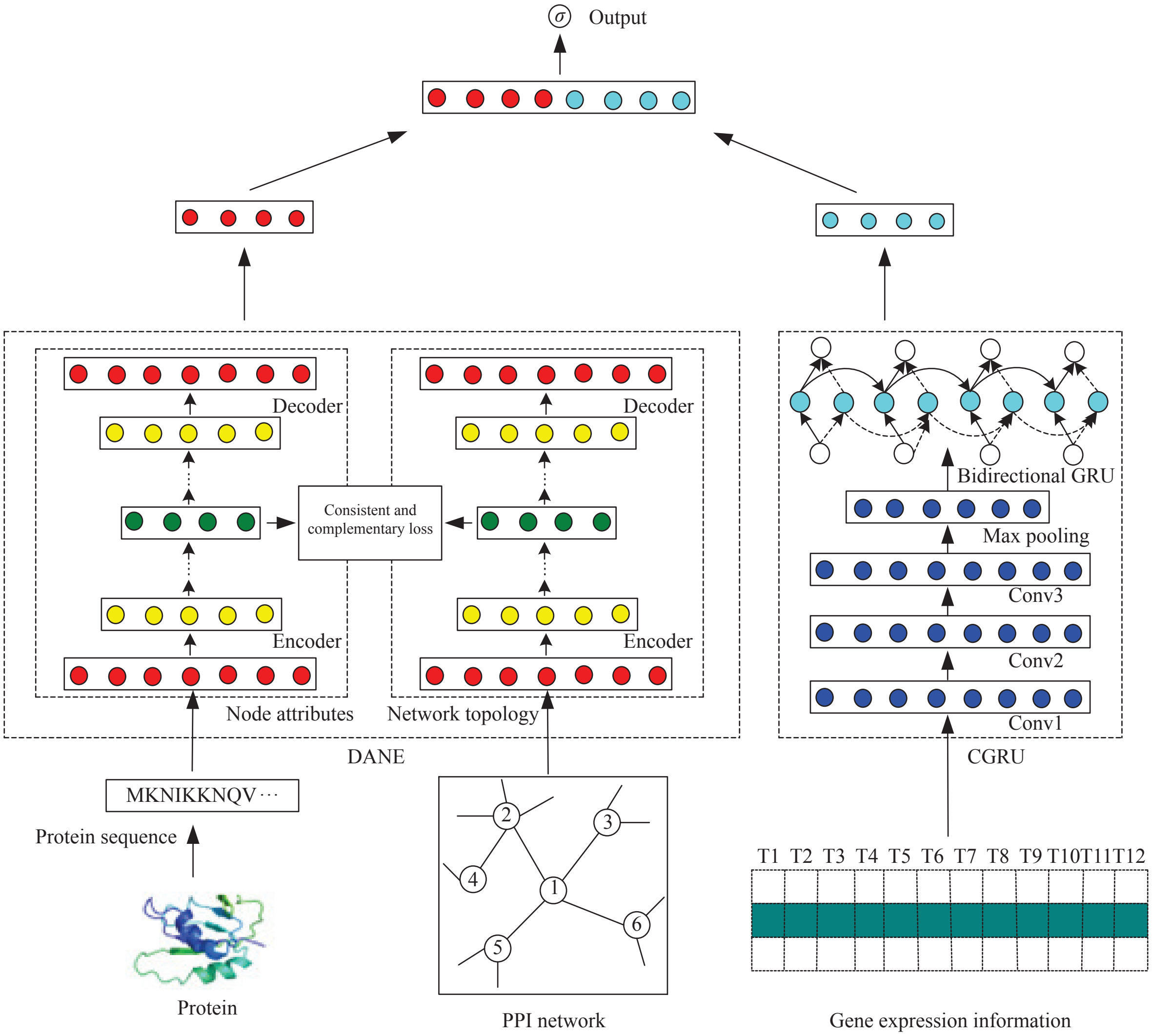


Fig. 1. An architecture of DeepIEP. The input of DeepIEP includes the PPI network, protein sequences, and gene expression profiles.

contextual features based on the PPI network and maps each node into a low-dimensional vector. CGRU extracts discriminative local and long-term dependency features from gene expression profiles. Particularly, the input of DeepIEP includes PPI networks, protein sequences, and gene expression profiles. The contextual representation learning part is composed of the DANE component and CGRU component. The DANE component is based on PPI networks attached protein sequence as the attribute of each node and maps each node with attributes to a low dimensional vector. Briefly, a deep attributed network embedding framework is used to capture the complex structure and attribute information of protein-protein interaction (PPI) networks with sequence features, and the CGRU component is composed of multi-scale convolutions and bidirectional GRU, which are used to extract informative features from gene expression profile. Next, the predictor combines spatial-temporal features to identify essential proteins.

The discriminative features extracted by DeepIEP are formulated as H from the PPI networks, gene expression data, and protein sequence information of protein heterogeneous networks:

$$\left\{\begin{array}{l} H=f_{FI}\left(H_{ST};H_{TF}\right) \\ H_{ST}=f_{GRU}\left(f_{MC}\left(I_{CE}\right)\right) \\ H_{TF}=f_{DANE}\left(I_{PPI}\right) \end{array}\right. \tag{1}$$

where H_{ST} represents spatial-temporal interaction features. H_{TF} represents topological information features of PPI networks and amino acid information. I_{CE} represents input features of the CGRU component. I_{PPI} represents input features of the DANE component. f_{FI} represents the non-linear transformation function of the feature fusion in the essential protein predictor. f_{MC} represents the non-linear transformation function of multi-scale convolutions. f_{GRU} represents the non-linear transformation function of bidirectional GRU. f_{DANE} represents the non-linear transformation function of the deep attributed graph embedding algorithm.

2. Deep attributed network embedding of proteins

The PPI network is a highly complex network, and contains a rich structure and attributes information about proteins. The researchers [6], [16] have indicated that the PPI network is useful to identify essential proteins. Specifically, the PPI network with n nodes can be represented as $G=(Z,E)$, where $Z=[Z_{ij}]\in^{n\times m}$ is the attribute matrix of the protein in the PPI network, and m is the length of node vectors. Z_i represents the attribute of the i-th protein from the attribute matrix. $E=[E_{ij}]\in^{n\times 2}$ is the adjacency matrix of edges in the PPI network. The amino acid sequence of a protein is first split into a k-mer sequence via a sliding window [44], where we divided each amino acid sequence into k-mer subsequences by using the window of length k with stride s. An k-mer embedding of the protein sequence is formulated as $Z_i=[s_1,s_2,\ldots,s_q]$, where q represents the length of the k-mer sequence.

The first-order proximity of two nodes i and j is determined by E_{ij}. A larger E_{ij} denotes larger proximity between the i-th node and the j-th one. Consequently, the high-order proximity of protein nodes i and j is determined by the similarity of $M_{i.}$ and $M_{j.}$, and $E=\ddot{E}+\ddot{E}^2+\cdots+\ddot{E}^t$ is the high-order proximity matrix of PPI networks, and $\ddot{E}$ is the 1-step probability transition matrix that is obtained from the row-wise normalization of the adjacency matrix E. The semantic proximity of node i and node j is determined by the similarity of $Z_{i.}$ and $Z_{j.}$. To fully extract the topological structure and node attributes of a protein, we adopted DANE [33] to learn the attributes and topological features of PPI networks, and DANE obtain a mapping of nodes f: $\{Z,E\}\rightarrow H$.

DANE is about to learn the low-dimensional representation of each node based on the adjacency matrix E and the attribute matrix Z, and the learned embedding can preserve the proximity existing from the topological structure and node attributes. DANE incudes two branches which are composed of a multi-layer nonlinear function. The first branch captures the highly non-linear network structure of PPI networks and maps the input M to a low-dimensional embedding representation, and the second branch maps the input Z to a low-dimensional embedding representation. The architecture of DANE is shown in Fig.2 [33]. To extract nonlinear features of the PPI network, DANE [33] uses autoencoders to learn network information of PPI and properties of protein sequences respectively. Each autoencoder is composed of an input layer, a hidden layer, and an output layer. Taking a protein sequence attribute as the input, the feature learning process of the autoencoder is formulated as the formulas:

$$\left\{\begin{array}{l} h_i=\sigma\left(W^1x_i+b^1\right) \\ \hat{x_i}=\sigma\left(W^2h_i+b^2\right) \end{array}\right. \tag{2}$$

where $x_i\in\mathcal{R}^d$ represents the i-th input protein data. $h_i\in\mathcal{R}^{d'}$ represents the hidden representation encoded by the encoder. $\hat{x_i}$ is the reconstructed data point from the decoder. $\sigma(\cdot)$ represents a nonlinear function, such as sigmoid, ReLu, and Tanh. All learnable parameters of DANE are formulated as $\Theta=\left\{W^1,W^2,W^K,\ldots,b^1,b^2,\ldots,b^K\right\}$, and the parameters are updated by minimizing the reconstruction error as $\min_\Theta\sum_{i=1}^n\|\hat{x_i}-x_i\|_2^2$.

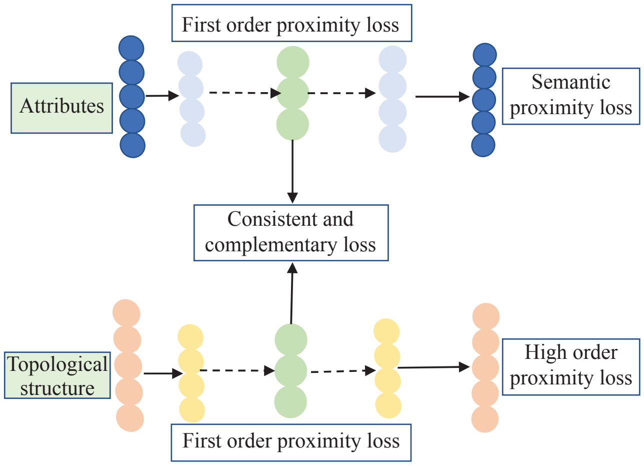

First order proximity loss
Attributes
Semantic proximity loss
Consistent and complementary loss
Topological structure
High order proximity loss
First order proximity loss

Fig. 2. The architecture of DANE. The input is the topological structure and the node attribute.

For DANE, the input of the first branch is the high-order proximity matrix to capture the non-linearity in the topological information of PPI networks. The input of the second branch is the attribute matrix Z to capture the non-linearity in the attribute. The two branches of DANE use N layers in the encoder and the computation process of hidden representation is formulated as

$$\begin{cases} h_i^1 &= \sigma\left(W^1 x_i + b^1\right) \\ h_i^N &= \sigma\left(W^N h_i^{N-1} + b^N\right) \\ \hat{x}_i &= \sigma\left(W^{N+1} h_i^N + b^{N+1}\right) \end{cases} \tag{3}$$

Moreover, there exist N layers in the decoder, and h_i^N represents the desired low-dimensional feature embedding of the i node. Consequently, we denoted the representation from the topological structure as H^M and denoted the representation from attributes as H^Z. Finally, the topological information features H_{TF} of PPI networks and amino acid information are generated by fusing topological structure H^M and attributes H^Z for each protein, $H_{TF} = f_{DANE}(Z) = \left[H^M; H^Z\right]$. To preserve the first-order proximity in the topological structure and attributes simultaneously, we minimized the negative log likelihood as $\zeta_f = -\sum_{E_{ij}} \log P_{ij}^M - \sum_{E_{ij}} \log P_{ij}^Z$. To preserve the high-order proximity, we minimized the reconstruction loss as $\zeta_h = \sum_{i=1}^{n} \left\| \hat{M}_i - M_i \right\|_2^2$. To preserve the semantic proximity, we minimized the reconstruction loss between the input Z and the output $\hat{Z}$ of the decoder as $\zeta_s = \sum_{i=1}^{n} \left\| \hat{Z}_i - Z_i \right\|_2^2$. To combine the topological structure and attributes, we maximized the following likelihood estimation $\zeta_c = \sum_i (\log P_{ij} - \sum_{E_{ij}} \log(1 - P_{ij}))$. Consequently, to preserve the proximity and extract the consistent and complementary representation information for the PPI networks, the objective function jointly ζ_{loss} optimized as follows:

$$\zeta_{loss} = \zeta_s + \zeta_h + \zeta_f + \zeta_c, \tag{4}$$

where P_{ij} is the joint distribution between node i and node j and it is formulated as $\mathrm{P}_{ij} = \frac{1}{1+\mathrm{e}^{\left(-H_{i\cdot}^{M}\left(\left(H_{j\cdot}^{Z}\right)^{T}\right)\right)}}$.

3. Convolutional gated recurrent units of gene expression profile

Protein is the product of gene expression, so gene expression profiles can provide inevitable help for the identification of essential proteins. To extract spatial-temporal interaction features from gene expression profiles, we developed the CGRU component by multi-scale CNN and bidirectional GRU. Gene expression information of a protein has 3 consecutive periods, and each period has 12-time points. It can be regarded as a matrix M$_*$, and each row represents a period of gene expression information. The matrix is formalized as

$$M_* = \begin{bmatrix} \alpha_1 & \alpha_2 & \cdots & \alpha_{12} \\ \beta_1 & \beta_2 & \cdots & \beta_{12} \\ \gamma_1 & \gamma_2 & \cdots & \gamma_{12} \end{bmatrix} \tag{5}$$

where α_i is the data of gene expression in the first consecutive period and i time point. β_i is the data of gene expression in the second consecutive period and i time points. γ_i is the data of gene expression in the third consecutive period and i time points.

In the CGRU component, the multi-scale convolution operation includes three types of convolutional layers with different kernel sizes (Conv1, Conv2, and Conv3) and a pooling layer (Pooling). The gene expression information matrix M_* is fed into the first layer of convolution, which is formulated as f_{Conv_I} (I=1, 2, 3). Each row of the matrix M_{*i} represents the vector corresponding to the i-th period in the input data. Convolutional operations are used to extract different scale features of a gene expression profile.

The computation process of convolution operations is defined as

$$f_{Conv_I}\left(M_{*i:i+j-1}\right) = \delta\left(W_I \cdot M_{*i:i+j-1} + b_I\right) \tag{6}$$

$$R_{conv} = f_{conv_3}\left(f_{conv_2}\left(f_{conv_1}\right)\right) \tag{7}$$

where $M_{*i}, M_{*i+1}, \ldots, M_{*i:i+j-1}$ represents the period $M_{*i:i+j-1}$ of a gene expression profile, W_I represents the convolutional kernel, b_I is the bias term, $\delta(\cdot)$ is the Relu activation function [45]. After convolutional operations, the feature output of convolutional filters is sent to the pooling layer, and the pooling layer is used to extract the most important features of the matrix without changing the shape, and this obtains the multi-scale complex hidden interaction R^s and the process is formulated as the formula:

$$R^s = f_{pool}\left(R_{conv}\right) \tag{8}$$

where f_{pool} is the maximum pooling operation.

Compared with traditional LSTM [30], bidirectional GRU [29] can not only avoid vanishing gradients and obtain better recognition performance with fewer parameters. Considering the success and computational advantages of GRU neural networks in sequential data, we employ GRU to capture the long-term dependencies in gene expression profiles from two different directions. Bidirectional GRU cells combine two GRU cells, one moves forward from the start of the sequence and another moves forward from the end of the sequence. Thus, the output of each step depends on both the past and the future data in bidirectional GRU, which can extract long contexts in complete sequences. Particularly, we treat the multi-scale spatial features of gene expression profiles as the input of bidirectional GRU, and the spatial features are denoted as $R^s = [x_1, x_2, \dots, x_T], x_i \in \mathcal{R}^k$, and then are passed into bidirectional GRU for extracting long dependency features. In bidirectional GRU, the hidden state h_t at step t is calculated as the formula:

$$h_t = f_{GRU}\left(x_t, \overrightarrow{h_{t-1}}, \overleftarrow{h_{t-1}}\right) \tag{9}$$

where $\overrightarrow{h_{t-1}}$ and $\overleftarrow{h_{t-1}}$ are the front-to-back context and the back-to-forward context at time $t-1$ respectively. Then, h_t is formulated as

$$h_t = \left[\overrightarrow{h_t}, \overleftarrow{h_t}\right] \tag{10}$$

$$\overrightarrow{h_t} = GRU\left(x_i, \overrightarrow{h_{t-1}}\right) \tag{11}$$

$$\overleftarrow{h_t} = GRU\left(x_i, \overleftarrow{h_{t-1}}\right) \tag{12}$$

Finally, DeepIEP obtains a long-term dependency features $H_{ST} = [h_1, h_2, \dots, h_T]$ from the gene expression sequences.

4. Feature fusion

In the feature fusion component, the learnable weights and biases are W_{FI} and b_{FI}. For each protein, we combine the feature H_{TF} obtained through the DANE component with the feature H_{ST} obtained through the CGRU component, and obtained the global dependency features H that can determine essential proteins.

$$H = f_{FI}\left[H_{ST}, H_{TF}\right] + b_{FI} \tag{13}$$

Finally, DeepIEP uses a fully connected layer to infer the category possibility of each protein, and the learnable weight and bias are W_y and b_y. The process is formulated as the formula:

$$y = f_{sigmoid}(H) = \frac{1}{1 + \mathrm{e}^{W_y \cdot H + b_y}} \tag{14}$$

5. Parameter learning

In this work, the essential protein recognition is a binary classification task. The binary cross entropy function is the loss function of DeepIEP since it is a commonly used function for binary classification. Therefore, the loss function can be formulated as $\xi_{loss} = -\sum_{i=1}^{n} y_i' \log(y_i) + (1 - y_i') \log(1 - y_i')$. The training parameters of DeepIEP are defined as θ, and the target of the training process is used to maximize the log-likelihood probability $\log P$ based on all parameters θ of DeepIEP and reduce the loss values between true labels and predicted labels. The log-likelihood probability is formulated as

$$\theta \rightarrow \sum_{\mathrm{N}} \log P\left(y_i' | N_i, \theta\right) \tag{15}$$

where N is the number of the training signal data and y_i' is a correct label of a sequence N_i. Adam [46] is selected as the optimizer to minimize the loss function, which is an algorithm for first-order gradient-based optimization of stochastic objective functions via adaptive estimates of lower-order moments.

Algorithm 1 describes the feature learning process of DeepIEP.

Algorithm 1 The feature learning of DeepIEP

```
Input: The PPI network $I_{PPI}$, protein sequences $Z_i$, and
    gene expression profiles $I_{CE}$.
Output: Discriminative contextual features $H$ for identifying essential proteins.
Begin
1: Calculate spatial-temporal interaction features $H_{ST}$ of
    Gene expression information via multi-scale convolutions
    and bidirectional GRU;
2: Calculate topological information features of PPI networks
    and amino acid information $H_{TF}$ via deep attributed graph
    embedding algorithm;
3: Combine $H_{ST}$ with $H_{TF}$ to obtain discriminative contextual features $H$ via fully connected layers;
End
```

IV. Experiments and Results

This section introduces datasets, parameter settings, evaluation metrics, results and comparison analysis, the ablation study, and discussion respectively.

1. Datasets

To verify the effectiveness of DeepIEP, we conducted extensive experiments and compared them with baseline models in terms of Accuracy, Precision, Recall, F-measure, and AUC. For the essential protein prediction experiments, we adopted PPI networks of the S. cerevisiae dataset [16], essential protein data, and gene expression data. Besides, amino acid sequences of proteins are collected from the NCBI website (*https://www.ncbi.nlm.nih.gov/*).

1) PPI network dataset & essential protein dataset

Two PPI network datasets including BioGRID-PPI and DIP-PPI are downloaded from the BioGRID [47] and DIP [48] datasets, and then the duplicate protein-protein interactions are eliminated from the datasets. The vast majority of data on DIP is from yeast, Helicobacter pylori, and human. The data of DIP is from Saccharomyces cerevisiae, Caenorhabditis elegans, Drosophila melanogaster, and Homo sapiens. Next, the proteins are removed, where their amino acid sequence cannot be found on the NCBI website. Finally, the BioGRID-PPI dataset includes 5506 nodes, 52226 pairs of protein-protein interactions, and 1192 essential proteins. The DIP-PPI dataset includes 4722 nodes, 15093 pairs of protein-protein interactions, and 1125 essential proteins.

2) Gene expression dataset

The gene expression data is extracted from GEO (Gene Expression Omnibus) [49] with accession number GSE3431. Briefly, the dataset contains 6777 proteins and 36-time points totally and has 3 consecutive metabolic cycles and each cycle with 12-time points in a cycle.

3) Sequence dataset

We downloaded the amino acid sequences of each protein from NCBI and removed duplicated proteins in the PPI network dataset [16]. Moreover, proteins are removed when the amino acid sequence is not found in the PPI network, and we eliminated proteins whose sequence length is greater than 3000.

2. Evaluation metrics and model parameters

All the deep models of this work are implemented in Tensorflow, and the 10-fold cross-validation is used to evaluate the performance and the robustness of the models. Our models can be downloaded from the website (*https://github.com/yxinshidai/pro.git*). For the hyper-parameters of models, the settings are taken from [16], [26], [27], [40], and are slightly fine-tuned to select the optimal parameters via the grid searching strategy manually. The main parameters of DeepIEP are shown in Table 2.

Table 2. Parameter settings of DeepIEP

Parameters	Settings
Number of iterations	50
Batch size of datasets for DeepIEP	64
Optimizer for the loss function	Adam
Learning rate	0.001
Feature dimension of DANE and GCRU	64,6
Number of different convolutional filters	8,8,8
Kernel size of different convolutions	1,3,5
Number of hidden units for Bidirectional GRU	6

To compare their performance between DeepIEP and baselines, this work uses accuracy (A_{cc}), precision (P_{re}), recall (R_{ec}), F-measure ($F_{mesasure}$) and area under the receiver operating characteristic curve (AUC) as evaluation metrics. They are formulated as the formulas:

$$\begin{cases} A_{cc} = \dfrac{TP+TN}{TP+TN+FP+FN} \\ P_{re} = \dfrac{TP}{TP+FP} \\ R_{ec} = \dfrac{TP}{TP+FN} \\ F_{measure} = \dfrac{(1+\beta^2)\cdot P_{re}\cdot R_{ec}}{\beta^2\cdot P_{re}+R_{ec}} \end{cases} \tag{16}$$

where TP and TN represent the number of samples of essential and non-essential proteins that are correctly classified, while FN and FP represent the number of samples of essential and non-essential proteins that are incorrectly classified, and β is the adjustment weight between the precision rate and the recall rate. The parameter of β is 1 in the experiment.

3. Experimental results and comparison analysis

This section introduces a comparison of experimental results with centrality-based methods, traditional machine learning-based methods, and deep learning-based methods, respectively.

1) Comparison with centrality-based methods

To evaluate the representation ability of DeepIEP for topological characteristics of PPI networks, we first selected centrality-based methods as baseline methods as Zeng *et al.* [16] did, which include degree centrality (DC) [10], betweenness centrality (BC) [11], closeness centrality (CC) [12], eigenvector centrality (EC) [14], network centrality (NC) [50] and local average connectivity (LAC) [51]. These centrality-based methods largely depend on the reliability of PPI networks. We calculated scores of proteins and ranked them in descending order via the scores, and then we selected the top 1192 of BioGRID-PPI and 1125 proteins of DIP-PPI ranked by these methods as their predicted essential proteins. The rest of the proteins are considered to be non-essential proteins. Finally, in comparison with the true labels of essential proteins and non-essential proteins, a confusion matrix was obtained to calculate metrics. Fig.3 and Fig.4 report experimental results on BioGRID-PPI and DIP-PPI datasets. These experimental results show that our model performed better than all centrality-based methods for identifying essential proteins.

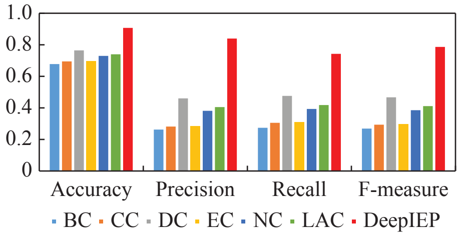


Fig. 3. Performance comparison of DeepIEP and centrality methods on BioGRID-PPI dataset.

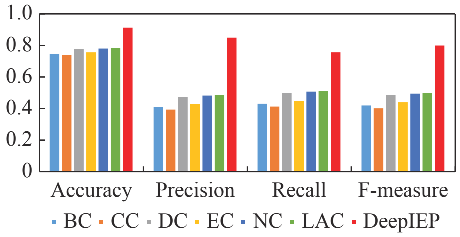


Fig. 4. Performance comparison of DeepIEP and centrality methods on DIP-PPI datasets.

Fig.3 shows that all the evaluation metrics of DeepIEP are higher than baselines including DC, BC, CC, EC, NC, and LAC. The centrality methods used various topological features of PPI networks. Our DeepIEP obtained remarkable performance with 0.901, 0.841, 0.743, and 0.783, respectively on BioGRID-PPI datasets in terms of accuracy, precision, recall, and F-measure. On average, DeepIEP achieved better predictive results with accuracy increasing by 19.05%, precision score increasing by 49.45%, recall score increasing by 38.05%, and F-measure score increasing by 43.37% respectively on BioGRID-PPI than all baseline models. The reason may be that scoring functions rely on a large amount of prior knowledge in centrality-based methods and they fail to consider the intrinsic biological characteristics of proteins. Thus, DeepIEP can automatically capture contextual representation features of proteins from PPI networks and then further conduct accurate identification of essential proteins.

Fig.4 shows that evaluation metrics of DeepIEP are higher than baselines including DC, BC, CC, EC, NC, and LAC. Our DeepIEP obtained remarkable performance with 0.910, 0.847, 0.754, and 0.792, respectively on DIP-PPI datasets in terms of accuracy, precision, recall, and F-measure. On average, DeepIEP achieved better predictive results with accuracy increasing by 14.83%, precision score increasing by 40.40%, recall score increasing by 28.80%, and F-measure score increasing by 34.27% respectively on DIP-PPI than baselines. The results indicate that DeepIEP can capture complex contextual features from PPI networks and improve the performance of essential protein recognition.

2) Comparison with traditional machine learning-based methods

To evaluate the feature learning ability of DeepIEP, we selected four machine learning methods including decision tree (DT), random forest (RF), support vector machine (SVM), and Adaboost as baseline methods. The models merged pattern features learned by the DANE component and CGRU component as the input and identify the essential proteins. Table 3 reports the experimental results of DeepIEP and baselines on BioGRID-PPI and DIP-PPI datasets. Table 3 shows that DeepIEP obtained remarkable performance of essential protein prediction in terms of Accuracy, Precision, Recall, F-measure, and AUC, and the results are superior to machine learning-based methods in all evaluation metrics. On average, DeepIEP achieved better predictive results with accuracy increasing by 16.10%, precision score increasing by 31.85%, recall score increasing by 51.05%, F-measure score increasing by 47.80% and AUC score increasing by 16.70% respectively on BioGRID-PPI than baseline models, and achieved better recognition results with accuracy increasing by 16.25%, precision score increasing by 30.95%, recall score increasing by 51.90%, F-measure score increasing by 48.70% and AUC score increasing by 13.60% respectively on DIP-PPI than baseline models. Experimental results show that our DeepIEP framework can not only automatically extract more discriminative features than machine learning-based methods from protein networks, but also effectively identify essential proteins based on contextual representations.

3) Comparison with deep learning-based methods

We selected four remarkable deep neural networks as baseline models to show the effectiveness and superiority of our DeepIEP framework for identifying essential proteins. The details of the models are as follows: DeepEP: Zeng *et al.* [26] developed a deep neural convolutional framework called DeepEP to combine two biological information based on convolutional neural networks that used the node2vec technique and a sampling technique to identify essential proteins. DeepEP-LSTM: Zeng *et al.* [16] used bidirectional LSTM to design a deep learning framework called DeepEP-LSTM for identifying essential proteins by integrating biological information of PPI networks, gene expression profiles, and subcellular localization information. DeepHE: Zhang *et al.* [27] proposed a deep multilayer perceptron-based model called DeepHE to predict human essential genes by integrating features derived from sequence data and PPI networks. ADRSNet: Altwaijry *et al.* [40] designed an Arabic handwriting recognition system by

Table 3. Performance comparison between DeepIEP and baselines on two PPI datasets

Datasets	Model	Accuracy	Precision	Recall	F-measure	AUC
BioGRID-PPI	DT	0.73	0.33	0.26	0.29	0.60
DIP-PPI	DT	0.72	0.55	0.30	0.34	0.70
BioGRID-PPI	RF	0.78	0.47	0.12	0.26	0.65
DIP-PPI	RF	0.80	0.54	0.18	0.27	0.75
BioGRID-PPI	SVM	0.73	0.80	0.28	0.32	0.72
DIP-PPI	SVM	0.70	0.51	0.25	0.31	0.68
BioGRID-PPI	Adaboost	0.72	0.49	0.27	0.35	0.73
DIP-PPI	Adaboost	0.77	0.55	0.21	0.30	0.73
BioGRID-PPI	DeepIEP	0.901	0.841	0.743	0.783	0.842
DIP-PPI	DeepIEP	0.910	0.847	0.754	0.792	0.851

using a convolutional neural network, which is denoted as ADRSNet, and it was inherently suitable for problems with high dimensionality. Besides, we further designed two variant models of DeepIEP as baseline models: 1) IEPHDL includes bidirectional gated recurrent units and multilayer perceptron which is composed of three fully connected layers. 2) DeepANN is composed of attention networks and multilayer perceptron which include three fully connected layers. Table 4 and Table 5 report the experimental results of DeepIEP and baseline models on BioGRID-PPI and DIP-PPI datasets, respectively.

Table 4. Performance comparison of DeepIEP and baselines on BioGRID-PPI datasets

Datasets	Model	Accuracy	Precision	Recall	F-measure	AUC
BioGRID-PPI	DeepEP	0.826	0.584	0.524	0.552	0.816
BioGRID-PPI	DeepHE	0.828	0.663	0.410	0.507	0.678
BioGRID-PPI	DeepEP-LSTM	0.848	0.721	0.427	0.536	0.831
BioGRID-PPI	ADRSNet	0.719	0.295	0.141	0.165	0.510
BioGRID-PPI	IEPHDL	0.887	0.862	0.683	0.758	0.852
BioGRID-PPI	DeepANN	0.851	0.730	0.549	0.587	0.818
BioGRID-PPI	DeepIEP	0.901	0.841	0.743	0.783	0.842

Table 5. Performance comparison of DeepIEP and deep models on DIP-PPI datasets

Datasets	Model	Accuracy	Precision	Recall	F-measure	AUC
DIP-PPI	DeepEP	0.836	0.641	0.538	0.541	0.762
DIP-PPI	DeepHE	0.814	0.667	0.417	0.511	0.677
DIP-PPI	DeepEP-LSTM	0.852	0.703	0.458	0.574	0.839
DIP-PPI	ADRSNet	0.795	0.526	0.617	0.536	0.734
DIP-PPI	IEPHDL	0.871	0.848	0.625	0.716	0.843
DIP-PPI	DeepANN	0.845	0.712	0.583	0.542	0.800
DIP-PPI	DeepIEP	0.910	0.847	0.754	0.792	0.851

Table 4 indicates that our DeepIEP framework obtained the best performance in terms of accuracy, precision, recall, F-measure, and AUC on BioGRID-PPI datasets. DeepIEP averagely achieved better predictive results with accuracy increasing by 7.45%, precision increasing by 19.85%, recall increasing by 28.73%, F-measure increasing by 26.55%, and AUC increasing by 9.12% respectively. Specifically, DeepIEP achieved the performance improvement with accuracy increasing by 5.30%, precision increasing by 12.0%, recall increasing by 31.60%, F-measure increasing by 24.70%, and AUC increasing by 1.10% than DeepEP-LSTM on BioGRID-PPI datasets. The possible reason is that DeepIEP can automatically capture more contextual and discriminative features than DeepEP-LSTM based on protein network information from the PPI network and the topological structure and attributes of proteins are useful for improving the performance of identifying essential proteins. Besides, DeepIEP achieved the performance improvement with accuracy increasing by 7.50%, precision increasing by 26.10%, recall increasing by 22.30%, F-measure increasing by 23.30%, and AUC increasing by 2.20% than DeepEP on BioGRID-PPI datasets. The above experiments imply that the features of protein sequences can significantly improve the performance of essential protein identification, and DeepIEP could capture more complex and spatial-temporal features.

Besides, to evaluate the generalization ability of DeepIEP, we further conducted the essential protein prediction on DIP-PPI datasets and compared performance with state-of-the-art deep learning predictors. Table 5 reports their experimental results. DeepIEP obtains the best performance than baselines on DIP-PPI datasets. Compared to the deep learning methods, DeepIEP averagely achieved better predictive results with accuracy increasing by 7.45%, precision increasing by 16.42%, recall increasing by 21.43%, F-measure increasing by 22.20% and AUC increasing by 7.52% respectively. Particularly, compared to DeepHE, DeepIEP achieved the results with accuracy increasing by 9.60%, precision increasing by 18.00%, recall increasing by 33.70%, F-measure increasing by 28.10%, and AUC increasing by 17.40% respectively. Compared to ADRS-Net, DeepIEP achieved the results with accuracy increasing by 11.50%, precision increasing by 32.10%, recall increasing by 13.70%, F-measure increasing by 25.60%, and AUC increasing by 11.70% respectively. The above analysis fully indicates that our DeepIEP framework is an effective and efficient deep neural model for identifying essential proteins.

4. Ablation study

This section introduces an ablation study to evaluate the robustness and efficiency of models.

1) Impact of the graph embedding method

To verify the efficiency of DANE in feature extraction, we selected an attribute network representation learning (ANRL) method [52] as a feature representation method for PPI networks. In experiments, we replaced DANE of DeepIEP with ANRL. ANRL also used protein sequence features as the attribute features of the network nodes on the PPI network. The encoding methods of protein sequences include integer [53], *k*-mer, and *k*-mer-bow [30]. Table 6 reports experimental results of DeepIEP with different encoding methods on DIP-PPI datasets.

From the results of Table 6, our DeepIEP framework obtained remarkable results based on *k*-mer embedding information for identifying essential proteins. When ANRL is used to capture the topological information of *k*-mer in the PPI network, the performance of DeepIEP decreased by 9.0%, 18.5%, 37.0%, 47.5%, and 23.2% respectively from Accuracy, Precision, Recall, F-measure and AUC. Obviously, DANE can capture more discriminative information than ANRL for topological structure and node attributes from the PPI networks.

Table 6. Experimental results of DeepIEP with different protein sequence representation

Feature-model	Sequence-encoding	Accuracy	Precision	Recall	F-measure	AUC
ANRL	Integer	0.798	0.617	0.182	0.281	0.576
ANRL	*k*-mer	0.801	0.624	0.320	0.203	0.575
ANRL	*k*-mer-bow	0.803	0.682	0.371	0.274	0.576
DANE	Integer	0.884	0.795	0.622	0.697	0.789
DANE	*k*-mer	0.901	0.841	0.743	0.783	0.842
DANE	*k*-mer-bow	0.887	0.845	0.617	0.702	0.791

2) Impact of different feature input

This part analyzes the impact of different input features on DeepIEP and confirms whether it is necessary to use protein sequence features and gene expression data as node's attribute feature at the same time. We conducted three experiments to show the necessity of protein sequence features and gene expression data for essential protein recognition. The parameters of DeepIEP are shown in Table 2 on DIP-PPI datasets. Firstly, only gene expression information is used as node feature input of DeepIEP and the accuracy of DeepIEP is 0.834. Secondly, only protein sequence is used as node feature input, and the accuracy of DeepIEP is 0.890. Finally, using gene expression information and protein sequence as node feature input at the same time, the accuracy of DeepIEP is 0.901. Thus, DeepIEP obtained the best predictive results for identifying essential proteins when using gene expression information and protein sequences as attribute features of nodes, and the results show the importance of protein sequence features for identifying essential proteins.

3) Dimension of graph embedding

In DeepIEP, the DANE component is used to capture high nonlinearity information and preserve various proximities both topological structure and node attributes of the PPI networks. Fig.5 reports the performance of DeepIEP based on feature representations with different dimensions. It can be seen from Fig.5 that the dimension of embedding has a great impact on the recognition performance of DeepIEP for essential proteins. For DeepIEP, when *k*-mer embedding was set to 64 and our model obtained a remarkable performance with Accuracy of 0.901, Precision of 0.841, Recall of 0.743, F-measure of 0.783, and AUC of 0.842 respectively on BioGRID-PPI. When dimensions are greater than 64, the recognition performance of the model gradually decreases. Consequently, the dimension of DANE output is set to 64 for accurately identifying essential proteins.

5. Discussion

The PPI network usually has thousands of vertices and tens of thousands of edges and the output of designed score functions is just a real number such as DC

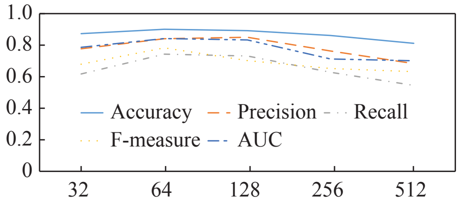


Fig. 5. Performance of DeepIEP based on feature representations of different dimensions.

[10], BC [11], CC [12], EC [14], NC [50] and LAC [51], and it is difficult to represent complete topological features by a real number. For machine learning-based algorithms, their selected biological properties could not represent the complete features of biological information, such as DT, RF, SVM, and Adaboost. Furthermore, there is a lack of efficient algorithms to automatically select contextual features from various biological data. To obtain a discriminative embedding, the network embedding and deep model need to consider: 1) The PPI network is incomplete and inherently noisy via high-throughput technologies, and the topological based methods ignored the intrinsic biological property of essential proteins. 2) The underlying structure of the topological structure and attributes are highly non-linear, and the proximity in an attributed network depends both on the topological structure and the attribute. 3) Consistent and complementary information in the topological structure and attributes provide different views for each protein node in the PPI networks. However, these centrality-based methods largely depend on the reliability of PPI networks, and these methods fail to consider intrinsic biological characteristics of proteins. To address the limitations of traditional methods, deep attributed network embedding (DANE) [33] is effective and efficient for identifying essential proteins in both topological topological structure and node attributes of the PPI network.

Due to the powerful learning capability of deep models, some CNN and LSTM-based methods have been proposed to automatically learn informative feature patterns such as DeepEP [26], DeepEP-LSTM [16], DeepHE [27], and ADRSNet [40], which can avoid the trouble of artificially designing various special features for biological signal processing and analysis. Although the computational methods have presented promising results of essential protein recognition, and however the methods rely heavily on convolutional operations to capture local complex discriminative features of the raw protein data. The deep models do not fully extract complex hierarchical dependency features and high proximities of protein sequences, and seldom utilize multiscale convolutional operations to generate the contextual feature representation for the prediction task. Thus, we design the efficient deep framework to automatically extract spatial-temporal context features for accurately identifying essential proteins.

Although DeepIEP shows a great performance for identifying essential proteins, there exist some limitations. For example, the class is imbalance problem, where the number of non-essential proteins usually far exceeds that of essential proteins. The classifiers aim at maximizing the overall classification accuracy and tend to misclassify minority classes as majority classes. Protein properties are diverse and have the characteristics of a spatial hierarchy, such as secondary structures, solvent accessibility, and backbone angles. In the future, we will evaluate the efficacy of DeepIEP in microbe-disease associations [54], and lncRNA-disease associations [55], and further explore more effective algorithms to integrate different biological information of proteins for identifying essential proteins, and evaluate the efficacy of our model in some special species. Besides, we will explore interpretable neural networks to model protein interaction networks based on Bayesian neural networks and design an efficient strategy for reducing the effect of imbalance problems.

V. Conclusions

With the development of high-throughput sequencing techniques, massive protein sequences have been obtained, which makes it possible for us to adopt remarkable intelligent algorithms. In this work, we combine sequence features extracted from protein sequence, gene expression information, and features learned from the PPI network, and then propose an effective deep learning framework, which is called DeepIEP, to capture the high nonlinearity and preserve various proximities of topological structure and node attributes from PPI networks. Moreover, the deep attributed network embedding algorithm is an effective and efficient tool to improve the performance of identifying essential proteins, which includes the extraction of low-dimensional representations by preserving the structure and the attribute information. Experimental results show that fusing biological features can improve the effectiveness of methods for identifying essential proteins, and DeepIEP achieves state-of-the-art performance compared to shallow machine learning methods and existing deep models.

References

[1] W. He, L. Zhang, O. D. Villarreal, *et al.*, “De novo identification of essential protein domains from CRISPR-Cas9 tiling-sgRNA knockout screens,” *Nature Communications*, vol.10, article no.articleno.4541, 2019.

[2] P. Y. Zhang, M. T. Zhang, H. Liu, *et al.*, “Prediction of

protein subcellular localization based on microscopic images via multi-task multi-instance learning," *Chinese Journal of Electronics*, vol.31, no.5, pp.888–896, 2022.

[3] X. Q. Yang, X. J. Lei, and J. Zhao, "Essential protein prediction based on shuffled frog-leaping algorithm," *Chinese Journal of Electronics*, vol.30, no.4, pp.704–711, 2021.

[4] M. R. Fan, M. Li, Z. F. Liu, *et al.*, "Crystal structures of the PsbS protein essential for photoprotection in plants," *Nature Structural & Molecular Biology*, vol.22, no.9, pp.729–735, 2015.

[5] M. Li, R. Q. Zheng, H. H. Zhang, *et al.*, "Effective identification of essential proteins based on priori knowledge, network topology and gene expressions," *Methods*, vol.67, no.3, pp.325–333, 2014.

[6] X. Y. Li, W. K. Li, M. Zeng, *et al.*, "Network-based methods for predicting essential genes or proteins: A survey," *Briefings in Bioinformatics*, vol.21, no.2, pp.566–583, 2020.

[7] L. M. Cullen and G. M. Arndt, "Genome-wide screening for gene function using RNAi in mammalian cells," *Immunology & Cell Biology*, vol.83, no.3, pp.217–223, 2005.

[8] T. Roemer, B. Jiang, J. Davison, *et al.*, "Large-scale essential gene identification in *Candida albicans* and applications to antifungal drug discovery," *Molecular Microbiology*, vol.50, no.1, pp.167–181, 2003.

[9] H. Jeong, S. P. Mason, A. L. Barabási, *et al.*, "Lethality and centrality in protein networks," *Nature*, vol.411, no.6833, pp.41–42, 2001.

[10] M. W. Hahn and A. D. Kern, "Comparative genomics of centrality and essentiality in three eukaryotic protein-interaction networks," *Molecular Biology and Evolution*, vol.22, no.4, pp.803–806, 2005.

[11] M. P. Joy, A. Brock, D. E. Ingber, *et al.*, "High-betweenness proteins in the yeast protein interaction network," *Journal of Biomedicine and Biotechnology*, vol.2005, no.2, article no.594674, 2005.

[12] S. Wuchty and P. F. Stadler, "Centers of complex networks," *Journal of Theoretical Biology*, vol.223, no.1, pp.45–53, 2003.

[13] E. Estrada and J. A. Rodriguez-Velazquez, "Subgraph centrality in complex networks," *Physical Review E*, vol.71, no.5, article no.056103, 2005.

[14] P. Bonacich, "Power and centrality: A family of measures," *American Journal of Sociology*, vol.92, no.5, pp.1170–1182, 1987.

[15] K. Stephenson and M. Zelen, "Rethinking centrality: Methods and examples," *Social Networks*, vol.11, no.1, pp.1–37, 1989.

[16] M. Zeng, M. Li, Z. H. Fei, *et al.*, "A deep learning framework for identifying essential proteins by integrating multiple types of biological information," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol.18, no.1, pp.296–305, 2021.

[17] E. Nasiri, K. Berahmand, M. Rostami, *et al.*, "A novel link prediction algorithm for protein-protein interaction networks by attributed graph embedding," *Computers in Biology and Medicine*, vol.137, article no.104772, 2021.

[18] G. S. Li, M. Li, J. X. Wang, J. *et al.*, "Predicting essential proteins based on subcellular localization, orthology and PPI networks," *BMC Bioinformatics*, vol.17, no.S8, article no.279, 2016.

[19] A. M. Gustafson, E. S. Snitkin, S. C. J. Parker, *et al.*, "Towards the identification of essential genes using targeted genome sequencing and comparative analysis," *BMC Genomics*, vol.7, article no.265, 2006.

[20] J. C. Zhong, J. X. Wang, W. Peng, *et al.*, "Prediction of essential proteins based on gene expression programming," *BMC Genomics*, vol.14, no.S4, article no.S7, 2013.

[21] X. Y. Zhu, Y. C. Zhu, Y. H. Tan, *et al.*, "An iterative method for predicting essential proteins based on multifeature fusion and linear neighborhood similarity," *Frontiers in Aging Neuroscience*, vol.13, article no.799500, 2022.

[22] L. Wang, J. X. Peng, L. N. Kuang, *et al.*, "Identification of essential proteins based on local random walk and adaptive multi-view multi-label learning," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol.19, no.6, pp.3507–3516, 2022.

[23] Y. C. Hwang, C. C. Lin, J. Y. Chang, *et al.*, "Predicting essential genes based on network and sequence analysis," *Molecular BioSystems*, vol.5, no.12, pp.1672–1678, 2009.

[24] J. Y. Deng, L. Deng, S. C. Su, *et al.*, "Investigating the predictability of essential genes across distantly related organisms using an integrative approach," *Nucleic Acids Research*, vol.39, no.3, pp.795–807, 2011.

[25] A. K. Payra and A. Ghosh, "Identifying essential proteins using modified-monkey algorithm (MMA)," *Computational Biology and Chemistry*, vol.88, article no.107324, 2020.

[26] M. Zeng, M. Li, F. X. Wu, *et al.*, "DeepEP: A deep learning framework for identifying essential proteins," *BMC Bioinformatics*, vol.20, no.S16, article no.506, 2019.

[27] X. Zhang, W. X. Xiao, and W. J. Xiao, "DeepHE: Accurately predicting human essential genes based on deep learning," *PLoS Computational Biology*, vol.16, no.9, article no.e1008229, 2020.

[28] M. H. Chen, C. J. T. Ju, G. Y. Zhou, *et al.*, "Multifaceted protein-protein interaction prediction based on Siamese residual RCNN," *Bioinformatics*, vol.35, no.14, pp.i305–i314, 2019.

[29] Y. B. Guo, B. Y. Wang, W. H. Li, *et al.*, "Protein secondary structure prediction improved by recurrent neural networks integrated with two-dimensional convolutional neural networks," *Journal of Bioinformatics and Computational Biology*, vol.16, no.5, article no.1850021, 2018.

[30] Y. B. Guo, D. M. Zhou, R. C. Nie, *et al.*, "DeepANF: A deep attentive neural framework with distributed representation for chromatin accessibility prediction," *Neurocomputing*, vol.379, pp.305–318, 2020.

[31] M. Ghandi, D. Lee, M. Mohammad-Noori, *et al.*, "Enhanced regulatory sequence prediction using gapped *k*-mer features," *PLoS Computational Biology*, vol.10, no.7, article no.e1003711, 2014.

[32] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Francisco, CA, USA, pp.855–864, 2016.

[33] H. C. Gao and H. Huang, "Deep attributed network embedding", in *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, Stockholm, Sweden, pp.3364–3370, 2018.

[34] X. W. Tang, J. X. Wang, J. C. Zhong, *et al.*, "Predicting essential proteins based on weighted degree centrality," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol.11, no.2, pp.407–418, 2014.

[35] M. Li, H. H. Zhang, J. X. Wang, *et al.*, "A new essential protein discovery method based on the integration of protein-protein interaction and gene expression data," *BMC Systems Biology*, vol.6, article no.15, 2012.

[36] K. Plaimas, R. Eils, and R. König, "Identifying essential genes in bacterial metabolic networks with machine learning methods," *BMC Systems Biology*, vol.4, article no.56, 2010.

[37] D. S. Huang and C. H. Zheng, "Independent component

analysis-based penalized discriminant method for tumor classification using gene expression data," *Bioinformatics*, vol.22, no.15, pp.1855–1862, 2006.

[38] R. Fakoor, F. Ladhak, A. Nazi, *et al.*, "Using deep learning to enhance cancer diagnosis and classification," in *Proceedings of the 30th International Conference on Machine Learning*, Atlanta, GA, USA, pp.3937–3949, 2013.

[39] Y. B. Guo, W. H. Li, B. Y. Wang, *et al.*, "DeepACLSTM: Deep asymmetric convolutional long short-term memory neural models for protein secondary structure prediction," *BMC Bioinformatics*, vol.20, article no.341, 2009.

[40] N. Altwaijry and I. Al-Turaiki, "Arabic handwriting recognition system using convolutional neural network," *Neural Computing and Applications*, vol.33, no.7, pp.2249–2261, 2021.

[41] Z. Cheng, L. Liu, G. L. Lin, *et al.*, "ReHiC: Enhancing Hi-C data resolution via residual convolutional network," *Journal of Bioinformatics and Computational Biology*, vol.19, no.2, article no.2150001, 2021.

[42] M. Alkhodari and L. Fraiwan, "Convolutional and recurrent neural networks for the detection of valvular heart diseases in phonocardiogram recordings," *Computer Methods and Programs in Biomedicine*, vol.200, article no.105940, 2021.

[43] J. B. Wang, "Automated detection of premature ventricular contraction based on the improved gated recurrent unit network," *Computer Methods and Programs in Biomedicine*, vol.208, article no.106284, 2021.

[44] The PLOS Computational Biology Staff, "Correction: Enhanced regulatory sequence prediction using gapped k-mer features," *PLoS Computational Biology*, vol.10, no.7, article no.e1004035, 2014.

[45] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th International Conference on Machine Learning*, Haifa, Israel, pp.807–814, 2010.

[46] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proceedings of the 3rd International Conference on Learning Representations*, San Diego, CA, USA, 2015.

[47] C. Stark, B. J. Breitkreutz, T. Reguly, *et al.*, "BioGRID: A general repository for interaction datasets," *Nucleic Acids Research*, vol.34, no.S1, pp.D535–D539, 2006.

[48] I. Xenarios, L. Salwinski, X. J. Duan, *et al.*, "DIP, the Database of Interacting Proteins: A research tool for studying cellular networks of protein interactions," *Nucleic Acids Research*, vol.30, no.1, pp.303–305, 2002.

[49] B. P. Tu, A. Kudlicki, M. Rowicka, *et al.*, "Logic of the yeast metabolic cycle: Temporal compartmentalization of cellular processes," *Science*, vol.310, no.5751, pp.1152–1158, 2005.

[50] J. X. Wang, M. Li, H. Wang, *et al.*, "Identification of essential proteins based on edge clustering coefficient," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol.9, no.4, pp.1070–1080, 2012.

[51] M. Li, J. X. Wang, X. Chen, *et al.*, "A local average connectivity-based method for identifying essential proteins from the network level," *Computational Biology and Chemistry*, vol.35, no.3, pp.143–150, 2011.

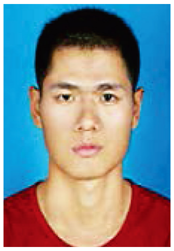
[52] Z. Zhang, H. X. Yang, J. J. Bu, *et al.*, "ANRL: attributed network representation learning via deep neural networks", in *Proceedings of the Twenty Seventh International Joint Conference on Artificial Intelligence*, Stockholm, Sweden, pp.3155–3161, 2018.

[53] H. Öztürk, A. Özgür, and E. Ozkirimli, "DeepDTA: Deep drug-target binding affinity prediction," *Bioinformatics*, vol.34, no.17, pp.i821–i829, 2018.

[54] L. Wang, H. Li, Y. Q. Wang, *et al.*, "MDADP: A webserver integrating database and prediction tools for microbe-disease associations," *IEEE Journal of Biomedical and Health Informatics*, vol.26, no.7, pp.3427–3434, 2022.

[55] P. Y. Ping, L. Wang, L. N. Kuang, *et al.*, "A novel method for LncRNA-Disease association prediction based on an lncRNA-Disease association network," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol.16, no.2, pp.688–693, 2019.

LI Weihua received the Ph.D. degree from Yunnan University, Kunming, China. She is currently an Associate Professor in the School of Information Science and Engineering at Yunnan University, Kunming, China. Her research interests include bioinformatics, data mining and knowledge engineering. (Email: liweihua@ynu.edu.cn)



LIU Wenyang received the B.S. degree from Henan Polytechnic University, Jiaozuo, China. He received the M.S. degree in the School of Information Science and Engineering at Yunnan University, Kunming, China. His research interests include neural networks, intelligent computing, and bioinformatics. (Email: wyl20180901@163.com)

GUO Yanbu (corresponding author) received the Ph.D. degree from Yunnan University, Kunming, China. He is currently a Lecturer in the College of Software Engineering at Zhengzhou University of Light Industry, Zhengzhou, China. His current interests include neural networks, biomedical, and health informatics. (Email: guoyanbu@zzuli.edu.cn)

WANG Bingyi received the Ph.D. from Chinese Academy of Forestry, Kunming, China. He is currently an Associate Research Fellow with the Institute of Highland Forest Science, Chinese Academy of Forestry, Kunming, China. His research interests include bioinformatics and molecular regulation. (Email: wangbykm@163.com)

QING Hua received the Ph.D. degree from South China University of Technology, Guangzhou, China. She is currently a Lecturer in the College of Software Engineering at Zhengzhou University of Light Industry, Zhengzhou, China. Her research interests include machine learning and signal processing. (Email: huaqing@zzuli.edu.cn)