Computational and Structural Biotechnology Journal 23 (2024) 4315–4323

Contents lists available at ScienceDirect

Computational and Structural Biotechnology Journal

journal homepage: www.elsevier.com/locate/csbj

Method Article

AttentionEP: Predicting essential proteins via fusion of multiscale features by attention mechanisms

Chuanyan Wu [a,1], Bentao Lin [a,1], Jialin Zhang [b], Rui Gao [b,*], Rui Song [b,*], Zhi-Ping Liu [b,*]

[a] School of Intelligent Engineering, Shandong Management University, No.3500 Dingxiang Road, Jinan, Shandong, 250357, China
[b] School of Control Science and Engineering, Shandong University, No.17923 Jingshi Road, Jinan, Shandong, 250061, China

ARTICLE INFO

Keywords:
Essential protein prediction
Feature fusion
Deep learning
Attention mechanism

ABSTRACT

Identifying essential proteins is of utmost importance in the field of biomedical research due to their essential functions in cellular activities and their involvement in mechanisms related to diseases. In this research, a novel approach called AttentionEP for predicting essential proteins (EP) is introduced by attention mechanisms. This method leverages both cross-attention and self-attention frameworks, focusing on enhancing prediction accuracy through the integration of features across diverse scales. Spatial characteristics of proteins are obtained from the protein-protein interaction (PPI) network by employing Graph Convolutional Networks (GCN) and Graph Attention Networks (GAT). Following this, Bidirectional Long Short-Term Memory networks (BiLSTM) are employed to derive temporal features from gene expression datasets. Furthermore, spatial characteristics are derived by integrating data on subcellular localization with the application of Deep Neural Networks (DNN). In order to effectively integrate features across multiple scales, initial steps involve the application of self-attention techniques to derive essential insights from each unique data set. Following this, mechanisms involving self-attention and cross-attention are employed to enhance the interaction between diverse information sources. To identify essential proteins, a classifier based on the ResNet architecture is developed. The findings from the experiments indicate that the method introduced here shows superior performance in identifying essential proteins, recording an Area Under the Curve (AUC) value of 0.9433. This approach shows a considerable advantage over established techniques. The findings of this study provide a significant advancement in the comprehension of critical proteins, revealing promising potential for applications in the development of therapeutics and addressing various diseases.

1. Introduction

Essential proteins are vital for cellular life, as they participate in key biological processes. These proteins are not only integral to maintaining cellular functions, but are also pivotal in the manifestation and progression of diseases [1]. Therefore, accurately identifying essential proteins is a cornerstone in the fields of molecular biology and bioinformatics to aid in drug discovery and provide insights into disease treatment strategies [2,3].

The task of essential protein identification has traditionally been approached through various experimental techniques, which are often labor-intensive, time-consuming, and costly [4,5]. This has led to increased interest in developing more efficient computational methods for predicting essential proteins [6–10]. For the identification of essential proteins, these approaches integrate various biological datasets, including networks of protein interactions, profiling of gene expression, and data regarding subcellular locations [11,12]. However, despite the progress made, many existing computational approaches face challenges in effectively integrating multiscale and multi-domain features, often resulting in suboptimal prediction performance.

To tackle these obstacles, various computational techniques have been developed. Initial investigations largely concentrated on the structural characteristics obtained from PPI networks, employing metrics such as degree, betweenness, and closeness centralities to identify essential proteins. While these strategies yielded important findings, they frequently encountered restrictions stemming from their focus solely on

* Corresponding authors.
E-mail addresses: gaorui@sdu.edu.cn (R. Gao), rsong@sdu.edu.cn (R. Song), zpliu@sdu.edu.cn (Z.-P. Liu).
[1] Chuanyan Wu and Bentao Lin contributed equally.

https://doi.org/10.1016/j.csbj.2024.11.039
Received 25 August 2024; Received in revised form 17 November 2024; Accepted 25 November 2024
Available online 29 November 2024

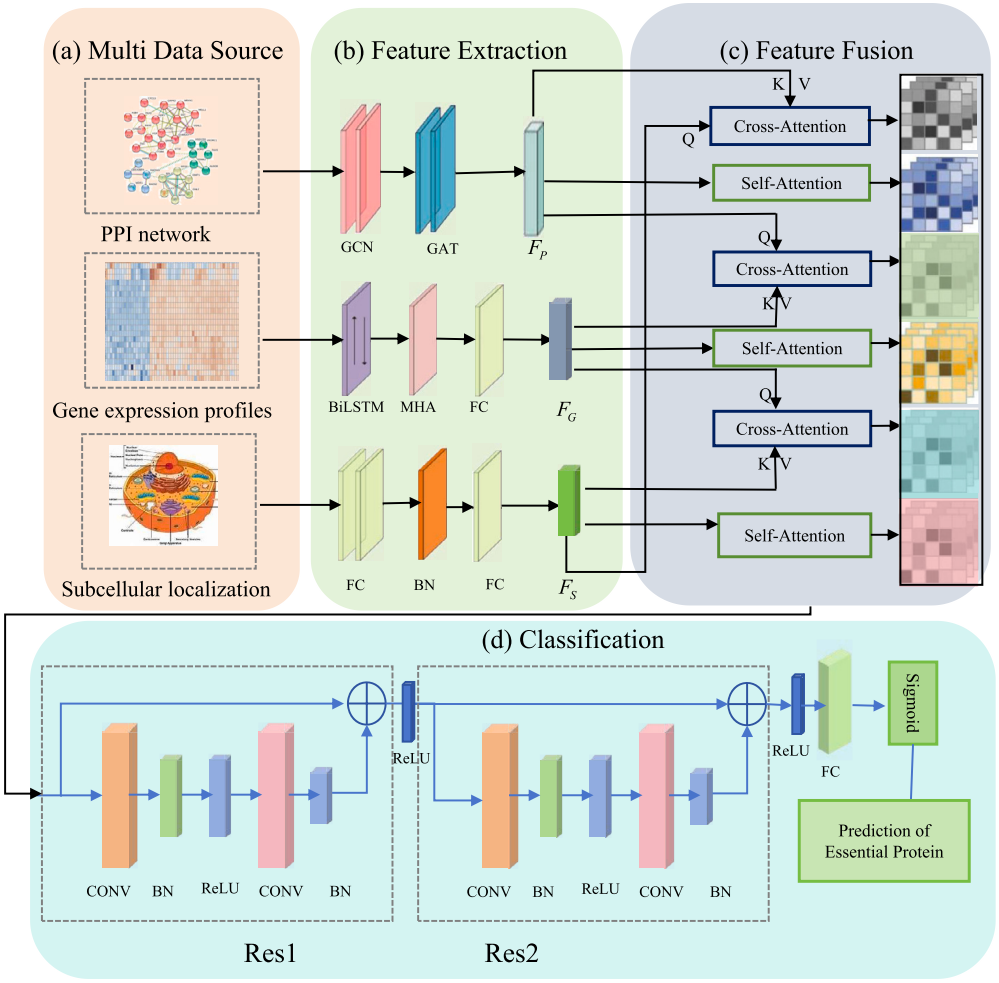

Fig. 1. Flowchart of the Proposed Model AttentionEP.

network structure, neglecting essential biological data found in various other sources and fields [13–17]. Although these methods provided valuable insights, they often suffered from limitations due to their reliance on network topology alone, overlooking critical biological information embedded in other data sources and domains.

To overcome these limitations, recent methods incorporate subcellular localization data and gene expression for better predictions [18]. For instance, Li et al. introduced a method using dynamic thresholding for gene expression binarization and combining centrality and Jaccard index to score interaction network proteins to enhance prediction accuracy [19]. Similarly, Zeng et al. introduced DeepEP by node2vec to automatically capture features of proteins [20].

Recently, deep learning methods have gained prominence in essential protein prediction. These methods excel by modeling complex, nonlinear biological data relationships [21]. For example, A deep learning framework with node2vec and BiLSTM characterized features for essential proteins [22]. Yue et al. used node2vec and depthwise separable convolution to predict essential proteins [23]. Li et al. introduced EP-EDL, which uses protein sequences and multiscale text CNNs for feature extraction to predict human essential proteins [21].

Moreover, attention mechanisms have gained popularity in computational biotechnology for focusing on relevant features in large datasets. Li et al. introduced DeepCellEss based on sequence information using CNN, BiLSTM, and multi-head self-attention for interpretability [24]. Despite significant advancements in computational methods for essential protein prediction, accurately identifying these crucial proteins remains a challenge. This challenge arises from integrating complex, diverse biological data sources, each providing unique insights into protein function and importance. Traditional methods often struggle with this integration, leading to suboptimal prediction performance. In this study, we address these challenges by proposing a novel essential protein prediction model called AttentionEP (Fig. 1). Our model leverages the strengths of both cross-attention and self-attention mechanisms to achieve more precise and reliable identification of essential proteins. AttentionEP integrates multiscale features from PPI networks, gene expression, and subcellular localization using advanced deep learning. We use GCNs and GATs to capture local and global structural information from PPI data. BiLSTMs and multi-head attention (MHA) are employed to extract dynamic temporal features from gene expression data, while DNNs are applied to elucidate spatially related features from subcellular localization data. To address the challenge of multiscale feature integration, our approach uses self-attention to refine multiscale features within each data domain. We then use cross-attention mechanisms to seamlessly integrate these refined features across different domains and scales. This comprehensive and multiscale feature fusion enables our model to capture complex interactions and dependencies among features, resulting in a significant enhancement in prediction performance. Our experimental validation shows that AttentionEP achieves an impressive AUC value of 0.9433. Our results highlight the effectiveness of our model in essential protein prediction, offering potential applications in drug discovery and disease treatment. This research provides new insights that could drive advancements in bioinformatics and molecular biology.

2. Methods

2.1. Protein network structural feature extraction based on GCN and GAT

We utilize a combined GCN and GAT approach to extract structural features from the PPI network. The rationale for this combination lies in the complementary strengths of both methods. GCNs effectively capture local topological information by aggregating features from neighboring nodes. However, GCNs treat all neighboring nodes equally, which may overlook the varying significance of different neighbors. On the other hand, GATs overcome this limitation by assigning varying weights to neighboring nodes based on their importance, enabling detailed feature extraction. Together, GCNs capture local patterns, while GATs enhance this by emphasizing relevant neighbors, improving complex relationship capture in PPI networks.

2.1.1. Component of graph convolution networks (GCN component)

Considering the adjacency matrix A associated with the PPI network along with the characteristics of the proteins with the protein feature matrix X_P, the operation of convolution on graphs at the l-th layer can be articulated as follows.

$$H^{(l+1)} = \sigma\left(\hat{A}H^{(l)}, W^{(l)}\right), \tag{1}$$

where $H^{(l)} \in \mathbb{R}^{N\times F}$ represents the feature matrix input at the l-th layer, $\hat{A} = D^{-\frac{1}{2}}AD^{-\frac{1}{2}}$ signifies the normalized adjacency matrix, D refers to the degree matrix, $W^{(l)} \in \mathbb{R}^{F\times F'}$ denotes the learnable weight matrix at the l-th layer, F' indicates the output feature dimensions, and σ represents the activation function, which is generally ReLU (Rectified Linear Unit).

Note that $H^{(0)}$ can be initialized with the protein feature matrix X_P. The PPI network encompasses characteristics such as Closeness Centrality, Betweenness Centrality, and Eigenvector Centrality for each individual protein, as delineated in (2), (3), (4), and (5), respectively. Through this graph convolution operation, the network can learn to aggregate local neighborhood information and update the node features iteratively, capturing deeper structural patterns in the PPI network.

We design a two-layer GCNs to extract features. $\sigma_{st}(i)$ is the number of those paths that pass through node i.

Degree Centrality (DC), Closeness Centrality (CC), Betweenness Centrality (BC), and Eigenvector Centrality (υ) are defined as

$$\mathrm{DC}_i = \deg(i), \tag{2}$$

$$\mathrm{CC}_i = \frac{1}{\sum_{j\neq i} d(i,j)}, \tag{3}$$

$$\mathrm{BC}_i = \sum_{s\neq i\neq t} \frac{\sigma_{st}(i)}{\sigma_{st}}, \tag{4}$$

$$\upsilon = \frac{1}{\lambda}A\upsilon, \tag{5}$$

where $deg(i)$ signifies the total connections node i has, $d(i,j)$ indicates the minimum distance that exists between nodes i and j, and the sum in the denominator considers all nodes except i itself. Additionally, σ_{st} represents the overall count of the shortest paths connecting nodes s and t, whereas $\sigma_{st}(i)$ specifically indicates the count of those paths that traverse through node i. Moreover, υ is the eigenvector that corresponds to the largest eigenvalue λ.

2.1.2. Graph attention layer (GAT layer)

The GAT Layer operates by dynamically adjusting the weights assigned to neighboring nodes during the process of aggregation, with the aim of enhancing the representation of the graph structure. The GAT layer takes as input the features produced by the GCN, denoted as $H^{(2)} \in \mathbb{R}^{N\times F'}$. This layer utilizes an attention-based mechanism to generate weighted feature representations, which allows the model to highlight and focus on the adjacent nodes that are most relevant. The representation of features obtained from the GAT layer is characterized as

$$h_i = \sigma\left(\sum_{j\in\mathcal{N}(i)} \alpha_{ij} W h_j\right), \tag{6}$$

where, h_i and h_j signify the input feature vectors corresponding to node i and one of its neighboring nodes j, respectively. W represents a trainable weight matrix that transforms these feature vectors. $\mathcal{N}(i)$ refers to the set of neighboring nodes that are directly linked to node i. Additionally, the term α_{ij} reflects the attention weight that establishes a connection between nodes i and j, indicating the significance of node j in relation to node i during the process of aggregating features.

The calculation of the attention coefficient is carried out by

$$\alpha_{ij} = \frac{\exp\left(\mathrm{LeakyReLU}\left(a^T[Wh_i \parallel Wh_j]\right)\right)}{\sum_{k\in\mathcal{N}(i)}\exp\left(\mathrm{LeakyReLU}\left(a^T[Wh_i \parallel Wh_k]\right)\right)}, \tag{7}$$

where a is a learnable weight vector in the attention mechanism, and $\parallel$ denotes the concatenation operation.

Ultimately, the characteristics identified by the GCN and GAT layers are subsequently directed into a fully connected neural network for additional analysis. The architecture of the neural network encompasses several layers that include linear mappings, ReLU activations, normalization techniques for batches, and dropout mechanisms intended for regularization. The resulting output generated by the network, referred to as FP, is calculated as

$$F_P = \mathrm{Dropout}\left(\sigma\left(\mathrm{BatchNorm}\left(H'\right)\right)\right), \tag{8}$$

where $F_P \in \mathbb{R}^{N\times(F'+F'')}$ signifies the ultimate feature depiction of the nodes, with H' representing the resultant feature matrix obtained from the GAT layer. The term BatchNorm refers to the technique of batch normalization, while σ denotes activation function. Additionally, Dropout functions as a means of regularization to help prevent overfitting.

2.2. Temporal feature extraction based on BiLSTM and attention mechanism from gene expression

Gene expression data inherently captures temporal dependencies and contextual relationships, which are crucial for understanding the dynamic patterns of gene expression changes and identifying key time points that play critical roles in biological processes. To effectively extract these features, we employ a combination of BiLSTM and attention mechanisms. The BiLSTM is utilized to capture both forward and backward temporal interdependencies, while the attention mechanism further enhances this process by focusing on the most relevant parts of the sequence, thereby enabling a more nuanced understanding of the temporal relationships within the gene expression data.

2.2.1. BiLSTM encoder

The expression levels of genes can be represented in a matrix format $X \in \mathbb{R}^{n\times T}$, where one dimension reflects the count of proteins and the other captures the time aspect of the dataset. The level of gene expression for the protein indexed by at the time point indexed by is represented by each entry.

The BiLSTM structure consists of two distinct LSTM networks: one analyzes the sequence in its natural order (the forward LSTM), while the other examines it in the reverse sequence (the backward LSTM). At each time step, the results produced by both the forward and backward networks are combined, enhancing the representation of bidirectional temporal information.

For every protein, hidden states are produced by the forward LSTM as it traverses the sequence from start to end, whereas the backward LSTM derives hidden states by processing the sequence in reverse order. The following equations describe these processes.

The Forward LSTM is characterized by

$$\begin{aligned} f_t &= \sigma(W_f \cdot [h_{t-1}, x_t] + b_f), \\ i_t &= \sigma(W_i \cdot [h_{t-1}, x_t] + b_i), \\ \tilde{C}_t &= \tanh(W_C \cdot [h_{t-1}, x_t] + b_C), \\ C_t &= f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t, \\ o_t &= \sigma(W_o \cdot [h_{t-1}, x_t] + b_o), \\ h_t &= o_t \cdot \tanh(C_t). \end{aligned} \tag{9}$$

The definition of Backward LSTM is presented as follows

$$\begin{aligned} \overleftarrow{f}_t &= \sigma(W_{\overleftarrow{f}} \cdot [\overleftarrow{h}_{t+1}, x_{T-t+1}] + b_{\overleftarrow{f}}), \\ \overleftarrow{i}_t &= \sigma(W_{\overleftarrow{i}} \cdot [\overleftarrow{h}_{t+1}, x_{T-t+1}] + b_{\overleftarrow{i}}), \\ \overleftarrow{\tilde{C}}_t &= \tanh(W_{\overleftarrow{C}} \cdot [\overleftarrow{h}_{t+1}, x_{T-t+1}] + b_{\overleftarrow{C}}), \\ \overleftarrow{C}_t &= \overleftarrow{f}_t \cdot \overleftarrow{C}_{t+1} + \overleftarrow{i}_t \cdot \overleftarrow{\tilde{C}}_t, \\ \overleftarrow{o}_t &= \sigma(W_{\overleftarrow{o}} \cdot [\overleftarrow{h}_{t+1}, x_{T-t+1}] + b_{\overleftarrow{o}}), \\ \overleftarrow{h}_t &= \overleftarrow{o}_t \cdot \tanh(\overleftarrow{C}_t). \end{aligned} \tag{10}$$

In this context, the gates responsible for forgetting, inputting, and outputting are identified as f_t, i_t, and o_t, while their corresponding counterparts in the reverse LSTM are noted as $\overleftarrow{f}_t$, $\overleftarrow{i}_t$, and $\overleftarrow{o}_t$. $\tilde{C}_t$ and C_t denote the candidate memory and cell state, while h_t and $\overleftarrow{h}_t$ signify the hidden states at the moment t. The weights W and biases b can be adjusted during the learning process, whereas the functions utilized here are the sigmoid and hyperbolic tangent activations.

For each time point, the hidden state that combines information from both directions is formed by merging the states from the forward and backward passes:

$$h_t = [h_t; \overleftarrow{h}_t],$$

where h_t offers a detailed encapsulation of the contexts leading up to and following time t. The concluding result generated by the BiLSTM encoder manifests as a matrix of features $H \in \mathbb{R}^{T \times 2 \times \text{hidden\_size}}$, capturing temporal attributes that are pertinent for subsequent applications.

2.2.2. The utilization of a multi-head attention framework enhances the capability to capture diverse contextual information

After deriving the hidden states H through the BiLSTM encoder, a mechanism known as Multi-Head Attention is utilized to encompass the various aspects of the sequential data. The MHA allows the model to focus on multiple parts of the sequence at the same time by utilizing several concurrent attention functions.

The attention output O_j for each specific attention head denoted by j is obtained through the following process.

$$Q_j = HW_j^Q, \quad K_j = HW_j^K, \quad V_j = HW_j^V \tag{11}$$

$$O_j = \text{softmax}\left(\frac{Q_j, K_j^\top}{\sqrt{d_k}}\right) V_j, \tag{12}$$

where Q_j, K_j, and V_j are the query, key, and value matrices for the j-th head, respectively; W_j^Q, W_j^K, and W_j^V are learnable weight matrices, and d_k is the dimensionality of the keys.

The outputs of all attention heads are then concatenated as

$$O = \text{Concat}(O_1, O_2, \ldots, O_h), \tag{13}$$

where h denotes the number of attention heads.

Subsequently, the results from all the attention heads are concatenated together, resulting in

$$F_A = OW^O, \tag{14}$$

where W^O is a learnable weight matrix, F_A has the same shape as H and represents the attended features for each protein.

2.2.3. Feature extraction by fully connected DNN

After obtaining the attended features from the Multi-Head Attention mechanism, these features are further processed through several fully connected layers to transform them into a suitable representation for subsequent tasks.

The final output from the last fully connected layer provides a feature representation suitable for subsequent tasks. The feature vector obtained after processing through the series of fully connected layers is

$$F_G = W_4 \cdot \text{ReLU}(W_3 \cdot \text{ReLU}(W_2 \cdot \text{ReLU}(W_1 \cdot F_A + b_1) + b_2) + b_3) + b_4. \tag{15}$$

2.3. Spatial feature extraction for essential proteins based on subcellular localization features

The subcellular localization data S is standardized to S', and then passed through a three-layer DNN for feature extraction. In the first layer, the feature representation $h_1 \in \mathbb{R}^{n \times u_1}$ is first obtained through a linear transformation $S'W_1 + b_1$, followed by a non-linear activation using the ReLU function as

$$h_1 = \text{ReLU}(S'W_1 + b_1).$$

Then, dropout is applied to h_1 to obtain h_1'.

The second layer applies a linear transformation and ReLU activation, followed by batch normalization as

$$h_2 = \text{Dropout}(\text{BatchNorm}(\text{ReLU}(h_1'W_2 + b_2)), p2).$$

In the third layer, another linear transformation, ReLU activation and a final dropout operation are performed to obtain the final feature representation.

$$F_S = \text{Dropout}(\text{ReLU}(h_2W_3 + b_3), p).$$

2.4. Feature fusion

To achieve essential protein prediction, we propose a feature fusion model based on self-attention and cross-attention mechanisms. This model integrates multiscale and multi-dimensional features from gene expression, subcellular localization, and protein networks to achieve deep feature fusion.

2.4.1. Self-attention mechanism

Firstly, self-attention mechanisms are applied to each input feature (F_G, F_S, F_P). For each type of feature F_i, the self-attended feature can be calculated as follows:

$$F_{SA}(F_i) = \text{Softmax}\left(\frac{F_iW_q(F_iW_k)^T}{\sqrt{d_k}}\right) F_iW_v, \tag{16}$$

where W_q, W_k, and W_v are trainable weight matrices, and d_k is a scaling factor used for stability.

2.4.2. Cross-attention mechanism

To further integrate information across different types of features, cross-attention mechanisms are employed. For example, a cross-attended feature can be created by integrating information from subcellular features (F_S) into gene features (F_G). This process is computed as follows:

$$F_{CA}(F_G, F_S) = \text{Softmax}\left(\frac{F_GW_q(F_SW_k)^T}{\sqrt{d_k}}\right) F_SW_v, \tag{17}$$

where the attention weights are computed by having the gene features (F_G) attend to the subcellular features (F_S).

Similarly, cross-attention mechanisms are applied in the opposite direction, where subcellular features (F_S) attend to gene features (F_G). Additionally, cross-attention is applied between gene and protein features, as well as between subcellular and protein features, in both directions.

2.4.3. Feature fusion

After extracting the self-attention and cross-attention features, the features are concatenated as follows:

$$\begin{aligned} F_{Fusion} = [&F_{SA}(F_G), F_{SA}(F_S), F_{SA}(F_P), \\ &F_{CA}(F_G, F_S), F_{CA}(F_S, F_P), \\ &F_{CA}(F_P, F_G), F_{CA}(F_S, F_G), \\ &F_{CA}(F_P, F_S), F_{CA}(F_G, F_P)]. \end{aligned} \tag{18}$$

2.5. Classification

The architecture utilizes residual blocks, which enhance the ability to extract features more deeply, resulting in improved performance for classification tasks. Initially, the concatenated features are forwarded through a sequence of residual blocks for processing. In each of the residual blocks, a layer that is fully connected utilizes ReLU activation along with dropout, and is followed by a shortcut connection that maintains the identity.

$$y_1 = \text{ResidualBlock}(F_{Fusion}) = \text{ReLU}(\text{Dropout}(\text{FC}(F_{Fusion}))) + F_{Fusion} \tag{19}$$

Following the completion of the residual blocks, the resultant output is then sent through further fully connected layers to condense the features into a single numerical value. Subsequently, it is subjected to a Sigmoid activation function.

$$\text{Output} = \text{Sigmoid}(\text{FC}(y_1)) \tag{20}$$

2.6. Evaluation metrics

The performance of the AttentionEP is evaluated by using the AUC obtained from the analysis of the Receiver Operating Characteristic (ROC).

$$\text{AUC} = \int_0^1 \text{TPR}(FPR)\, d(FPR). \tag{21}$$

In this context, TPR refers to the rate of correctly identified positive cases, while FPR indicates the rate of incorrectly identified negative cases.

3. Results

3.1. Datasets

The interaction data related to proteins was sourced from the BioGRID repository (https://thebiogrid.org/) [25]. Once duplicate entries were eliminated, the dataset pertaining to yeast protein-protein interactions comprises a total of 5,616 records. In order to pinpoint the necessary proteins within the network, information was gathered from various sources including SSGD [26], MIPS [27], DEG [28], and SGDP [29], leading to the identification of a total of 1,199 proteins that are deemed essential. In order to obtain the gene expression data relevant to these proteins, information was collected from the Gene Expression Omnibus (GEO) platform (https://www.ncbi.nlm.nih.gov/geo/) [30]. Furthermore, information regarding the subcellular localization was obtained from the COMPARTMENTS database (https://compartments.jensenlab.org/) [31].

Table 1
Impact of different learning rates and batch sizes on the proposed model performance.

Learning Rate	Bath size	AUC
0.5	128	0.9290
0.1	128	0.9026
0.01	128	0.9026
0.05	128	0.9195
0.001	128	0.9210
0.0001	128	0.9433
0.0001	16	0.9364
0.0001	32	0.9413
0.0001	64	0.9422
0.0001	128	0.9433
0.0001	256	0.9420
0.0001	512	0.9409
0.0001	1024	0.9388

Table 2
The influence of dropout rates on the performance of the proposed model.

Learning Rate	Bath size	drop out	AUC
0.0001	128	0.1	0.9354
0.0001	128	0.3	0.9385
0.0001	128	0.5	**0.9433**
0.0001	128	0.7	0.9412
0.0001	128	0.9	0.9413

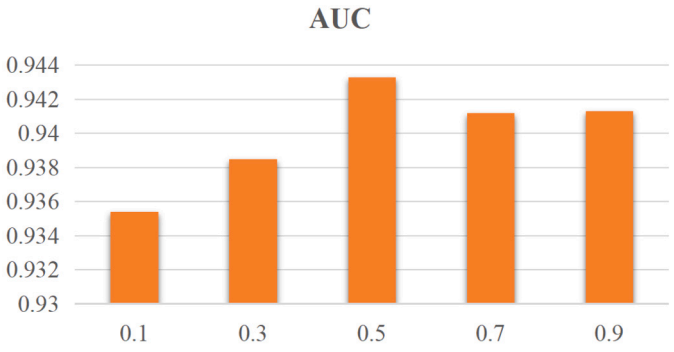

Fig. 2. Impact of dropout rates on the proposed model performance.

3.2. Configuration of the hyperparameters

To investigate how various adjustments to hyperparameters affect the efficacy of the suggested model, modifications were made to the learning rate, batch size, and dropout rate. The study focused on manipulating these hyperparameters while measuring the AUC scores across different combinations to determine the best settings.

Initially, the effects of different learning rates and batch sizes on the performance of the model were assessed, as illustrated in Table 1. The findings suggest that a decreased learning rate tends to enhance the AUC metric. In particular, the optimal AUC of 0.9433 was attained when the learning rate was configured at 0.0001 with a batch size of 128. While increasing the batch sizes resulted in a slower training timeframe, this change did not have a meaningful impact on the overall performance of the model.

Subsequently, we investigated how various dropout rates influenced the performance of the model, as summarized in Table 2 and depicted in Fig. 2. The model exhibited its peak AUC of 0.9416 with a dropout rate of 0.5, suggesting that an optimal dropout rate serves to mitigate overfitting while boosting the model's generalization capabilities. Yet, further increases in the dropout rate did not yield significant enhancements in performance and, in certain instances, may have led to minor declines.

Table 3
Comparison of Fused Features Performance against Conventional and Deep Learning Models.

Model	AUC
SVM	0.9166
RF	0.9355
AdaBoost	0.9320
EP-EDL	0.9365
AttentionEP	**0.9433**

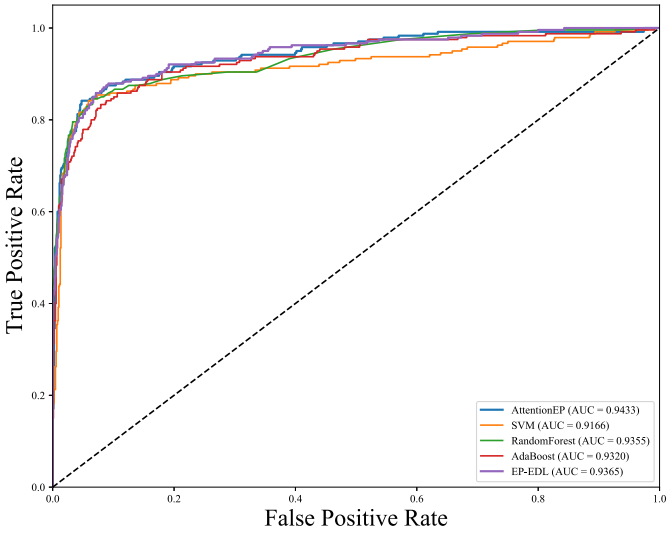

Fig. 3. The ROC plots for the AttentionEP model and other models.

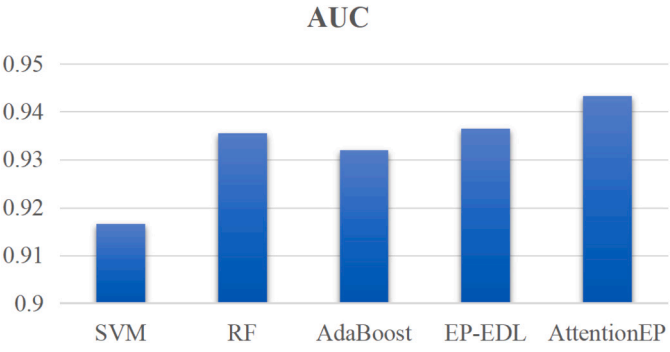

Fig. 4. Comparison results with alternative machine learning approaches.

To conclude, the analysis conducted on the hyperparameters revealed that the optimal performance of the model is achieved with a learning rate set at 0. 0001, a mini-batch size of 128, along with a dropout rate set at 0. 5. This setup served as the basis for training and evaluating the model in later trials.

3.3. Performance of fused features with traditional and deep learning models

In order to evaluate how well the fused features are integrated and to compare the performance of the AttentionEP model with others, experiments were performed utilizing various machine learning techniques, such as Support Vector Machines (SVM), Random Forests (RF), AdaBoost, and EP-EDL [21]. A summary of the performance evaluations for these algorithms is provided in Table 3 and illustrated in Figs. 3 and 4 showcasing the AUC metrics corresponding to each model.

Table 4
Effectiveness of Individual Features with Conventional Models.

Feature	SVM	RF	AdaBoost
Gene Expression Feature	0.6318	0.6993	0.7074
Subcellular Localization Feature	0.9082	0.9293	0.9309
PPI Feature	0.6250	0.6697	0.6947
Fused Feature	0.9166	0.9355	0.9320

Table 5
The comparison of performance among various feature combinations.

Feature	AUC
G	0.7191
S	0.9208
P	0.6629
G+S	0.9010
G+P	0.6743
S+P	0.8982
Cross(G,S)	0.9188
Cross(P,S)	0.6662
Cross(G,P)	0.7145
Cross(G,S)+Cross(P,S)	0.9053
Cross(G,S)+Cross(G,P)	0.9065
Cross(P,S)+Cross(G,P)	0.6597
All	0.9433

Table 3 and Figs. 3 and 4 demonstrate that AttentionEP secures the top AUC score of 0.9433 exceeding the performance of established models such as SVM, RF, and AdaBoost, as well as EP-EDL, which recorded an AUC of 0. 9365.

Notably, the data in Table 4 indicates that conventional machine learning approaches, when utilizing single features for training, exhibit markedly inferior performance in comparison to those that leverage feature integration. For instance, when gene expression data is solely used in training an SVM, the resulting AUC value stands at 0.6318. When the same model utilizes combined features, the AUC improves to 0.9166. This highlights the essential impact of integrating features to enhance the effectiveness of the model.

In addition, the AttentionEP model boosts its effectiveness through the implementation of cutting-edge deep learning frameworks, incorporating residual networks along with attention-based methods, which enables it to more effectively capture the interplay among various features and attain enhanced accuracy levels. The evaluation against EP-EDL indicates that AttentionEP outperforms other models that rely on deep learning, thanks to its innovative approach to feature integration and tailored design.

3.4. Ablation experiments

In this section, we conducted ablation experiments to evaluate the impact of various feature combinations on classification performance. Table 5 and Fig. 5 present the performance of various feature combinations.

When using individual features, the AUC values for gene expression self-attention features (G), subcellular localization self-attention features (S), and PPI self-attention features (P) were 0.7191, 0.9208, and 0.6629, respectively (see Table 5). Among these, the subcellular localization feature (S) performed the best, indicating its significant contribution to the classification task. However, single features alone did not achieve optimal classification performance.

For feature combinations, the AUC values for G+S and S+P were 0.9010 and 0.8982, respectively. Although these combinations showed improved performance, they did not surpass the performance of the individual subcellular localization feature (S). This suggests that simple feature combinations do not fully exploit the complementary information between features. Notably, the G+P combination performed the

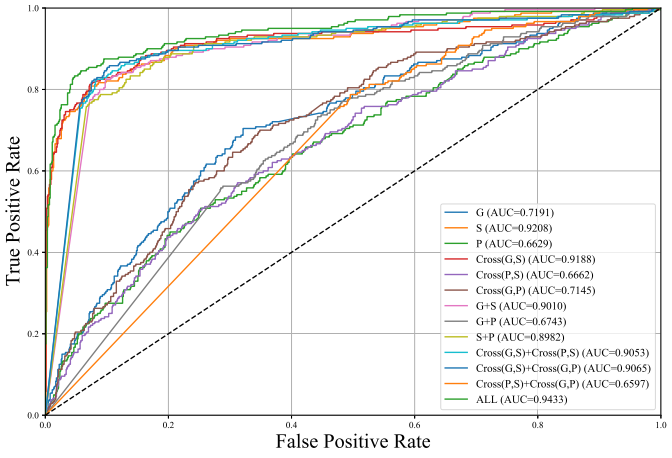

Fig. 5. The ROC plots of various feature combinations.

worst (AUC = 0.6743), indicating limited contribution of this feature pair to the classification task.

The introduction of cross-attention features significantly improved classification performance. For instance, Cross(G,S) achieved an AUC of 0.9188, close to the performance of the individual subcellular localization feature (S). The reason for this improvement lies in the strong functional complementarity between gene expression data (G) and subcellular localization data (S). Gene expression provides direct insights into protein activity, which aligns well with subcellular data that indicates where proteins are active within the cell. This fusion allows the model to extract more biologically relevant information, enhancing performance. However, other cross-attention features, such as Cross(P,S) and Cross(G,P), exhibited relatively lower AUC values, possibly due to weaker complementary relationships between these features. For example, the Cross(P,S) combination underperforms because the structural relationships in protein interaction data (P) may not align well with the spatial characteristics of subcellular localization (S), leading to less effective feature fusion.

Ultimately, the model incorporating all features and their cross-attention features (G + S + P + Cross(G,S) + Cross(P,S) + Cross(G,P)) achieved the highest classification performance, with an AUC of 0.9433. This result indicates that single features or simple feature combinations do not provide the best performance. Instead, combining all features and their cross-attention mechanisms maximizes the complementary information between features, leading to a significant improvement in classification performance. This finding underscores the importance of comprehensive feature fusion and cross-attention mechanisms in complex classification tasks.

The results demonstrate that neither single features nor simple combinations of two features achieved the best performance. Instead, the use of cross-attention mechanisms significantly enhanced model performance. This study demonstrates that single features or simple feature combinations are insufficient for optimal performance. The introduction of cross-attention mechanisms significantly enhances model capability. Ultimately, combining all features and their cross-attention mechanisms achieves the best classification performance, providing strong support for future feature fusion and model optimization efforts.

3.5. Evaluation of GCN and GAT layers through ablation analysis

To justify the combination of GCN and GAT layers used in the PPI extraction channel, while keeping the processing of other data sources and channels unchanged, we conducted an ablation study. The results (Table 6) demonstrate the performance of different configurations based on the AUC metric.

The ablation study shows that the combination of GCN and GAT layers yields the highest AUC score (0.9433), indicating that the integration of both layers enhances the model's performance compared to using GCN or GAT individually. Specifically, the GCN-only model achieves an AUC of 0.9308, while the GAT-only model achieves an AUC of 0.9397.

Table 6
Evaluation of AUC for GCN and GAT Layer Ablation.

Layer Configuration	AUC
GCN	0.9308
GAT	0.9397
GCN + GAT	0.9433

Table 7
Evaluation of AUC for BiLSTM and MHA Layer Ablation.

Layer Configuration	AUC
BiLSTM	0.9236
MHA	0.9278
BiLSTM + MHA	0.9433

These results support our design choice to combine GCN and GAT layers, as the combination allows the model to leverage both local structural information (captured by GCN) and fine-grained dependencies between nodes (captured by GAT). This synergy results in improved performance, highlighting the complementary nature of these two approaches in graph-based learning tasks.

3.6. Evaluation of BiLSTM and MHA layers through ablation analysis

To specifically evaluate the impact of different mechanisms used in the gene expression data extraction channel, while keeping the processing of other data sources and channels unchanged, we conducted an ablation study. This experiment focuses solely on the gene expression feature extraction framework, isolating the contributions of BiLSTM and MHA mechanisms. The rest of the model, including other data channels such as subcellular localization and protein interaction networks, remained unchanged during the evaluation to ensure that the observed effects are solely attributable to the changes in the gene expression feature extraction channel.

The results of this ablation study (Table 7) show the AUC scores for different configurations of the gene expression data extraction framework.

In this experiment, we observe that the combination of BiLSTM and MHA achieves the best performance with an AUC of 0.9433, significantly outperforming configurations that use only BiLSTM or only MHA. The BiLSTM-only configuration results in an AUC of 0.9236, while the MHA-only configuration achieves a slightly higher AUC of 0.9278. These findings highlight that while both BiLSTM and MHA independently contribute to improving the feature extraction process, their combined use leads to a more comprehensive representation of the gene expression data, thus resulting in superior performance.

BiLSTM is responsible for capturing the temporal dependencies within the gene expression data by processing the sequences in both forward and backward directions. This ensures that both past and future contexts are taken into account when learning the feature representations.

MHA focuses on capturing relationships between different time steps in the gene expression data through its attention mechanism, allowing the model to weigh different time points based on their relevance and capture more complex interactions.

By keeping all other data channels—such as subcellular localization and protein network feature extraction—unchanged, this ablation study isolates the effect of different architectures on the gene expression feature extraction framework. The results suggest that the integration of both BiLSTM and MHA is crucial for effectively modeling the temporal

Table 8
The performance of AttentionEP.

Accuracy	Precision	Recall	F1 Score	Specificity	AUC
0.9288	0.8874	0.7750	0.8230	0.9706	0.9433

and contextual relationships in gene expression data, leading to improved overall model performance.

3.7. Model performance evaluation

In addition to the widely used AUC, we computed several other common classification metrics to provide a more comprehensive assessment of our protein prediction model. These metrics include Accuracy, Precision, Recall, F1 Score, and Specificity, offering a deeper understanding of the model's performance, particularly when addressing class imbalance.

As shown in Table 8, the model achieved an Accuracy of 0.9288, indicating that it correctly classifies the majority of the samples. Precision, which measures the reliability of positive predictions, is 0.8874, showing that the model has a high level of accuracy in identifying true positives. Meanwhile, Recall (Sensitivity) stands at 0.775, demonstrating the model's capability to recognize most positive samples.

The F1 Score, which balances Precision and Recall, is 0.8230, suggesting that the model performs well in imbalanced datasets. Additionally, the model achieves a Specificity of 0.9706, highlighting its strong ability to accurately identify negative samples. These results demonstrate the model's overall robustness and effectiveness in the protein prediction task.

4. Conclusions

In this study, we proposed a novel essential protein prediction method that leverages the power of cross-attention and self-attention mechanisms to integrate multiscale and multi-domain features. By extracting spatial features from PPI data using GCN and GAT, temporal features from gene expression data using BiLSTM, and spatial-related features from subcellular localization information using DNN, we achieved a comprehensive representation of proteins.

Our approach demonstrated superior performance in essential protein prediction, with the model achieving an AUC score of 0.9433, significantly outperforming traditional machine learning models such as SVM, RF, and AdaBoost. The ablation experiments further highlighted the importance of feature integration, showing that individual features or simple combinations were insufficient for optimal performance. Instead, the introduction of cross-attention mechanisms significantly enhanced the model's ability to capture complex patterns and interactions within the data, leading to substantial improvements in classification accuracy.

These findings underscore the critical role of advanced attention mechanisms and comprehensive feature fusion in computational biology. The success of our method not only demonstrates the effectiveness of integrating diverse biological data but also provides a foundation for future research aimed at optimizing protein prediction models. This study contributes valuable insights into the development of more accurate and robust computational tools for essential protein identification, with potential applications in understanding disease mechanisms and developing targeted therapies.

Future work should explore how the model can be adapted or fine-tuned for different species, especially for those with more complex biological systems. By doing so, the research can extend its impact and offer a valuable tool for studying essential protein prediction across a wider range of organisms.

CRediT authorship contribution statement

Chuanyan Wu: Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Conceptualization. **Bentao Lin:** Writing – original draft, Visualization, Validation, Methodology, Conceptualization. **Jialin Zhang:** Resources, Investigation. **Rui Gao:** Writing – review & editing, Conceptualization. **Rui Song:** Writing – review & editing, Project administration, Conceptualization. **Zhi-Ping Liu:** Writing – review & editing, Supervision, Methodology, Investigation, Funding acquisition, Conceptualization.

Declaration of competing interest

The authors have no competing interests to declare.

Acknowledgements

This work was supported in part by the Natural Science Foundation of China under Grant number 62373216. This work was supported in part by the Doctoral Starting up Foundation of Shandong Management University under Grant number SDMUD202101, PD2020A17, PA2021A23 and QH2021Z02.

References

[1] Arfin S, Jha NK, Jha SK, Kesari KK, Ruokolainen J, Roychoudhury S, et al. Oxidative stress in cancer cell metabolism. Antioxidants 2021;10(5):642.
[2] Lodish HF. Molecular cell biology. Macmillan; 2008.
[3] Lu X, Wang X, Ding L, Li J, Gao Y, He K. frdriver: a functional region driver identification for protein sequence. IEEE/ACM Trans Comput Biol Bioinform 2020;18(5):1773–83.
[4] Hart GT, Ramani AK, Marcotte EM. How complete are current yeast and human protein-interaction networks? Genome Biol 2006;7:1–9.
[5] Lu X, Qian X, Li X, Miao Q, Peng S. Dmcm: a data-adaptive mutation clustering method to identify cancer-related mutation clusters. Bioinformatics 2019;35(3):389–97.
[6] Zhang W, Xue X, Xie C, Li Y, Liu J, Chen H, et al. Cegso: boosting essential proteins prediction by integrating protein complex, gene expression, gene ontology, subcellular localization and orthology information. Interdiscip Sci Comput Life Sci 2021;13:349–61.
[7] Boopathi V, Subramaniyam S, Malik A, Lee G, Manavalan B, Yang D-C. macppred: a support vector machine-based meta-predictor for identification of anticancer peptides. Int J Mol Sci 2019;20(8):1964.
[8] Ao C, Zhou W, Gao L, Dong B, Yu L. Prediction of antioxidant proteins using hybrid feature representation method and random forest. Genomics 2020;112(6):4666–74.
[9] Wang N, Zeng M, Li Y, Wu F-X, Li M. Essential protein prediction based on node2vec and xgboost. J Comput Biol 2021;28(7):687–700.
[10] Wu C, Gao R, Zhang Y, De Marinis Y. Ptpd: predicting therapeutic peptides by deep learning and word2vec. BMC Bioinform 2019;20:1–8.
[11] Zhang W, Xu J, Zou X. Predicting essential proteins by integrating network topology, subcellular localization information, gene expression profile and go annotation data. IEEE/ACM Trans Comput Biol Bioinform 2020;17(6):2053–61. https://doi.org/10.1109/TCBB.2019.2916038.
[12] Li X, Li W, Zeng M, Zheng R, Li M. Network-based methods for predicting essential genes or proteins: a survey. Brief Bioinform 2020;21(2):566–83.
[13] Wu C, Lin B, Shi K, Zhang Q, Gao R, Yu Z, et al. Peprf: identification of essential proteins by integrating topological features of ppi network and sequence-based features via random forest. Curr Bioinform 2021;16(9):1161–8.
[14] Jeong H, Mason SP, Barabási A-L, Oltvai ZN. Lethality and centrality in protein networks. Nature 2001;411(6833):41–2.
[15] Yu H, Greenbaum D, Lu HX, Zhu X, Gerstein M. Genomic analysis of essentiality within protein networks. Trends Genet 2004;20(6):227–31.
[16] Zhang Z, Ruan J, Gao J, Wu F-X. Predicting essential proteins from protein-protein interactions using order statistics. J Theor Biol 2019;480:274–83.
[17] Tang X, Wang J, Zhong J, Pan Y. Predicting essential proteins based on weighted degree centrality. IEEE/ACM Trans Comput Biol Bioinform 2013;11(2):407–18.
[18] Li M, Li W, Wu F-X, Pan Y, Wang J. Identifying essential proteins based on sub-network partition and prioritization by integrating subcellular localization information. J Theor Biol 2018;447:65–73.
[19] Zhong J, Tang C, Peng W, Xie M, Sun Y, Tang Q, et al. A novel essential protein identification method based on ppi networks and gene expression data. BMC Bioinform 2021;22(1):248.
[20] Zeng M, Li M, Wu F-X, Li Y, Pan Y. Deepep: a deep learning framework for identifying essential proteins. BMC Bioinform 2019;20:1–10.

[21] Li Y, Zeng M, Wu Y, Li Y, Li M. Accurate prediction of human essential proteins using ensemble deep learning. IEEE/ACM Trans Comput Biol Bioinform 2021;19(6):3263–71.
[22] Zeng M, Li M, Fei Z, Wu F-X, Li Y, Pan Y, et al. A deep learning framework for identifying essential proteins by integrating multiple types of biological information. IEEE/ACM Trans Comput Biol Bioinform 2019;18(1):296–305.
[23] Yue Y, Ye C, Peng P-Y, Zhai H-X, Ahmad I, Xia C, et al. A deep learning framework for identifying essential proteins based on multiple biological information. BMC Bioinform 2022;23(1):318.
[24] Li Y, Zeng M, Zhang F, Wu F-X, Li M. Deepcelless: cell line-specific essential protein prediction with attention-based interpretable deep learning. Bioinformatics 2023;39(1):btac779.
[25] Chatr-Aryamontri A, Oughtred R, Boucher L, Rust J, Chang C, Kolas NK, et al. The biogrid interaction database: 2017 update. Nucleic Acids Res 2017;45(D1):D369–79.
[26] Cherry JM, Adler C, Ball C, Chervitz SA, Dwight SS, Hester ET, et al. Sgd: saccharomyces genome database. Nucleic Acids Res 1998;26(1):73–9.
[27] Mewes H-W, Frishman D, Güldener U, Mannhaupt G, Mayer K, Mokrejs M, et al. Mips: a database for genomes and protein sequences. Nucleic Acids Res 2002;30(1):31–4.
[28] Zhang R, Lin Y. Deg 5.0, a database of essential genes in both prokaryotes and eukaryotes. Nucleic Acids Res 2009;37(suppl_1):D455–8.
[29] Winzeler EA, Shoemaker DD, Astromoff A, Liang H, Anderson K, Andre B, et al. Functional characterization of the s. cerevisiae genome by gene deletion and parallel analysis. Science 1999;285(5429):901–6.
[30] Tu BP, Kudlicki A, Rowicka M, McKnight SL. Logic of the yeast metabolic cycle: temporal compartmentalization of cellular processes. Science 2005;310(5751):1152–8.
[31] Binder JX, Pletscher-Frankild S, Tsafou K, Stolte C, O'Donoghue SI, Schneider R, et al. Compartments: unification and visualization of protein subcellular localization evidence. Database 2014;2014:bau012.