AG-GATCN: A novel method for predicting essential proteins

Peishi Yang(杨培实), Pengli Lu(卢鹏丽)†, and Teng Zhang(张腾)

School of Computer and Communication, Lanzhou University of Technology, Lanzhou 730050, *China*

(Received 18 October 2022; revised manuscript received 8 December 2022; accepted manuscript online 8 February 2023)

Essential proteins play an important role in disease diagnosis and drug development. Many methods have been devoted to the essential protein prediction by using some kinds of biological information. However, they either ignore the noise presented in the biological information itself or the noise generated during feature extraction. To overcome these problems, in this paper, we propose a novel method for predicting essential proteins called attention gate-graph attention network and temporal convolutional network (AG-GATCN). In AG-GATCN method, we use improved temporal convolutional network (TCN) to extract features from gene expression sequence. To address the noise in the gene expression sequence itself and the noise generated after the dilated causal convolution, we introduce attention mechanism and gating mechanism in TCN. In addition, we use graph attention network (GAT) to extract protein–protein interaction (PPI) network features, in which we construct the feature matrix by introducing node2vec technique and 7 centrality metrics, and to solve the GAT oversmoothing problem, we introduce gated tanh unit (GTU) in GAT. Finally, two types of features are integrated by us to predict essential proteins. Compared with the existing methods for predicting essential proteins, the experimental results show that AG-GATCN achieves better performance.

Keywords: complex networks, essential proteins, temporal convolutional network, graph attention network, gene expression

PACS: 89.75.–k **DOI:** 10.1088/1674-1056/acb9f9

1. Introduction

Protein is an essential and vital component of all cells and tissues and is involved in all activities of the organism. The absence of some proteins will lead to the failure of the organism to function properly or even death.[1,2] These proteins are called essential proteins. Predicting essential proteins and analyzing the principles of their interactions will not only help us understanding the minimum needs of organisms for survival, but also help in the diagnosis of diseases and the development of drugs, which will further contribute to the development of the biomedical field. Therefore, the prediction of essential proteins has become one of the research hotspots in bioinformatics. Biological experiments based methods such as single gene knockout,[3] condition gene knockout,[4] and RNA interference[5] have been proposed for essential protein prediction. Although they can accurately predict essential proteins, they are time-consuming, resources-using and less efficient. Therefore, it has become urgent to efficiently predict essential proteins through computable approaches.

With the rapid progress of high-throughput technologies such as yeast two-hybrid screening[6] and protein chip technology,[7] many protein–protein interaction (PPI) networks have been discovered, which build a data base for applying computable methods to predict essential proteins. Subsequently, many centrality approaches based on PPI networks were successively proposed to predict essential proteins, such as degree centrality (DC),[8] betweenness centrality (BC),[9] closeness centrality (CC),[10] subgraph centrality (SC),[11] network centrality (NC)[12] and the local average connectivity-based method (LAC).[13] These methods are similar in nature in that they only take into account the topological characteristics of the PPI network. Although these proposed centrality approaches have illustrated the relevance of PPI network topology properties to the importance of proteins, each centrality method usually captures only one or a few topologies associated with protein importance[14] and does not comprehensively taking account of the full topological information of the proteins in the PPI network, in addition to the existence of false positives and false negatives in the PPI network itself. These aforementioned factors impact the overall performance of centrality methods. With the aim of solving the problem of topological singularity of centrality methods and further improving the accuracy of predicting essential proteins, researchers have started to integrate different factors affecting protein importance and combine them to design new algorithms, which has led to many biological information including gene expression sequence,[15,16] subcellular localization[17] and protein complexes[18] to be combined with PPI networks for predicting essential proteins. During this period, many approaches based on network topology and biological information were proposed to predict essential proteins. Among them, LIDC[19] and united complex centrality (UC)[20] predict essential proteins by binding protein complexes, PeC[21] and JDC[22] predict essential proteins by bind-

†Corresponding author. E-mail: lupengli88@163.com

© 2023 Chinese Physical Society and IOP Publishing Ltd

ing gene expression sequence, SPP[23] is an essential protein prediction approach based on the subnetwork delineation and prioritization of subcellular localization information.

Many machine learning algorithms such as support vector machines (SVM), extreme gradient boosting (XGBoost) and random forest (RF) have been used to predict essential proteins. Among these algorithms, PEPRF uses RF to predict essential proteins by integrating sequence features and topological features,[24] ESS-NEXG uses XGBoost to predict essential proteins by combining multiple biological information to construct weighted networks.[25] Intelligent optimization algorithms have also been used to predict essential proteins by researchers. For instance, EPD-RW combines network topology and multiple biological information to predict essential proteins using a random walk algorithm,[26] and in addition, AFSO-EP is an essential protein prediction approach based on the artificial fish population optimization.[27] Existing methods for predicting essential proteins have yielded promising results, which have had a positive impact on the development of the field of essential protein prediction. However, they are still somewhat limited in some ways and there are still a lot of scopes for accurately predicting essential proteins. Incomplete PPI networks and biological information lead to noise between the data, making existing methods unable to accurately predict the importance of proteins. First, for centrality methods, they more or less ignore part of the information of nodes in the network. Second, for machine learning algorithms, the selection of features relies heavily on the researcher's prior knowledge. Finally, for intelligent optimization algorithms, it is easily trapped in local optimality and leads to low accuracy. In response to these problems, deep learning methods are beginning to be used to predict essential proteins. Zeng *et al.* extracted features from gene expression sequences by treating them as image and inputting them to convolutional neural networks to predict essential proteins.[28] EP-EDL applies multi-scale textual convolutional neural networks (CNN) in the base classifier to extract features from protein sequences and finally integrates multiple base classifiers to predict essential proteins.[29]

In the face of irregular non-Euclidean data such as social networks and biological networks, traditional classical models such as CNNs have low effectiveness. Meanwhile, considering all the above mentioned methods of fusing biological information ignore the noise presented in the gene expression sequence itself and generated during feature extraction. Fortunately, graph neural networks have started to be applied to various graph-structured data and achieved promising results.[30,31] In this paper, we put forward a novel method attention gate-graph attention network and temporal convolutional network (AG-GATCN) to predict essential proteins. To eliminate the effect of noise on gene expression sequences, we extract the features of gene expression sequences using temporal convolutional network (TCN),[32] which based on the attention mechanism and gating mechanism. Among them, we introduce soft attention[33] behind the input layer. This approach makes TCN to pay more attention to the information in the gene expression sequence that plays a role in the importance of the protein and reduces the influence of noise. In addition, we introduce a gating mechanism behind the dilated causal convolution layer to reduce the effect of the noise generated after convolution. For the purpose of extracting richer PPI network features, we first introduce a network embedding method, named node2vec,[34] which is an unsupervised learning algorithm that automatically learns the features of nodes in the network and thus avoids manual feature selection. Then the feature matrix is constructed by combining the seven centrality metrics (including DC, BC, CC, SC, LAC, NC, and UC) and the features learn by node2vec, and finally extract the features of the PPI network using the graph attention network (GAT). It is worth stating that in order to overcome the severe over-smoothing problem of GAT, we also introduce GTU[35] in GAT. For the purpose of verifying the performance of AG-GATCN, we compare it with seven centrality methods, six machine learning algorithms and two state-of-the-art methods using gene expression sequences and PPI networks, and the experimental results demonstrate that AG-GATCN outperforms these methods in all metrics.

2. Methods

In this section, we will introduce AG-GATCN method in detail. AG-GATCN is divided into three main parts, and its workflow is illustrated in Fig. 1. First, we extract the characteristics of gene expression data using improved TCN. Then, the node2vec technology is introduced to learn the features of PPI network and combines 7 centrality indicators as feature matrix, and further extracts richer PPI network features using GAT. Finally, we perform the classification task using DNN by integrating the two types of features.

2.1. Temporal convolutional networks based on the attention mechanism and gating mechanism

Gene expression data are sequence data, where each gene (protein) has 3 sequences, each with a length of 12. TCN is a network structure capable of handling time series data and has been shown to outperform models such as RNN in many studies. Therefore, we use TCN to process the gene expression sequence. However, most of the existing algorithms for predicting essential proteins based on fused gene expression sequence ignore the confounding and invalid messages

such as noise presented in the gene expression sequence itself, which has great negative impact on the model and results in low performance. To overcome and remedy this issue, we introduce the attention mechanism in front of TCN. We make the model pays more attention to features that are more deterministic for essential proteins by using soft attention, thus reducing the attention to interference features. Finally, the data passed through the attention layer is fed into the TCN. In addition, the data information that undergoes dilated causal convolution also generates interference information such as noise and irrelevant information. In an effort to control the messages transfer in the model, we introduce a gating mechanism. The gating mechanism can be used to control how much messages need to be retained and how much need to be discarded after the dilated causal convolution, thus allowing the model to learn dependencies that span a relatively long period of time.

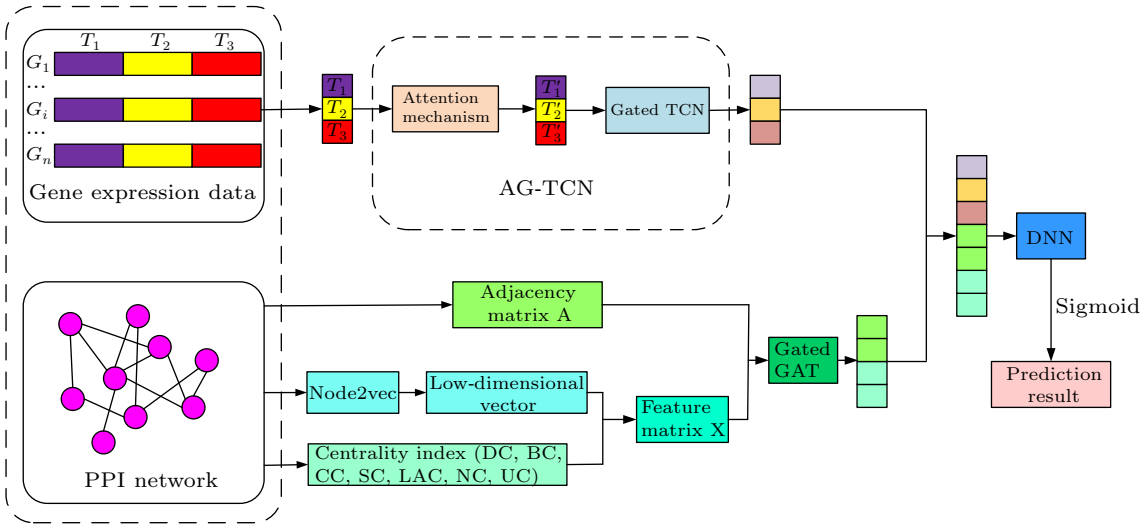


Fig. 1. The workflow of AG-GATCN.

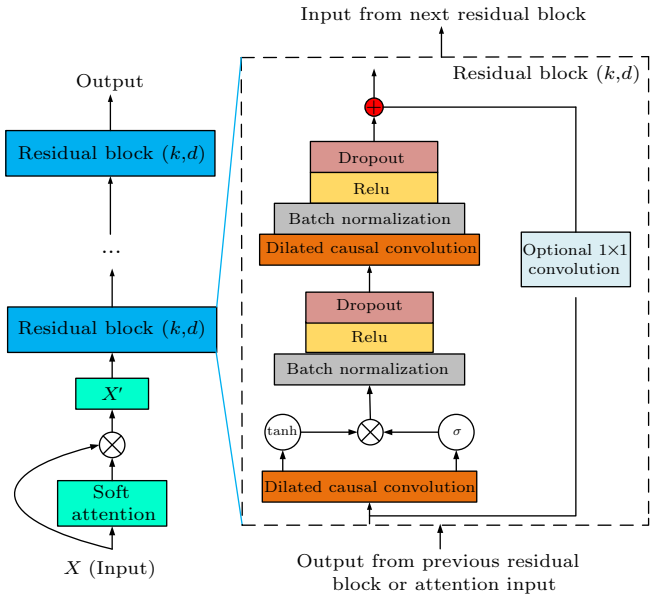


Fig. 2. TCN model based on the attention and gating mechanisms.

Figure 2 shows the TCN model based on the attention and gating mechanisms (AG-TCN). It can be clearly seen that the AG-TCN model introduces attention mechanism at the input side to focus on the valid information, followed by the use of multiple residual blocks. In each residual block, two layers of dilated causal convolution are included, where the first layer of dilated causal convolution introduces a gating mechanism to reduce the interference of irrelevant information. Each layer of the dilated causal convolution is followed by a batch normalization layer to solve the gradient disappearance and explosion problem, a dropout layer to solve the overfitting phenomenon, and ReLU is the activation function.

Supposing filter $f : \{0,\ldots,k-1\} \to \mathbb{R}$ and sequence $X = (x_1, x_2, \ldots, x_T)$, the convolution operation F for the first layer of dilated causal convolution with a dilatation factor of d at x_t is defined as

$$F(t) = \sigma_1\left(\sum_{i=0}^{k-1} f(i)\cdot X_{t-d\cdot i}\right) \otimes \sigma_2\left(\sum_{i=0}^{k-1} f(i)\cdot X_{t-d\cdot i}\right). \quad (1)$$

The convolution operation F for the second layer of dilated causal convolution with a dilatation factor of d at x_t is defined as

$$F(t) = \sum_{i=0}^{k-1} f(i)\cdot X_{t-d\cdot i}, \quad (2)$$

where d is the dilation factor, k is the filter size, σ_1 is the tanh function, σ_2 is the sigmoid function and $\otimes$ denotes matrix multiplication. Through AG-TCN, we finally get 6 features for each protein from the gene expression data.

2.2. Graph attention networks based on the GTU units

The importance of proteins in the PPI network is affected differently by the neighbor information, i.e., different neigh-

bor nodes contribute differently to the importance of the target node. However, many methods such as DC assume the same contribution of each neighbor node when evaluating the protein importance degree, which is not in accordance with the actual conditions. In addition, as non-Euclidean data, PPI network data do not work well in classical models such as CNN. Therefore, in this paper, we process the PPI network through an improved GAT model that enables each node to perform feature aggregation based on the different contributions (weights) of its neighboring nodes to gain a feature representation of each node. Details of the implementation process are as follows.

(i) Constructing the feature matrix. Most of the centrality metrics explain the importance of nodes in the network from different perspectives. However, all of these metrics have their own drawbacks, which result in the difficulty to completely describe the importance of nodes. To construct a richer feature matrix, we introduce a network embedding method node2vec to automatically learn the low-dimensional vector representation of the nodes in the PPI network. Finally, we combine the 128-dimensional vectors obtained by the node2vec technique and 7 centrality metrics to construct the feature matrix of GAT.

(ii) Introduce gating mechanism in GAT. GAT presents a better performance on many graph structure data, yet we get poor results when we charge GAT to process PPI network, even when using only 2 graph attention layers. After several experimental explorations, we find that GAT in PPI network showed severe oversmoothing, i.e., the prediction results of all nodes tend to be pretty close which means that the model can not distinguish very well the importance between different classes of nodes. To solve this problem, we introduce a gating mechanism (GTU units) in the GAT.

In particular, we address the data imbalance by weighting the different categories, where we assign larger weights to the minority samples and smaller weights to the majority samples. Depending on the proportion of different categories in the dataset, we set the weight for nonessential proteins to 1 and for essential proteins to 3.65 in experiments. Finally, with the constructed feature matrix, we extract 64-dimensional features for each protein using the GTU units-based GAT model.

2.3. Deep neural network-based classification model

In the AG-GATCN approach, we use DNN for the final classification task, which mainly consists of an input layer, multiple hidden layers and an output layer. We stitch together the features extracted by AG-TCN and gated GAT as the input to the DNN, and use the sigmoid function for the output layer. To avoid the overfitting phenomenon of the model, we add dropout layers after each hidden layer. In addition, to enhance the generalization ability of the model and speed up the training of the model, we also add BN layers after each hidden layer. Here, the loss function is used as binary cross-entropy, the optimizer is Adam, and our approach to solve the data imbalance problem is similar to the one in GAT.

3. Experiment data and evaluation metrics

In this section, we will present the datasets and evaluation metrics used in the experiments of the AG-GATCN method.

3.1. Experiment data

To assess the overall capability of AG-GATCN, we use the following three main datasets in our experiments, including gene expression dataset, PPI network dataset and essential protein dataset. Among them, the PPI network dataset is downloaded from the BioGRID database (https://thebiogrid.org/).[36] After removing the repeated proteins and interactions, we collect a total of 5616 proteins and 52833 protein–protein interactions in the processed PPI network data. The essential protein dataset is collected by 4 different databases: MIPS,[37] SGD,[38] DEG,[39] and SGDP.[40] The gene expression dataset comes from the gene expression omnibus (GEO) database (https://www.ncbi.nlm.nih.gov/geo/).[41] It contains 9336 proteins and 36 features.

After processing, we finally construct a dataset including 5460 proteins common to both the gene expression dataset and the PPI network dataset for our experiments. The specific statistical information of the dataset is shown in Table 1.

Table 1. Statistical information about the dataset.

Dataset name	No. of proteins	No. of features (known interactions)	No. of essential proteins	Essential protein density (%)
Gene expression dataset	9336	36	1292	13.84
PPI network dataset	5616	52833	1199	21.35
Constructed dataset	5460	6 + 64	1199	21.96

3.2. Evaluation metrics

In order to evaluate the effectiveness of the AG-GATCN method, we use a variety of evaluation metrics. Firstly, we explain true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). TP is the number of candidate essential proteins correctly identified as essential proteins. FN is the number of candidate essential proteins that incorrectly identified as non-essential proteins. FP is the number of candidate non-essential proteins that misidentified as essential proteins. TN denotes the number of candidate non-essential proteins correctly identified as non-essential proteins. Subsequently, we present 6 statistical indicators, which are sensitivity, specificity, positive predictive value, negative predictive

value, F-measure and accuracy, they are defined as

$$\mathrm{SN} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}}, \tag{3}$$

$$\mathrm{SP} = \frac{\mathrm{TN}}{\mathrm{FP}+\mathrm{TN}}, \tag{4}$$

$$\mathrm{PPV} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}}, \tag{5}$$

$$\mathrm{NPV} = \frac{\mathrm{TN}}{\mathrm{TN}+\mathrm{FN}}, \tag{6}$$

$$F = \frac{2\times \mathrm{precision}\times \mathrm{recall}}{\mathrm{precision}+\mathrm{recall}}, \tag{7}$$

$$\mathrm{Acc} = \frac{\mathrm{TP}+\mathrm{TN}}{\mathrm{TP}+\mathrm{TN}+\mathrm{FP}+\mathrm{FN}}, \tag{8}$$

where precision indicates the proportion of positive samples in the predicted positive sample, and recall indicates the proportion of positive samples in the true positive sample that are classified as positive cases. Precision and recall are defined as

$$\mathrm{Precision} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}}, \tag{9}$$

$$\mathrm{Recall} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}}. \tag{10}$$

Moreover, we choose the receiver operating characteristic (ROC) curve as a comprehensive metric to evaluate AG-GATCN. To more intuitively reflect the performance of AG-GATCN, we use the area under the ROC curve (AUC). The AUC visually reflects the classification ability expressed by the ROC curve, and the classification ability of the algorithm is stronger when the AUC value is larger. Because of data imbalance, we also use an AUPR value that is more indicative than AUC, which is the area under the precision recall (PR) curve.

4. Experiment results and discussion

In this section, we introduce the experimental details of the AG-GATCN method, comparison results with three types of methods (centrality methods, machine learning algorithms and deep learning methods), parameter setting experiments, and ablation experiments.

4.1. Experiment details

AG-GATCN is implemented through Keras, which is one of the most classic and widely used open source deep learning libraries.[42] Specifically, GAT is implemented through spektral, an open source Python library for graph deep learning based on the Keras API and TensorFlow 2. In the training process, we set the dropout rate as 0.2 to avoid overfitting, in addition to choosing Adam as the optimizer and setting the learning rate to 0.001. To address the data imbalance, we set the weights of both essential and non-essential proteins according to their proportions in the dataset, and the final weights are 3.65 and 1.0, respectively.

4.2. Comparison with centrality methods

As widely used classical centrality approaches for predicting essential proteins, 7 methods including DC, BC, CC, SC, LAC, NC, and UC are used for comparison with AG-GATCN. The centrality method predicts essential proteins in the following steps.

(i) Calculate the corresponding values for the proteins in the ppi network according to each method.

(ii) Sort the calculated values in descending order.

(iii) The top 1199 (number of essential proteins) proteins in descending order are chosen as candidate essential proteins.

(iv) The candidate essential proteins are compared with the real essential proteins and the corresponding evaluation values are calculated.

Table 2 reflects the performance comparison of the centrality methods and AG-GATCN on six statistical metrics. From Table 2, we can clearly see that AG-GATCN outperforms all the centrality approaches in all metrics. For example, compared to the accuracy of all centrality methods, the accuracy of AG-GATCN is improved by 18.7%, 20.5%, 25.4%, 20.1%, 17.7%, 17.4%, and 19.4%, respectively. In the evaluation metric SN, the value of AG-GATCN is 0.807, while none of the centrality methods exceeds fifty percent, which is more important in the case of unbalanced data categories and fully demonstrates the ability of the AG-GATCN method in predicting essential proteins. In the PPV metric, the value of AG-GATCN is 0.834, which means that the proportion of true essential proteins among our predicted essential proteins is also much higher than in the centrality methods. In other metrics, AG-GATCN is also more effective than the centrality approaches. The experimental results indicate that AG-GATCN has a better performance in predicting essential proteins compared to the centrality methods.

Table 2. Performance comparison between AG-GATCN and other centrality methods.

Method	SN	SP	PPV	NPV	F-measure	Accuracy
DC	0.415	0.848	0.449	0.830	0.432	0.749
BC	0.376	0.836	0.406	0.818	0.390	0.731
CC	0.248	0.811	0.281	0.784	0.263	0.682
SC	0.381	0.840	0.415	0.820	0.400	0.735
LAC	0.436	0.855	0.473	0.836	0.454	0.759
NC	0.445	0.857	0.481	0.838	0.462	0.762
UC	0.386	0.848	0.432	0.823	0.408	0.742
AG-GATCN	0.807	0.936	0.834	0.980	0.866	0.936

4.3. Comparison with machine learning algorithms

To further validate the effectiveness of AG-GATCN, we also compare it with 6 classical machine learning algorithms that are widely used to predict essential proteins, such as SVM, AdaBoost, RF, k-nearest neighbors (KNN), logistic regression (LR), and XGBoost. It is worth mentioning that all machine

learning algorithms are implemented through Python's scikit-learn library.[43] The machine learning algorithm predicts essential proteins with the following steps.

(i) Dataset processing and partitioning. We integrate the values of seven centrality indicators with 36-dimensional features from the original gene expression data to construct $N \times 43$ dimensional feature data as the input data for the machine learning algorithm. Subsequently, the data are normalized by column and randomly disrupted, and finally we divide them into training set and test set, where the ratio of training set and test set to the data set is 0.8 and 0.2, respectively.

(ii) Parameter settings. We use class weights to solve the class imbalance problem in the machine learning algorithm, where the value of essential proteins is assigned 3.65 and the value of non-essential proteins is assigned 1.0 based on the ratio of both in the dataset.

(iii) Train the classifier using the training set.

(iv) Evaluate classifier performance using a test set.

Table 3 shows the results of our experiments on the 6 evaluation metrics of precision, recall, F-measure, accuracy, AUC, and AUPR. As seen in Table 3, AG-GATCN obtains the best performance in all metrics. The accuracy of AG-GATCN is 0.936, while the accuracy of SVM, KNN, RF, AdaBoost, LR and XGBoost is 0.767, 0.783, 0.761, 0.780, 0.738, and 0.728, respectively. In the comprehensive evaluation metric F-measure, the value of AG-GATCN is 0.866, which is 0.458, 0.445, 0.354, 0.372, 0.377, and 0.388 higher than SVM, KNN, RF, AdaBoost, LR, and XGBoost, respectively. The precision obtained by AG-GATCN is 0.834, which is 0.305 higher than the highest precision value of all machine learning algorithms. Similar results are found in the evaluation metric recall. The ROC curve and PR curve of AG-GATCN and all machine learning algorithms are shown in Fig. 3. The AUC and AUPR values obtained by AG-GATCN are 0.979 and 0.933, which are much higher than the values obtained by all machine learning algorithms. Among all machine learning algorithms, AdaBoost obtains the highest AUC and AUPR values of 0.763 and 0.558, respectively. The superiority of AG-GATCN is further demonstrated by comparing it with machine learning algorithms.

Table 3. Performance comparison between AG-GATCN and other machine learning algorithms.

Method	Precision	Recall	F-measure	Accuracy	AUC	AUPR
SVM	0.340	0.512	0.408	0.767	0.747	0.453
KNN	0.332	0.573	0.421	0.783	0.693	0.439
RF	0.529	0.496	0.512	0.761	0.753	0.544
AdaBoost	0.452	0.544	0.494	0.780	0.763	0.558
LR	0.529	0.455	0.489	0.738	0.729	0.448
XGBoost	0.525	0.439	0.478	0.728	0.740	0.524
AG-GATCN	0.834	0.807	0.866	0.936	0.979	0.933

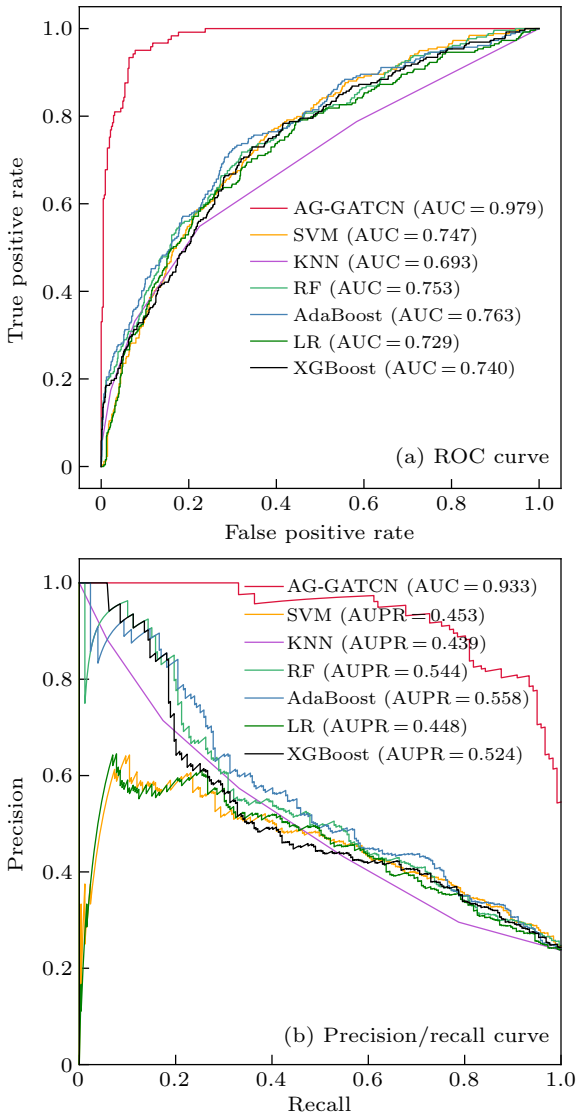


Fig. 3. ROC and PR curves of AG-GATCN and other machine learning methods.

4.4. Comparison with some deep learning methods

In addition to centrality methods and machine learning algorithms, we compare with the current state-of-the-art methods. We search for the latest methods to predict essential proteins using only gene expression data and PPI networks, unfortunately, we only find DeepEP[28] method and the method we call IMBI,[44] it is worth mentioning that subcellular localization data is also used in IMBI, in which we removed the subcellular localization data for our experiments. Table 4 shows the consequences of the comparison between AG-GATCN and these two methods on the evaluation methods of precision, recall, F-measure, accuracy, AUC and AUPR. Figure 4 shows the ROC curves and PR curves for AG-GATCN and these two methods. The comparison outcomes indicate that our method is more effictive than these two models in the 6 indicators of precision, recall, F-measure, accuracy, AUC, and AUPR. This shows that AG-GATCN has better results in predicting essential proteins.

Table 4. Performance comparison between AG-GATCN and other deep learning models.

Method	Precision	Recall	F-measure	Accuracy	AUC	AUPR
DeepEP	0.524	0.610	0.564	0.811	0.811	0.607
IMBI	0.707	0.376	0.491	0.840	0.835	0.616
AG-GATCN	0.834	0.807	0.866	0.936	0.979	0.933

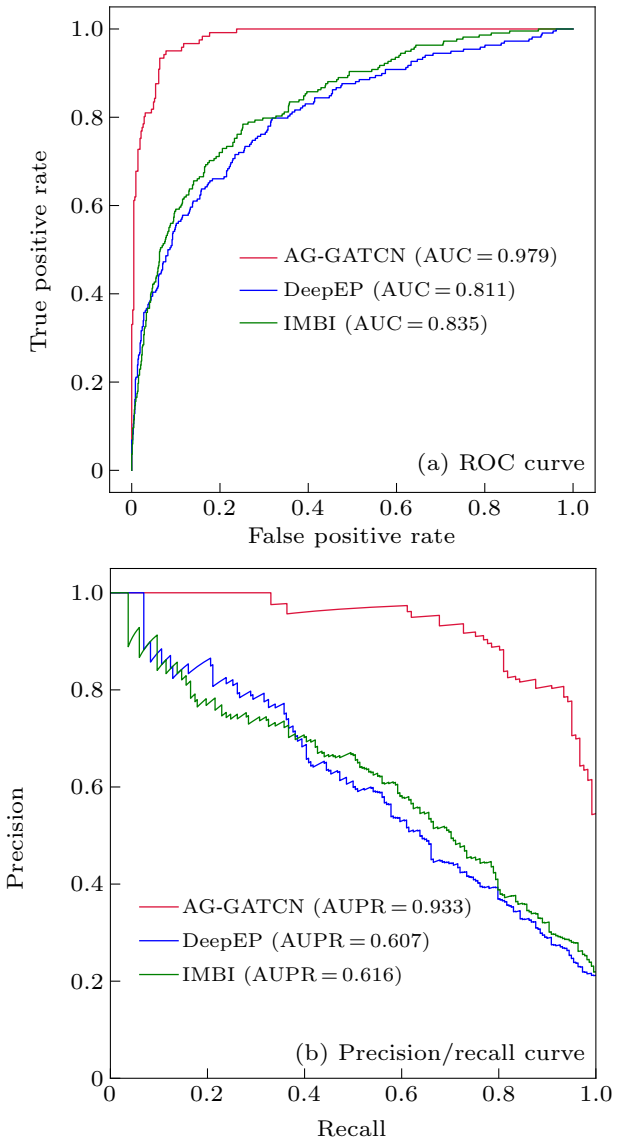


Fig. 4. ROC and PR curves of AG-GATCN and other deep learning methods.

4.5. Parameter setting

Different parameters have different effects on the performance of deep learning models, and here we will focus on the two most important parameters that affect model performance, initial learning rate and batchsize. The learning rate determines the step size of the network weight update, which will directly affect the convergence state of the model. A large initial learning rate will cause a failure of the model to converge and a tendency of the model to oscillate, while a small initial learning rate will cause a slow convergence rate and a tendency of the model to overfit. Batchsize affects the generalization ability of the model. Large batchsize reduces the training time of the model and improves the stability of the model, but too large batchsize will cause a decrease in the generalization ability of the model.

Finding the optimal initial learning rate and batchsize is crucial to the performance of the AG-GATCN method. Therefore, experiments are performed by us using different initial learning rates and batchsizes. Table 5 shows the experimental outcomes, and after the experiments and comprehensive consideration, we finally set the initial learning rate as 0.001 and the batchsize as 32.

Table 5. Performance of AG-GATCN with different initial learning rates and batchsizes.

Initial learning rate	Batchsize	Precision	Recall	F-measure	Accuracy	AUC	AUPR
0.001	16	0.810	0.760	0.784	0.901	0.950	0.853
0.001	32	0.834	0.807	0.866	0.936	0.979	0.933
0.001	64	0.801	0.813	0.855	0.932	0.977	0.931
0.001	128	0.776	0.809	0.841	0.927	0.975	0.926
0.001	256	0.768	0.808	0.837	0.925	0.970	0.892
0.0001	32	0.809	0.730	0.794	0.896	0.970	0.897
0.0005	32	0.817	0.822	0.867	0.934	0.962	0.889
0.005	32	0.769	0.806	0.823	0.927	0.968	0.913
0.01	32	0.726	0.769	0.797	0.907	0.962	0.919

4.6. Ablation study

In the AG-GATCN method, we extract the features of gene expression data by the TCN model based on attention mechanism and gating mechanism, extract the features of PPI network data by the GAT model based on GTU units, and finally work on the classification by integrating the two types of features. To illustrate the necessity of combining the features extracted by both methods, we conduct ablation experiments by removing the features extracted by one of the methods. Table 6 shows the performance of the model using different features. Accuracy is 0.896, 0.827, and 0.936 for using only gene expression features, only PPI network features, and both gene expression features and ppi network features, respectively. The F-measure using both gene expression features and PPI network features is 0.104 and 0.265 higher than that using only gene expression features and PPI network features, respectively. There are similar results in the assessment metrics precision and recall. Figure 5 shows the ROC and PR curves of the model when using different features. It is clear that the AUC and AUPR values of AG-GATCN are higher than both using only gene expression features and using only PPI network features. In summary, AG-GATCN has better performance using both gene expression features and PPI network features than using only gene expression features or using only PPI network features.

Table 6. Ablation study on different models.

Model	Precision	Recall	F-measure	Accuracy	AUC	AUPR
Gene expression features	0.752	0.771	0.762	0.896	0.946	0.846
PPI network features	0.608	0.594	0.601	0.827	0.861	0.603
Gene expression features & PPI network features	0.834	0.807	0.866	0.936	0.979	0.933

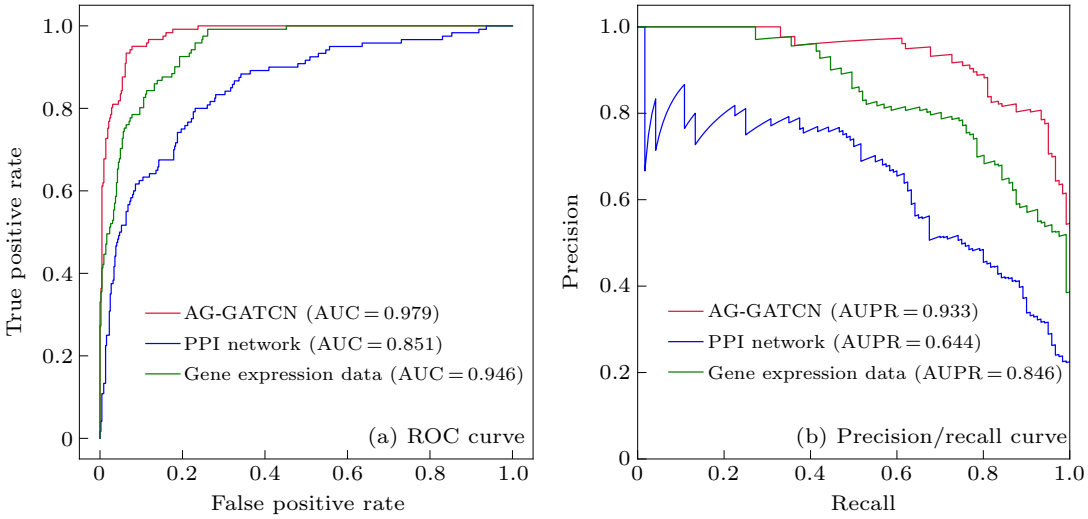


Fig. 5. ROC and PR curves of ablation study on different models.

5. Conclusion

Essential proteins play an extremely significant role in the biomedical field, providing a theoretical basis for disease diagnosis and drug development. Therefore, how to effectively improve the accuracy of predicting essential proteins has become a current research hotspot. In recent years, biological information such as gene expression data has been widely used to predict essential proteins. However, nearly all the existing approaches for predicting essential proteins ignore the noise that exists in the biological information itself. In this paper, we put forward a new method called AG-GATCN to predict essential proteins. The AG-GATCN method consists of the improved TCN model, GTU units-based GAT model and DNN model. In the improved TCN model, we reduce the impact of the noise present in the gene expression sequence itself and the noise generated after convolution on the model performance by introducing attention mechanism and gating mechanism. AG-GATCN uses the GTU units-based GAT model to extract the features of PPI networks. Here we learn the feature representation of proteins in PPI networks by introducing the node2vec technique and construct the feature matrix by combining 7 centrality metrics. In addition, to overcome the issue of over-smoothing of the GAT model, we also introduce the GTU units. Finally, the two types of features are integrated together and fed into the DNN model for the final essential protein prediction. For the purpose of illustrating the effectiveness of AG-GATCN, we compare it with a variety of existing methods for predicting essential proteins, and the experimental results show that AG-GATCN outperformes the existing state-of-the-art methods. It should be added that to illustrate the importance of combining the two types of features, we carry out ablation study by removing one type of feature, and the results demonstrate that combining the two types of features can effectively improve the accuracy of predicting essential proteins.

Acknowledgment

Project supported by the National Natural Science Foundation of China (Grant Nos. 11861045, 11361033, and 62162040).

References

[1] Kamath R S, Fraser A G, Dong Y, *et al.* 2003 *Nature* **421** 231
[2] Clatworthy A E, Pierson E, Hung D T, *et al.* 2007 *Nat. Chem. Biol.* **3** 541
[3] Giaever G, Chu A M, Ni L, *et al.* 2002 *Nature* **418** 387
[4] Cullen L M and Arndt G M 2005 *Immunol.* **83** 217
[5] Roemer T, Jiang B, Davison J, *et al.* 2003 *Mol. Microbiol.* **50** 167
[6] Maple J and Møller S G 2007 *Circadian Rhythms* p. 207
[7] Zhu H and Snyder M 2003 *Curr. Opin. Chem. Biol.* **7** 55
[8] Freeman L C 1978 *Soc. Networks* **1** 215
[9] Joy M P, Brock A, Ingber D E, *et al.* 2005 *Jour. Biom. Biot.* **2005** 96
[10] Wuchty S and Stadler P F 2003 *J. Theor. Biol.* **223** 45
[11] Estrada E and Rodriguez-Velazquez J A 2005 *Phys. Rev. E.* **71** 056103
[12] Wang J X, Li M, Wang H, *et al.* 2011 *IEEE ACM Trans. Comput. Biol. Bioi.* **9** 1070
[13] Li M, Wang J X, Chen X, *et al.* 2011 *Comput. Biol. Chem.* **35** 143
[14] Zhang X, Xiao W X, Acencio M L, *et al.* 2016 *BMC Bioi.* **17** 322
[15] Zhang C L and Zhang S W 2013 *Comput. Biol. Med.* **43** 568

[16] Xiao Q H, Wang J X and Peng X, *et al.* 2015 *BMC Genom.* **16** S1
[17] Tang X W, Wang J X, Zhong J C, *et al.* 2013 *IEEE ACM Trans. Comput. Biol. Bioi.* **11** 407
[18] Yugandhar K and Gromiha M M 2014 *Prot. Stru. Func. Bioi.* **82** 2088
[19] Luo J W and Qi Y 2015 *PloS one* **10** e0131418
[20] Li M, Lu Y, Niu Z B, *et al.* 2015 *IEEE ACM Trans. Comput. Biol. Bioi.* **14** 370
[21] Li M, Zhang H H, Wang J X, *et al.* 2012 *BMC Syst. Biol.* **6** 1
[22] Zhong J C, Tang C, Peng W, *et al.* 2021 *BMC Bioi.* **22** 1
[23] Li M, Li W K, Wu F X, *et al.* 2018 *J. Theor. Biol.* **447** 65
[24] Wu C Y, Lin B T, Shi K, *et al.* 2021 *Curr. Bioi.* **16** 1161
[25] Wang N, Zeng M, Li Y M, *et al.* 2021 *J. Comput. Biol.* **28** 687
[26] Ahmed N M, Chen L, Li B, *et al.* 2021 *Soft Comput.* **25** 8883
[27] Lei X J, Yang X Q, Wu F X, *et al.* 2018 *IEEE ACM Trans. Comput. Biol. Bioi.* **17** 495
[28] Zeng M, Li M, Wu F X, *et al.* 2019 *BMC Bioi.* **20** 506
[29] Li Y M, Zeng M, Wu Y F, *et al.* 2021 *IEEE ACM Trans. Comput. Biol. Bioi.* **19** 3263
[30] Kipf T N and Welling M 2016 arXiv: 1609.02907
[31] Veličković P, Cucurull G, Casanova A, *et al.* 2017 arXiv: 1710.10903
[32] Bai S J, Kolter J Z, Koltun V, *et al.* 2018 arXiv: 1803.01271
[33] Bahdanau D, Cho K, Bengio Y, *et al.* 2014 arXiv: 1409.0473
[34] Grover A and Leskovec J 2016 *Proceedings of the* 22*nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* pp. 855–864
[35] Dauphin Y N, Fan A, Auli M, *et al.* 2017 *International conference on machine learning* pp. 933–941
[36] Chatr-Aryamontri A, Oughtred R, Boucher L, *et al.* 2017 *Nucleic Acids Res.* **45** D369
[37] Mewes H W, Frishman D, Güldener U, *et al.* 2002 *Nucleic Acids Res.* **30** 31
[38] Cherry J M, Adler C, Ball C, *et al.* 1998 *Nucleic Acids Res.* **26** 73
[39] Zhang R and Lin Y 2009 *Nucleic Acids Res.* **37** D455
[40] Winzeler E A, Shoemaker D D, Astromoff A, *et al.* 1999 *Science* **285** 901
[41] Tu B P, Kudlicki A, Rowicka M, *et al.* 2005 *Science* **310** 1152
[42] Zahidi Y, El Younoussi Y, Azroumahli C, *et al.* 2019 2019 5*th International Conference on Optimization and Applications* pp. 1–10
[43] Pedregosa F, Varoquaux G, Gramfort A, *et al.* 2011 *J. Mach. Learn. Res.* **12** 2825
[44] Zeng M, Li M, Fei Z H, *et al.* 2021 *IEEE ACM Trans. Comput. Biol. Bioi.* **18** 296