

Accurate prediction of essential proteins using ensemble machine learning

Dezhi Lu(鲁德志), Hao Wu(吴淏), Yutong Hou(侯俞彤), Yuncheng Wu(吴云成), Yuanyuan Liu(刘媛媛), and Jinwu Wang(王金武)

Citation: Chin. Phys. B, 2025, 34 (**1**): 018901. DOI: 10.1088/1674-1056/ad8db2

What follows is a list of articles you may be interested in

Impact of environmental factors on the coevolution of information-emotions-epidemic dynamics in activity-driven multiplex networks

Liang'an Huo(霍良安), Bingjie Liu(刘炳杰), and Xiaomin Zhao(赵晓敏)
Chin. Phys. B, 2024, 33 (**12**): 128903. DOI: 10.1088/1674-1056/ad7df5

Source localization in signed networks with effective distance

Zhi-Wei Ma(马志伟), Lei Sun(孙蕾), Zhi-Guo Ding(丁智国), Yi-Zhen Huang(黄宜真), and Zhao-Long Hu(胡兆龙)
Chin. Phys. B, 2024, 33 (**2**): 028902. DOI: 10.1088/1674-1056/ad1482

Essential proteins identification method based on four-order distances and subcellular localization information

Pengli Lu(卢鹏丽), Yu Zhong(钟雨), and Peishi Yang(杨培实)
Chin. Phys. B, 2024, 33 (**1**): 018903. DOI: 10.1088/1674-1056/acd7ca

AG-GATCN: A novel method for predicting essential proteins

Peishi Yang(杨培实), Pengli Lu(卢鹏丽), and Teng Zhang(张腾)
Chin. Phys. B, 2023, 32 (**5**): 058902. DOI: 10.1088/1674-1056/acb9f9

Biased random walk with restart for essential proteins prediction

Pengli Lu(卢鹏丽), Yuntian Chen(陈云天), Teng Zhang(张腾), and Yonggang Liao(廖永刚)
Chin. Phys. B, 2022, 31 (**11**): 118901. DOI: 10.1088/1674-1056/ac7a17

SPECIAL TOPIC — Computational programs in complex systems

Accurate prediction of essential proteins using ensemble machine learning

Dezhi Lu(鲁德志)[1,†], Hao Wu(吴浩)[1,†], Yutong Hou(侯俞彤)[2], Yuncheng Wu(吴云成)[3], Yuanyuan Liu(刘媛媛)[1,‡], and Jinwu Wang(王金武)[1,2,§]

[1] *School of Medicine, Shanghai University, Shanghai* 200444, *China*
[2] *Shanghai Key Laboratory of Orthopaedic Implants, Department of Orthopaedic Surgery, Shanghai Ninth People's Hospital, Shanghai Jiao Tong University School of Medicine, Shanghai* 200011, *China*
[3] *University of Shanghai for Science and Technology, Shanghai* 200093, *China*

(Received 15 September 2024; revised manuscript received 21 October 2024; accepted manuscript online 1 November 2024)

Essential proteins are crucial for biological processes and can be identified through both experimental and computational methods. While experimental approaches are highly accurate, they often demand extensive time and resources. To address these challenges, we present a computational ensemble learning framework designed to identify essential proteins more efficiently. Our method begins by using node2vec to transform proteins in the protein–protein interaction (PPI) network into continuous, low-dimensional vectors. We also extract a range of features from protein sequences, including graph-theory-based, information-based, compositional, and physiochemical attributes. Additionally, we leverage deep learning techniques to analyze high-dimensional position-specific scoring matrices (PSSMs) and capture evolutionary information. We then combine these features for classification using various machine learning algorithms. To enhance performance, we integrate the outputs of these algorithms through ensemble methods such as voting, weighted averaging, and stacking. This approach effectively addresses data imbalances and improves both robustness and accuracy. Our ensemble learning framework achieves an AUC of 0.960 and an accuracy of 0.9252, outperforming other computational methods. These results demonstrate the effectiveness of our approach in accurately identifying essential proteins and highlight its superior feature extraction capabilities.

Keywords: protein–protein interaction (PPI), essential proteins, deep learning, ensemble learning

PACS: 89.75.–k

DOI: 10.1088/1674-1056/ad8db2 **CSTR:** 32038.14.CPB.ad8db2

1. Introduction

The protein–protein interaction (PPI) network is a fundamental tool in bioinformatics, offering valuable insights into the functions and interactions of proteins.[1,2] Within the network, pathogens tend to target highly connected proteins, as a single protein can impact numerous others.[3] These central proteins, often referred to as essential, are critical components of the network and represent promising targets for drug development due to their pivotal roles.[4] Accurately identifying these essential proteins is crucial for pinpointing potential therapeutic targets for various diseases. While traditional experimental methods have been employed to locate these key proteins, they are often complex, costly, and time-consuming. The advent of high-throughput data has introduced machine learning techniques as an effective alternative. These methods not only complement experimental approaches but also enhance the identification process by providing additional insights.[5]

Researchers have employed feature selection techniques to extract key features from protein data and other biological information, which are then used with machine learning algorithms to identify essential proteins. Prominent algorithms in this area include random forests (RF), decision trees (DT), and extreme gradient boosting (XGBoost).[6–8] These methods leverage biological data to train classifiers that pinpoint essential proteins. For example, Wu *et al.*[9] developed a model that identifies crucial proteins by analyzing topological features from PPI networks and incorporating protein sequence information. Their PEPRF model proved effective in this regard. In a similar vein, Zhong *et al.*[10] applied a feature selection approach using SVM-RFE, which utilized 26 features related to centrality and subcellular localization to predict essential proteins. Recently, deep learning techniques have emerged as more powerful alternatives to traditional methods.[11] Researchers have begun using these advanced techniques for protein prediction. Notably, Schapke *et al.*[12] introduced EPGAT, an attention-based graph neural network model capable of processing complex, non-Euclidean data. This model has

[†]These authors contributed equally to this work.
[‡]Corresponding author. E-mail: yuanyuan_liu@shu.edu.cn
[§]Corresponding author. E-mail: wangjw@shsmu.edu.cn

© 2025 Chinese Physical Society and IOP Publishing Ltd. All rights, including for text and data mining, AI training, and similar technologies, are reserved.

shown significant improvements over conventional network-based and shallow machine learning approaches.

When developing machine learning models for predicting essential proteins, two key challenges often arise.[13] The first challenge is the complexity and high dimensionality of protein sequence data, which can impede accurate predictions. Utilizing evolutionary information can help overcome this by simplifying the dataset, reducing its complexity, and improving predictive performance. The second challenge is class imbalance, as essential proteins are generally much less common than non-essential proteins. This imbalance can negatively impact model performance and hinder the accurate identification of essential proteins. Ensemble learning can address this issue by combining multiple models to enhance the prediction accuracy and reduce bias towards the majority class. Techniques such as bagging, boosting, and stacking leverage the strengths of various classifiers, offering a more balanced approach and improving the model's ability to effectively identify essential proteins.

Hence, we develope an ensemble machine learning framework that integrates both protein sequences and PPI data. Firstly, we use node2vec to convert proteins in the PPI network into continuous low-dimensional vectors. Evolutionary information is vital because it closely correlates with protein essentiality. We apply deep learning techniques to extract evolutionary information from high-dimensional position-specific scoring matrices (PSSMs). By employing multi-scale convolutional neural networks (CNNs), we enhance the ability to identify essential features in protein sequences, surpassing previous sequence-based methods. By employing a diverse array of feature extraction methods, it is possible to capture a more comprehensive spectrum of valuable information, thereby enhancing the overall integrity of the data. This facilitates the identification of the most pertinent features for classification, ultimately leading to a marked improvement in the model's precision. We combine these features for classification using a range of machine learning algorithms, including SVM, multilayer perceptron (MLP), AdaBoost, RF, extra trees (ET), XGBoost, gradient boosting (GB), and LightGBM. We merge the outputs of these algorithms using ensemble techniques such as voting, weighted averaging, and stacking. Ensemble learning effectively integrates the predictive outcomes of diverse algorithms, thereby mitigating the biases that may arise from relying on a singular model. This approach demonstrates superior performance in the context of data imbalance, enhancing both the robustness and generalizability of the model. The experimental outcomes indicate that our framework improves the classifier's performance, accuracy, and robustness while addressing class imbalance issues in essential protein predictions, resulting in more precise and reliable outcomes.

2. Materials and methods

2.1. Network architecture

As shown in Fig. 1, this is the structure of our proposed ensemble learning framework, which comprises two main components: feature extraction and classification. The model accepts two inputs: the protein expression profile and the PPI network.

During the feature extraction phase, we first analyze the PPI network using the node2vec method to capture the topological characteristics of each protein. Next, we process the protein expression profiles by normalizing them and applying an attention pooling technique to reduce the risk of overfitting. This technique highlights the most relevant features by assigning them greater importance. The results from both the PPI network and the protein expression profiles are then integrated through pointwise convolution, which combines information from different sources. In the classification phase, the combined features from the protein expression profiles and the PPI network are fed into two fully connected layers. To tackle the issue of imbalanced distribution of essential proteins, we employ an ensemble learning approach, which enhances the model's performance by leveraging multiple learning algorithms.

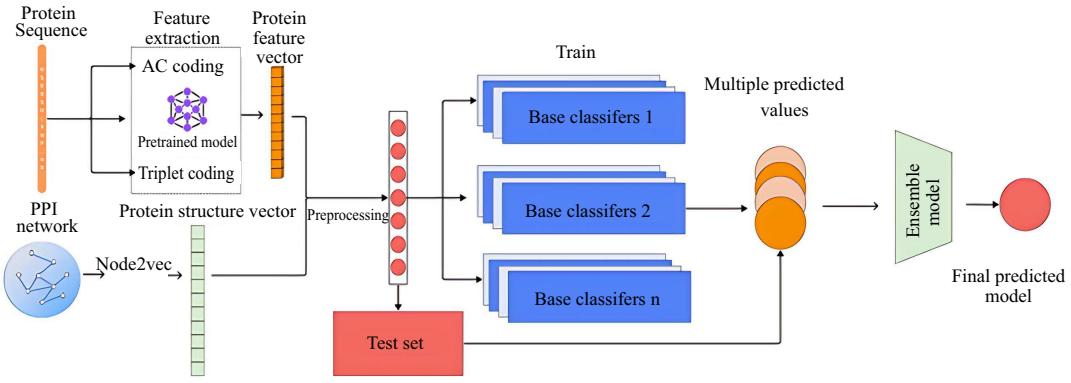


Fig. 1. The training process of ensemble learning framework.

2.2. Protein sequence representation

Similar to sentences in text, protein sequences are ordered lists of fundamental units that convey intricate information. The protein sequence consists of a chain of amino acids, which can be compared to a "sentence" made up of "words." To facilitate effective sequence encoding and modeling, we analyze the characteristics of these sequences. In this context, Shen *et al.*[14] introduced a triad technique for representing protein sequences (Tabel 1). This method transforms both the structural and functional information of the sequences into numerical features, enabling a more effective analysis of the data.

Table 1. Amino acid grouping.

Group 1	Group 2	Group 3	Group 4	Group 5	Group 6	Group 7
C	D, E	K, R	A, G, V	H, N, Q, W	M, S, T, Y	F, I, L, P

The auto-covariance (AC) method transforms amino acid sequences into a matrix that reflects the spatial relationships between residues at various distances. This technique is frequently employed in protein coding. For a protein sequence x of length L, the calculation formula for the AC method is as follows:

$$\mathrm{AC}(\mathrm{lag}, j) = \frac{1}{L-\mathrm{lag}} \sum_{i=1}^{L-\mathrm{lag}} \left(X_{ij} - \frac{1}{L} \sum_{i=1}^{L} X_{ij} \right) \times \left(X_{(i+\mathrm{lag}),j} - \frac{1}{L} \sum_{i=1}^{L} X_{ij} \right).$$

We use PSSMs to represent protein sequences. PSSMs offer a more nuanced view compared to raw sequences. In a PSSM, each amino acid position is represented by a 20-dimensional vector that indicates how conserved the position is relative to mutations in other amino acids. Highly conserved positions receive higher scores, while less conserved ones get lower scores. Studies have demonstrated that PSSMs are effective for various biological tasks, such as predicting protein function, identifying binding sites, and determining subcellular localization.[13,15]

2.3. Extract the features of the PPI network by node2vec

In general, a raw PPI network can be represented as an undirected graph $G(V,E)$, which consists of a set of nodes $V = \{v_1, v_2, v_3, \ldots, v_n\}$ and edges $E = \{e(v_i, v_j)\}$. Each node $v_i \in V$ refers to a protein in the PPI network, and each edge $e(v_i, v_j) \in E$ refers to the interaction between pairs of proteins v_i and v_j.

Given the essential role of the PPI network in predicting crucial proteins, it is vital to choose an effective method for extracting topological features. Peng *et al.*[16] proposed the DSEP algorithm to identify the essential proteins, which construct a topological information matrix based on the four-order distances of the PPI network. Traditional manual feature selection techniques have limitations, such as they can be computationally intensive, posing challenges when dealing with large-scale network data. The node2vec,[13] introduced in Fig. 2, allows for the extraction of topological features from vertices in a network. Specifically, we use the node2vec method,[17] which maps nodes to a low-dimensional feature space, allowing us to capture the topological features of the PPI network more effectively. By using the node2vec algorithm, we are able to derive more comprehensive features compared to traditional centrality methods. As a result, we obtain 128 distinct features for each protein in the PPI network.

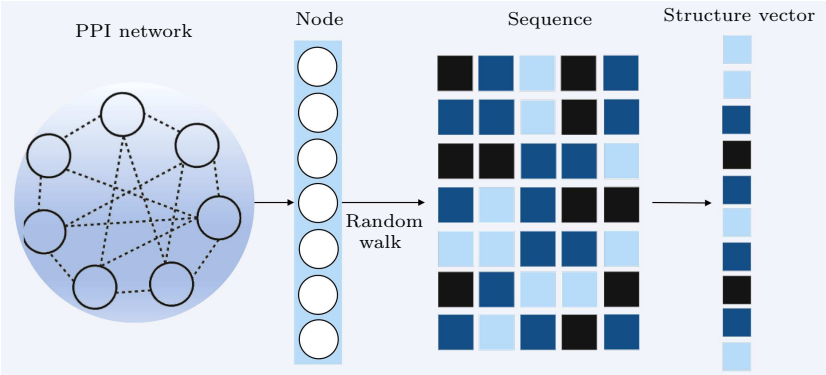


Fig. 2. PPI network processing process.

2.4. Feature extraction through deep learning

Deep learning has emerged as a prominent method for feature extraction, demonstrating significant success across various fields, including computer vision, natural language processing, and bioinformatics.[18,19] Research indicates that the features related to protein essentiality, which are generated automatically through deep learning techniques, outperform traditional handcrafted features in terms of efficiency.[20]

Many studies have demonstrated the successful application of PSSMs in various biological computational tasks, including function prediction and subcellular localization prediction.[21] In our study, we utilize PSSMs to represent protein sequence features. Essential proteins, which play critical roles in cellular functions, generally evolve more slowly than

non-essential proteins, showing a higher degree of evolutionary conservation. PSSMs are adept at capturing this evolutionary information, encoding the characteristics that are vital for predicting essential proteins. Each amino acid position in a PSSM is represented as a 20-dimensional vector, which reflects the conservation level of that position across different mutations. Positions that are highly conserved are assigned higher scores, signifying their stability and significance in terms of structure and function. This methodology offers a valuable tool for enhancing our understanding of essential protein functions.

To accurately predict essential proteins, it is essential to consider the local relationships within sequence data. These local features are often tied to specific functional protein families.[22] Therefore, we have implemented convolutional neural network (CNN) technology to extract relevant characteristics from protein sequences. Our CNN module is a fully convolutional network that includes an embedding layer, several convolutional layers, and an attention pooling layer. This network accepts amino acid sequences as input and converts them into embedded features through the embedding layer. The subsequent convolutional layers then encode these features, ultimately generating a comprehensive feature vector. In the CNN, local patterns within the sequence data are captured through parameter sharing and local connections. This approach leverages the fact that protein motifs consist of conserved amino acid residues found across various proteins. Consequently, CNNs can effectively extract meaningful local patterns from protein sequences, enhancing the accuracy of functional predictions.

Multiscale convolution This approach allows the model to effectively extract valuable local features from protein sequences. We employ three different scales of sliding windows during the convolution process: (5×20), (9×20), and (13×20). Each scale utilizes 128 kernels to generate diverse feature maps, aiding in the identification of specific protein functional domains and motifs. Additionally, we apply the ReLU activation function after each convolution to introduce non-linearity and enhance the model's learning capability,

$$\mathrm{ReLU}(x) = \max(0, x).$$

Attention pooling[23] Max pooling methods often result in the loss of important feature information, while attention-based strategies excel at retaining this information by emphasizing key features. The attention gathering approach introduced in this research effectively addresses this issue. Unlike max pooling, attention pooling maintains crucial information by focusing on the most significant features. A major benefit of attention pooling is its capacity to dynamically identify the most relevant features, thereby enhancing the model's learning performance. Furthermore, the adoption of an end-to-end training approach streamlines the model architecture, contributing to greater efficiency. Supporting this perspective, Meng *et al.*[24] concluded the advantages of the attention gathering strategy, showing that it outperforms other pooling methods, particularly average pooling. The latter tends to assign equal importance to all features, which is inadequate as it fails to recognize their varying significance.

The input of the deep learning-based model is the raw protein sequence. Subsequently, the model utilizes multi-scale convolutional layers with varying kernel sizes to extract advanced features. These features are subsequently passed to a fully connected layer for the final prediction. To address the imbalance in the training dataset, we apply sampling techniques to generate multiple balanced subsets for training each machine learning model. Final predictions are generated through ensemble learning models that aggregate the outputs of various models to enhance the accuracy and reliability.

2.5. Ensemble strategy

To enhance model performance and mitigate overfitting, we employ the least absolute shrinkage and selection operator (LASSO) method for feature selection. LASSO is a regularization technique predicated on sparsity, which constrains model parameters through the introduction of the L1 norm, thereby effectively eliminating redundant and inconsequential features. Following preliminary experiments, we select the top 100 features that contributed most significantly to the target variable for model training. The application of this feature selection approach markedly reduces the dimensionality of the feature space, improves the training efficiency, and diminishes the risk of overfitting.

Ensemble learning is an effective technique for constructing predictive models. The method begins by creating several base learners, which can operate either in parallel or sequentially.[25,26] Then these learners are combined, and another machine learning method is employed for the final prediction. The common combination methods for classification problems include voting, weighted averaging, and stacking.

Some machine learning is widely used to predict essential proteins. Existing calculation methods use SVM, AdaBoost, RF, XGBoost,[8] or CNN[13] models to predict essential proteins. These models have their advantages, and often learn useful protein sequence information through unique feature extractor. However, these models have limitations, including susceptibility to data noise and challenges in effectively managing complex patterns. We construct an ensemble learning framework consisting of base classifier SVM, MLP, AdaBoost, RF, ET,[27] XGBoost, GB,[28] and LightGBM for effective prediction. Integrating the prediction results from various models enhances the overall performance and robustness.

This approach leverages the strengths of each model, helping to minimize the bias and variance associated with any single model.

Ensemble learning models can be effectively constructed by incorporating base classifiers based on their cross-validation scores. After validating the performance of each classifier, their outputs are averaged and combined to improve the overall prediction accuracy. Finally, choosing the model with the highest score as the ensemble classifier achieves a favorable balance between diversity and robustness. This strategy not only enhances the model performance but also tackles the challenges posed by different data features.

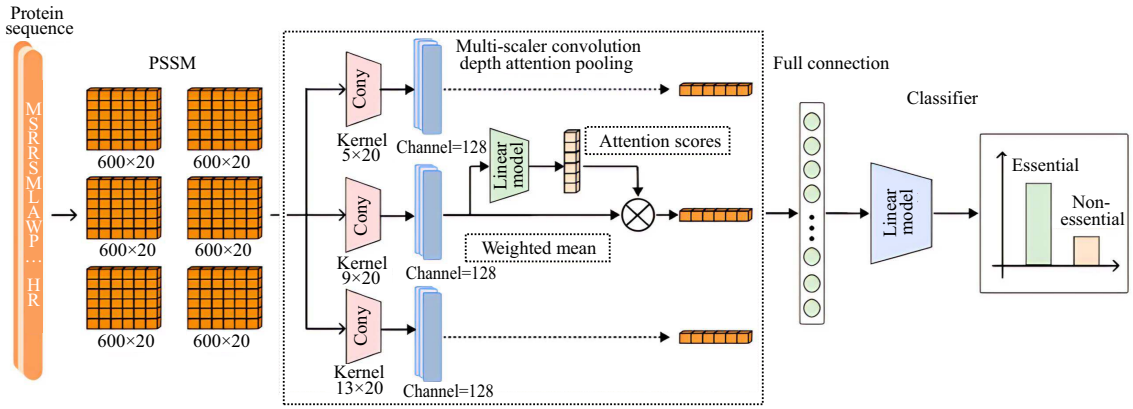


Fig. 3. A multi-scale pre trained feature extractor based on deep attention.

In our research, we implement an ensemble strategy to tackle class imbalance. By resampling multiple times, we convert the imbalanced dataset into several balanced subsets, effectively reducing classification bias. The integration framework is illustrated in Fig. 3. Each subset reflects unique distribution characteristics and produces a distinct set of classifiers upon training. When classifying new protein sequences, each base classifier independently computes a prediction score, with the final score being the average of all individual classifiers

$$\text{FinalScore}\,(p) = \frac{1}{m}\sum_{i=1}^{m} S_i,$$

where m denotes the number of base classifiers.

2.6. Evaluation method and metrics

In predicting essential proteins, we classify essential proteins as positive samples and non-essential proteins as negative samples. To assess our methods, we opt for the hold-out method rather than k-fold cross-validation. The k-fold cross-validation would result in each fold containing too few essential proteins to accurately evaluate the model performance. Instead, our hold-out approach involves randomly selecting a training set that includes all 841 essential proteins along with 6752 non-essential proteins. This method ensures a more robust evaluation of our model's effectiveness.

We use five metrics to evaluate the performances of prediction models: accuracy (ACC), precision (PRE), recall (REC), F-score, and the area under the receiver operator characteristic curve (AUC),

$$\text{ACC} = \frac{\text{TP}+\text{TN}}{\text{TP}+\text{TN}+\text{FP}+\text{FN}},$$

$$\text{PRE} = \frac{\text{TP}}{\text{TP}+\text{FP}},$$

$$\text{REC} = \frac{\text{TP}}{\text{TP}+\text{FN}},$$

$$F\text{-score} = \frac{2\times\text{precision}\times\text{recall}}{\text{precision}+\text{recall}}.$$

In this study, we use the following terms to describe classification outcomes: true positives (TP) are essential proteins correctly identified as such; false negatives (FN) are essential proteins incorrectly identified as non-essential; false positives (FP) are non-essential proteins incorrectly identified as essential; and true negatives (TN) are non-essential proteins correctly identified as such. The AUC is a commonly used metric for evaluating classifier performance and is derived from the receiver operating characteristic (ROC) curve. Both the F-score and AUC provide comprehensive assessments of a model's performance. However, given the class imbalance in our data, we emphasize the AUC in this study, as it offers a more informative measure in such contexts.

2.7. Implementation details

Our experiments were conducted using Visual Studio Code with Python 3.8 as the interpreter. The integrated learning framework was developed using PyTorch, a well-established and widely adopted open-source deep learning library.

2.8. Data collection and analysis

In this study, the proposed model consists of two inputs: PPI networks and proteins sequences information. We

also require essential proteins to label the dataset. We collected essential proteins from the Database of Essential Genes (DEG) (http://tubic.tju.edu.cn/deg/). Additionally, we compiled protein essentiality data from recent studies where researchers identified essential proteins across various cell lines using gene-trap techniques.[13,29] To ensure data reliability, we focused on the intersection of essential proteins identified in both studies. A protein is deemed essential only if it is unanimously classified as such in both works. The non-essential proteins and their sequences were collected from the UniPort database (https://www.uniprot.org/). A total of 841 essential proteins and 6752 non-essential proteins were generated in the final benchmark dataset. The PPI dataset was collected from the STRING database (https://cn.string-db.org/).

3. Results and discussion

Recently, several algorithms have emerged to identify essential proteins by leveraging both PPI networks and protein sequence data. However, challenges such as the presence of false positives in PPI data, the complexity and high dimensionality of protein sequences, and class imbalance can significantly compromise the accuracy of these predictions. To address these issues, we propose an ensemble learning framework that integrates protein sequence data with PPI networks. This approach aims to harness the strengths of deep learning for feature extraction alongside ensemble machine learning techniques to enhance the prediction of essential proteins. By combining these methods, we seek to improve both the accuracy and reliability of our predictions, ultimately contributing to more precise biological insights.

3.1. Performance of traditional machine learning models

To evaluate the effectiveness of our ensemble learning framework, we analyzed various machine learning methods to assess their performance and robustness in predicting essential proteins. We primarily used ACC, PRE, REC, F-score, and ROC curves to analyze these methods, focusing on metrics that are particularly relevant to imbalanced binary classification. While individual machine learning algorithms have their limitations, ensemble learning can enhance prediction accuracy. As illustrated in Fig. 4, the differences in ROC curves among single algorithms are minimal, with MLP demonstrating the best performance, achieving an AUC value of 0.83. Table 2 summarizes the performance of each machine learning algorithm, highlighting that LightGBM recorded the highest accuracy at 0.81, a precision of 0.78, a recall of 0.61, and an F-score of 0.62. However, there is no significant difference in performance among the various individual machine learning algorithms, indicating that the improvements seen with ensemble methods are particularly valuable.

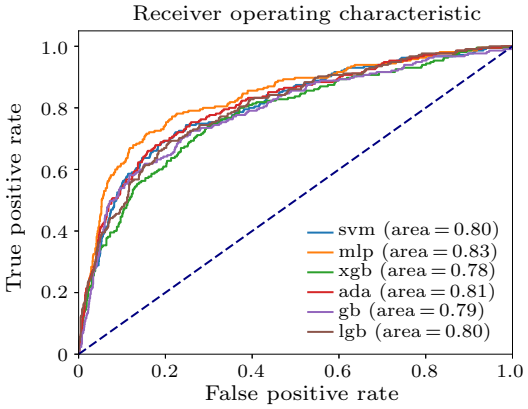


Fig. 4. ROC curves of other machine-learning algorithms.

Table 2. Comparison of prediction results of other machine learning classifiers.

Model	ACC	PRE	REC	F-score	AUC
SVM	0.7710	0.61	0.54	0.53	0.80
MLP	0.7664	0.62	0.57	0.57	0.83
XGBoost	0.7944	0.71	0.57	0.58	0.78
AdaBoost	0.7897	0.69	0.57	0.58	0.81
GB	0.7991	0.73	0.58	0.59	0.79
LightGBM	0.8131	0.78	0.61	0.62	0.80

3.2. Deep learning extracts features to improve performance

To assess the improvements in model performance achieved through our feature extraction method, we conducted a comprehensive comparison with traditional machine learning models. We analyzed several classical algorithms, including SVM, MLP, AdaBoost, RF, ET, XGBoost, GB, and LightGBM. The performance of each model was systematically evaluated using various metrics, with results detailed in Tables 2 and 3. Our findings indicate that models leveraging our feature extraction method significantly outperforms traditional models across key performance metrics, such as accuracy, precision, recall, $F1$ score, and AUC. These outcomes not only validate the effectiveness of our approach but also highlight its capacity to identify essential patterns in the data. By more effectively extracting critical information, our method enhances the overall learning capability of the models.

Table 3. Optimization of machine learning model performance.

Model	ACC	PRE	REC	F-score	AUC
SVM	0.8064	0.84	0.65	0.68	0.709
MLP	0.8011	0.81	0.68	0.65	0.739
AdaBoost	0.9159	0.91	0.83	0.87	0.926
RF	0.9008	0.92	0.78	0.83	0.866
ET	0.8318	0.85	0.62	0.65	0.896
XGBoost	0.9065	0.91	0.79	0.84	0.920
GB	0.9065	0.92	0.80	0.84	0.908
LightGBM	0.9112	0.92	0.81	0.85	0.906

As illustrated in Table 3, our deep learning feature extraction significantly increases the AUC score of AdaBoost, which rises from 0.81 to 0.926. This improvement indicates

a notable enhancement in the model's ability to distinguish between positive and negative samples. Furthermore, the $F1$ score for XGBoost is improved from 0.58 to 0.84, reflecting not only increased accuracy but also enhanced performance in handling imbalanced data. Similarly, the AUC score for XGBoost increases from 0.78 to 0.920, signifying a strengthened predictive capability. Additionally, the $F1$ score for SVM increases from 0.53 to 0.68, further illustrating its improved balance between accuracy and recall. These findings highlight the effectiveness of our feature extraction technique in enhancing model performance and capturing essential data patterns, underscoring its potential value in practical applications.

3.3. The performance of ensemble learning models

To further assess the impact of deep learning on our proposed framework, we compared the performance of models with and without deep learning using metrics such as ACC, PRE, REC, F-score, and AUC. As shown in Table 4, incorporating deep learning significantly improves the performance of our model. Specifically, deep learning increases the accuracy from 0.9019 to 0.9252, the recall from 0.84 to 0.91, the precision from 0.92 to 0.94, and the F-score from 0.88 to 0.90. These improvements highlight the effectiveness of deep learning in enhancing our model's performance.

Table 4. Comparison of model performance with deep learning.

Model	ACC	PRE	REC	F-score	AUC
Without deep learning	0.9019	0.92	0.84	0.88	0.948
Our model	0.9252	0.94	0.92	0.93	0.960

Figure 5 illustrates the ROC curves for our method compared to various machine learning techniques, with a higher AUC value indicating better performance. In Fig. 5(a), we evaluate several algorithms without utilizing deep learning for feature extraction. In this scenario, the stacking classifier excels, achieving an AUC value of 0.948, which surpasses both the voting classifier and the weighted average classifier. After incorporating deep learning for feature extraction, all algorithms show notable improvements in performance. Specifically, the AUC value for the voting classifier increases from 0.928 to 0.940, while the weighted average classifier is also improved from 0.934 to 0.946. The stacking classifier maintains its superior performance, with its AUC value rising from 0.948 to 0.960, outpacing the other ten methods. These results highlight the critical role of deep learning in enhancing the effectiveness of our framework for accurately predicting essential proteins. While each individual machine learning method may be susceptible to the challenges posed by class imbalance, the strength of the ensemble learning framework lies in its ability to amalgamate the decisions of multiple models, thereby alleviating the inherent biases associated with any single model.

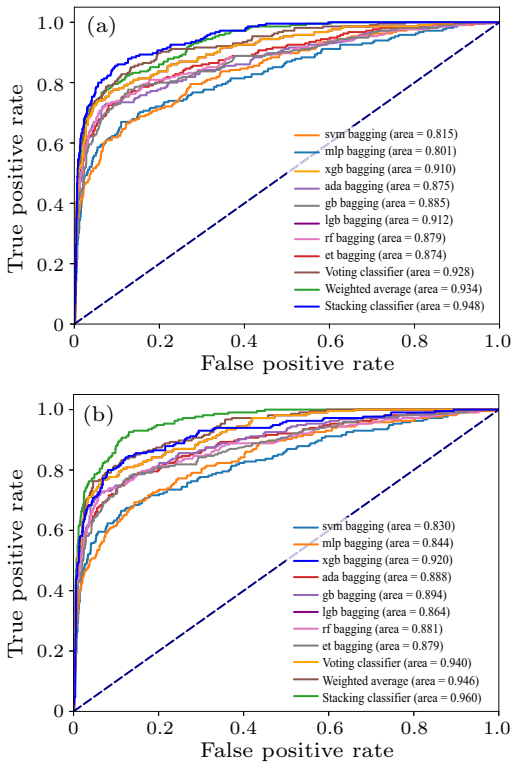


Fig. 5. Comparison of AUC and ROC curves of the model with and without deep learning.

In our study, we investigated the impact of various feature extraction schemes on model performance. As illustrated in Fig. 4, when utilizing AC and trimer features to extract characteristics from protein sequences, the performance of the machine learning model is suboptimal. However, as demonstrated in Fig. 5(a), by augmenting the feature set with information from the PPI network, we observe a substantial enhancement in model performance. Furthermore, as depicted in Fig. 5(b), the integration of features extracted through deep learning methods leads to an additional improvement in the overall model performance. Our findings underscore the critical importance of diverse and comprehensive feature extraction in bolstering model efficacy.

We selected the stacking model that demonstrated optimal performance on the validation set for further evaluation. The test dataset, sourced from DEG and UniPort database, comprises a total of 148 proteins, of which 18 are classified as critical proteins. Utilizing an imbalanced test set to evaluate model performance reveals true proficiency of a model in handling minority class samples, thereby avoiding the pitfalls associated with relying exclusively on overall accuracy. Following preprocessing, this data was input into the fully trained stacking ensemble model. We evaluated the generalization capability and performance of the model through the test set, which offers valuable insights into its adaptability and accuracy when confronted with novel data. The ensuing confusion matrix and ROC curve results in Fig. 6 illustrate that the stacking model exhibits robust performance.

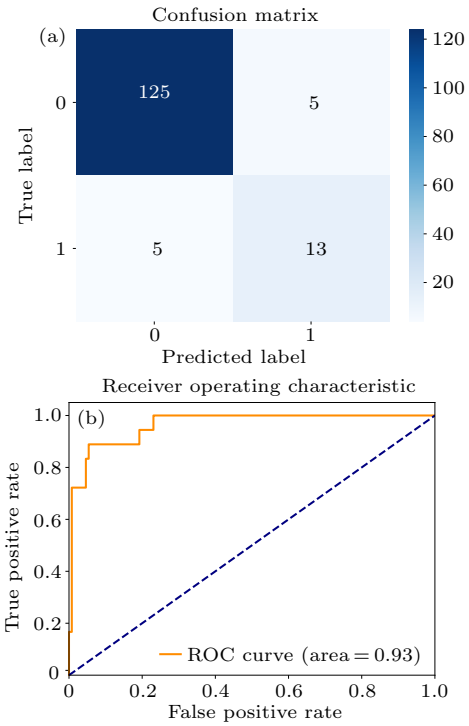


Fig. 6. The results of confusion matrix and ROC curve.

4. Conclusion

Our approach integrates protein sequence information with PPI network data, offering a practical and flexible solution for biologists. We utilize the node2vec technique to automatically extract detailed features from proteins within the PPI network. We apply deep learning techniques to extract evolutionary information from high-dimensional PSSMs, which are crucial for predicting essential proteins. By using a multi-scale CNN, we effectively capture essentiality-related features, significantly surpassing the performance of current sequence-based methods. Finally, we combine these two feature types and apply ensemble learning to complete the classification task. We evaluate our method by comparing it with eight machine learning algorithms and two ensemble models. Our ensemble learning framework addresses the class imbalance issue in essential protein prediction and enhances performance through its integrated strategy. Overall, our approach enhances the identification of essential proteins compared to previous machine learning methods. It achieves this by both improving predictive performance and creating a more robust model.

Program availability

The source code is openly available in Science Data Bank at https://doi.org/10.57760/sciencedb.16326.

Acknowledgements

This work was financially supported by the National Key R&D Program of China (Grant No. 2022YFF1202600), the National Natural Science Foundation of China (Grant No. 82301158), Science and Technology Innovation Action Plan of Shanghai Science and Technology Committee (Grant No. 22015820100), Two-hundred Talent Support (Grant No. 20152224), Translational Medicine Innovation Project of Shanghai Jiao Tong University School of Medicine (Grant No. TM201915), Clinical Research Project of Multi-Disciplinary Team, Shanghai Ninth People's Hospital, Shanghai Jiao Tong University School of Medicine (Grant No. 201914), and China Postdoctoral Science Foundation (Grant No. 2023M742332).

References

[1] Kovács I A, Luck K, Spirohn K, Wang Y, Pollis C, Schlabach S, Bian W, Kim D K, Kishore N, Hao T, Calderwood M A, Vidal M and Barabási A L 2019 *Nat. Commun.* **10** 1240
[2] Sengupta K, Saha S, Halder A K, Chatterjee P, Nasipuri M, Basu S and Plewczynski D 2022 *Frontiers in Genetics* **13** 969915
[3] Saha S, Chatterjee P, Nasipuri M and Basu S 2021 *PeerJ* **9** e12117
[4] Saha S, Halder A K, Bandyopadhyay S S, Chatterjee P, Nasipuri M and Basu S 2022 *Methods (San Diego, Calif.)* **203** 488
[5] Zhang X, Acencio M L and Lemke N 2016 *Frontiers in Physiology* **7** 75
[6] Ao C, Zhou W, Gao L, Dong B and Yu L 2020 *Genomics* **112** 4666
[7] Acencio M L and Lemke N 2009 *BMC Bioinformatics.* **10** 290
[8] Wang N, Zeng M, Li Y, Wu F X and Li M 2021 *Journal of Computational Biology* **28** 687
[9] Wu C Y, Lin B T, Shi K, Zhang Q J, Gao R, Yu Z, De Marinis Y, Liu Z P and Zhang Y 2021 *Current Bioinformatics* **16** 1161
[10] Zhong J, Wang J, Peng W, Zhang Z and Li M 2015 *Tsinghua Science and Technology* **20** 491
[11] Lu P L, Yang P S and Liao Y G 2023 *Journal of Shanghai Jiaotong University (Science)* **28** 1
[12] Schapke J, Tavares A and Recamonde-Mendoza M 2021 *IEEE/ACM Transactions on Computational Biology and Bioinformatics* **19** 1615
[13] Li Y M, Zeng M, Wu Y F, Li Y and Li M 2022 *IEEE/ACM Transactions on Computational Biology and Bioinformatics* **19** 3263
[14] Shen J, Zhang J, Luo X, Zhu W, Yu K, Chen K, Li Y and Jiang H 2007 *Proc. Natl. Acad. Sci. USA* **104** 4337
[15] Zeng M, Zhang F H, Wu F X, Li Y H, Wang J X and Li M 2019 *Bioinformatics* **36** 1114
[16] Lu P L, Zhong Y and Yang P S. 2023 *Chin. Phys. B* **33** 018903
[17] Grover A and Leskovec J 2016 *KDD: Proceedings. International Conference on Knowledge Discovery & Data Mining* **2016** 855
[18] Eraslan G, Avsec Ž, Gagneur J and Theis F J 2019 *Nat. Rev. Gene.* **20** 389
[19] Zhang F, Song H, Zeng M, Wu F X, Li Y, Pan Y and Li M 2021 *IEEE/ACM Transactions on Computational Biology and Bioinformatics* **18** 2208
[20] Zeng M, Li M, Wu F X, Li Y and Pan Y 2019 *BMC Bioinformatics* **20** 506
[21] Wang S F, Li W J, Fei Y, Cao Z C, Xu D and Guo H 2019 *IEEE Access* **7** 42384
[22] Lu P, Chen Y, Zhang T and Liao Y 2022 *Chin. Phys. B* **31** 118901
[23] Yang P, Lu P and Zhang T 2023 *Chin. Phys. B* **32** 058902
[24] Er M J, Zhang Y, Wang N and Pratama M 2016 *Information Sciences* **373** 388
[25] Lv Q, Chen G, He H, Yang Z, Zhao L, Chen H Y and Chen C Y 2023 *Chemical Science* **14** 10684
[26] Wu X H, Tao R, Sun Z Y, Zhang T Y, Li X Y, Yuan Y, Zheng S W, Cao C, Zhang Z H, Zhao X Y and Yang P 2024 *Spectrochimica acta. Part A, Molecular and Biomolecular Spectroscopy* **316** 124351
[27] Geurts P, Ernst D and Wehenkel L 2006 *Machine Learning* **63** 03
[28] Jerome H F 2001 *The Annals of Statistics* **29** 1189
[29] Blomen V A, Májek P, Jae L T, *et al.* 2015 *Science* **350** 1092