applied sciences

MDPI

Article

A Framework for Identifying Essential Proteins with Hybridizing Deep Neural Network and Ordinary Least Squares

Sai Zou [1,2,*], Yunbin Hu [3] and Wenya Yang [2]

[1] Jiuzhou Electric Group Co., Ltd., Mianyang 621000, China
[2] College of Big Data and Information Engineering, Guizhou University, Guiyang 550025, China; yang12340324@126.com
[3] College of Information Engineering, Xiamen University, Xiamen 361005, China; yunbinghu@stu.xmu.edu.cn
* Correspondence: dr-zousai@foxmail.com

Abstract: Essential proteins are vital for maintaining life activities and play a crucial role in biological processes. Identifying essential proteins is of utmost importance as it helps in understanding the minimal requirements for cell life, discovering pathogenic genes and drug targets, diagnosing diseases, and comprehending the mechanism of biological evolution. The latest research suggests that integrating protein–protein interaction (PPI) networks and relevant biological sequence features can enhance the accuracy and robustness of essential protein identification. In this paper, a deep neural network (DNN) method was used to identify a yeast essential protein, which was named IYEPDNN. The method combines gene expression profiles, PPI networks, and orthology as input features to improve the accuracy of DNN while reducing computational complexity. To enhance the robustness of the yeast dataset, the common least squares method is used to supplement absenting data. The correctness and effectiveness of the IYEPDNN method are verified using the DIP and GAVIN databases. Our experimental results demonstrate that IYEPDNN achieves an accuracy of 84%, and it outperforms state-of-the-art methods (WDC, PeC, OGN, ETBUPPI, RWAMVL, etc.) in terms of the number of essential proteins identified. The findings of this study demonstrate that the correlation between features plays a crucial role in enhancing the accuracy of essential protein prediction. Additionally, selecting the appropriate training data can effectively address the issue of imbalanced training data in essential protein identification.

Keywords: essential proteins; deep neural network; ordinary least squares; protein–protein interaction network

Citation: Zou, S.; Hu, Y.; Yang, W. A Framework for Identifying Essential Proteins with Hybridizing Deep Neural Network and Ordinary Least Squares. *Appl. Sci.* **2023**, *13*, 8613. https://doi.org/10.3390/app13158613

Academic Editor: Philippe Michaud

Received: 20 May 2023
Revised: 17 July 2023
Accepted: 19 July 2023
Published: 26 July 2023

1. Introduction

Protein is an essential element in body activity [1]. In the course of life, proteins are closely linked and interact with each other to form protein–protein interaction (PPI) networks [2]. When some of the essential proteins in the PPI network are removed, it leads to the loss of related functions and physical inactivity [3]. Therefore, the prediction of essential proteins based on PPI networks has a theoretical basis for the exploration of pathogenic genes and drug target development [4].

Early on, the identification of essential proteins mainly occurs through biological experiments [5]. Although biological experimental techniques have high accuracy, such experiments are time consuming and expensive [6]. With the development of information technology, it has become a new trend to predict essential proteins based on protein complexes or topological properties [7].

The topology of the PPI network describes the association of proteins in the form of nodes and edges [8]. Social network research shows that the greater the degree of connection between a node and other nodes, the more the node is important [9]. According to the number of degrees of nodes and the characteristics of PPI network topology, some scholars have proposed many classical algorithms, such as degree centrality (DC)

[10], betweenness centrality (BC) [11], closeness centrality (CC) [12], subgraph centrality (SC) [13], eigenvector centrality (EC) [14], information centrality (IC) [15], etc. In addition, some scholars have extended many mixed topological features to identify essential proteins based on degree centricity. This combined method has better identification accuracy than a single measurement method, such as edge clustering coefficient (ECC) combining nodes and edges [16].

The evaluation of essential proteins based on the topological characteristics of PPI networks ignores the biological significance of proteins as carriers of life activities. In addition, the essential proteins identified by these methods suffer from false negatives and false positives. To solve this problem, many algorithms have emerged to predict essential proteins by combining multiple biological information with topological features. For example, gene expression profiles are fused with network topological features [5,17,18], PPI networks, subcellular locations are fused with gene expression profiles [19], and so on. Experimental results show that these methods can improve the recognition accuracy of essential proteins. However, due to a large number of protein-related features, how to effectively use these features for essential protein recognition is another problem that needs to be solved.

Deep learning (DL) relies on the modeling ability of deep neural networks, which can not only automatically obtain multiple features from original data, but also model the non-linear relationship between features. Since it was proposed, DL has made breakthroughs in image processing and natural language understanding, and has also been widely used in the field of biological information [20–23]. The advantages of the DL framework in essential protein recognition have been confirmed by many scholars. It can provide good support for learning biological sequence data, capturing topological features from network models, and mapping network nodes into low-dimensional dense vectors [23].

Although DL can discover and characterize the complex structural features of the essential protein recognition process and improve performance, it is time consuming to train and complicated to verify the correctness of the model. At the same time, due to the absence of some data in the biological database, the robustness of the DL-based essential protein prediction method needs to be strengthened. In this paper, we propose a novel DL-based method to improve the accuracy of essential protein recognition. Our main contributions are as follows:

- We abstract the original data of the biological database through degree center, gene expression, and orthology methods and construct the DL prediction model of essential proteins, hence reducing the training time and the complexity of the training model.
- We introduce ordinary least squares to solve the metadata absence problem in biological databases, improving the robustness of the algorithm.
- Multiple simulations are designed to verify the accuracy and robustness of the IYEPDNN algorithm. When training only 80% of the DIP database, an accuracy of 87% is achieved against the remaining 20% of the DIP database, and an accuracy of 68% is achieved against GAVIN. When only 80% of the GAVIN database is trained, the remaining 20% of the GAVIN database is tested with an accuracy of 85%, and the DIP is tested with 80% accuracy. After training with 80% of randomly selected GAVIN and DIP data, the remaining 20% data is tested with an accuracy of 85.54%.

2. Materials and Data

We downloaded the yeast protein data from the DIP database [24] and GAVIN database [25] separately to build the PPI network. After removing invalid data from each dataset, 5093 proteins, 24,743 interaction relationships, and 1167 essential proteins were stored in the DIP database. There were 1855 proteins, 7669 interaction relationships, and 714 essential proteins with yeast in the GAVIN database.

To establish homology between proteins, we downloaded 100 complete genomes similar to yeast from the InParanoid database (Version 7 and 8) [26,27]. Additionally, the gene expression data of yeast was downloaded from the dataset provided by Tu BP [28].

To verify the algorithm, we downloaded 1285 essential genes of Saccharomyces from MIPS [29], SGDP [30], DEG [31], and SGD [32] databases.

3. Methods

The IYEPDNN processing flow is shown in Figure 1. As the protein–protein interaction (PPI) networks derived from the DIP and GAVIN databases only cover up to 95% of the gene expression data in the InParanoid database. It is necessary to handle absent data to enhance the algorithm's robustness. To reduce the training complexity of DNN, it is necessary to condense the input data, that is, extract the input features of DNN. Through the yeast protein association relationships in the DIP and GAVIN databases, the PPI network structure is constructed, and then the degree of each node is calculated. The gene influence of each node is calculated by the gene expression database. The homologous influence of each node is calculated from the homologous database. Then, the gene expression, PPI network, and orthology are used as input features of DNN, and the information in the essential protein library is used as output features of DNN. A DNN composed of multiple fully connected layers is used to learn 80% of randomly selected data, and the remaining 20% of data is used as a test set to construct the prediction model of IYEPDNN. The pseudocode of IYEPDNN is illustrated in Algorithm 1.

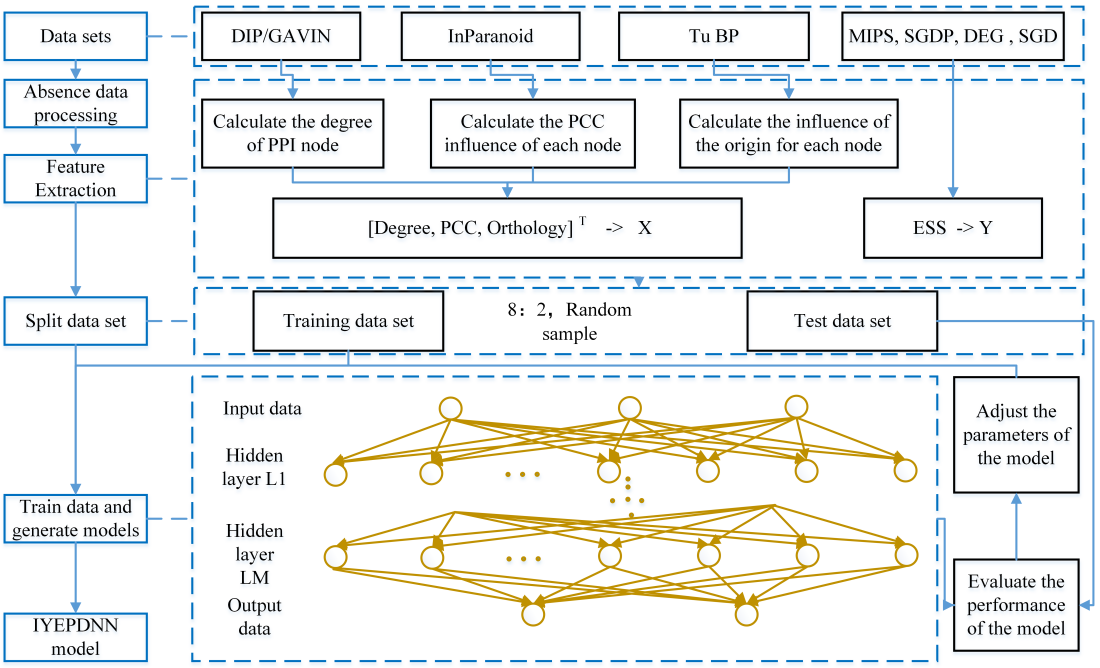

Figure 1. IYEPDNN's process.

Algorithm 1 The pseudocode of IYEPDNN

input: DIP, GAVIN, InParanoid, Tu BP, MIPS, SGDP, DEG, SGD
output: IYEPDNN model
Calculate $g_\theta(u)$ by (1);
Calculate θ by (3);
Calculate absenting gene data u by (4);
Calculate the degree of the PPI node by (6);
Calculate the PCC influence of each node by (7);
Calculate the influence of the origin for each node by (9);
$[Degree, PCC, Orthology]^T -> X$;
$ESS -> Y$;
$[Train, Test] = Splitdataset(X, Y)$;
Train data and generate models (Train);

3.1. Absent Data Processing

For the protein without corresponding gene expression data, the automatic complement is complete. Gene expression is the process by which a gene is expressed as a functional gene product; these products are often proteins. Gene expression is also widely used to identify essential proteins [33,34]. Therefore, we hope to reverse calculate gene expression information through protein information to make up for absent data. The ordinary least squares is a linear regression prediction problem [35], and its main idea is that the model is optimal when the distance between each point and the fitting model is the shortest (the residual is the least). The ordinary least squares are used to perform linear regression on the existing gene expression data. Through regression model and Gaussian perturbation, the absent gene expression data is obtained based on the existing protein information.

For a given gene, u, its gene expression at different times can be expressed by a vector, $Exp(u) = \{Exp(u,1), Exp(u,2), \cdots\}$, where $Exp(u,i)$ is the expression average level of gene u at time i. The protein degree information, d_u, and origin information, Ort, of gene u is given by

$$\begin{aligned} & g_\theta(u) \\ & = Exp(u) \times \theta \\ & = \theta_0 + Exp(u,1) \times \theta_1 + \cdots \end{aligned}, \tag{1}$$

where $\theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \vdots \end{bmatrix}$, $g_\theta(u) = \begin{bmatrix} d_u, & Ort_u \end{bmatrix}$. Let $g_\theta(u)$ be the actual value of the protein corresponding to gene u. When $\sum_{i=1} \left(\overline{g_\theta(u_i)} - g_\theta(u_i)\right)^2$ takes the minimum value, the linear fitting degree is the highest, that is, the regression model is just on the boundary of gene expression. Available:

$$\sum_{i=1} \left(\overline{g_\theta(u_i)} - g_\theta(u_i)\right)^2 = \left[\overline{g_\theta(u)} - g_\theta(u)\right]^T \cdot \left[\overline{g_\theta(u)} - g_\theta(u)\right]. \tag{2}$$

Let $\left[\overline{g_\theta(u)} - g_\theta(u)\right]^T \cdot \left[\overline{g_\theta(u)} - g_\theta(u)\right] = 0$. At the same time, take the derivative of θ from Appendix A.1.

So, we can obtain θ, as given by

$$\theta = \left(Exp(u)^T Exp(u)\right)^{-1} Exp(u)^T \overline{g_\theta(u)}. \tag{3}$$

The absent data is supplemented by the following formula.

$$Exp(u) = g_\theta^{-1}(u) + gaussian(\mu, \sigma^2), \tag{4}$$

where $gaussian(\mu, \sigma^2)$ stands for Gaussian disturbance.

3.2. Calculate the Degree of PPI Node

In a given PPI network, let V stand for node (protein) set and E stand for edge (protein–protein interaction) set so an undirected graph, $G = (V, E)$, based on the PPI network can be obtained.

Let graph $G = (V, E)$, $u \in V(G)$, $v \in V(G)$, and $e_{u,v} \in E(G)$, the degree d_u of u is

$$d_u = \sum_{v \in \Gamma(u)} Num(e_{u,v}), \tag{5}$$

where $\Gamma(u)$ indicates the set of neighbor nodes of node u. $Num()$ is a quantitative relationship. The value takes 1 if the neighbor node $e_{u,v}$ exists, and 0 otherwise.

Formula (5) is normalized, and the degree strength, Sd_u, of u is

$$Sd_u = \frac{d_u}{max(d_V)}. \tag{6}$$

3.3. Calculate Correlation of Gene Expression

The Pearson correlation coefficient (pcc) is used to measure the linear correlation between two variables, and its value is between $[-1, 1]$. We introduce the *PCC* to characterize the similarity of gene co-expression, which is widely used in the natural sciences. For genes u and v, the *PCC* between them can be calculated from Appendix A.2.

Based on Formula (A2), the average gene intensity of gene u in all nodes is given by

$$Gen_u = \frac{\frac{\sum_{v \in V} PCC_{u,v}}{n-1} - \min(Gen_V)}{\max(Gen_V) - \min(Gen_V)}. \tag{7}$$

3.4. Calculated Correlation of Origin

Semantic similarity defined by gene ontology (GO) aims to provide the functional relationship between different biological processes, molecular functions, or cellular components. We search for the shortest path that connects two terms or annotations, u and v, by using the sum of weights on the shortest path to compute the semantic similarity to measure the semantic similarity on GO. Based on the Tversky ratio model of similarity [28,29], the distance between u and v is given by

$$dis_{u,v} = \frac{dis(root, \tau)}{dis(root, \tau) + dis(\tau, u) + dis(\tau, v)}, \tag{8}$$

where τ is their lowest common ancestor, and *root* is their oldest ancestor.

Formula (8) is used to calculate the average homology intensity of node u in all nodes.

$$Ort_u = \frac{\frac{\sum_{v \in V} dis_{u,v}}{n-1} - \min(Ort_V)}{\max(Ort_V) - \min(Ort_V)}. \tag{9}$$

3.5. Training and Generation of TYEPDNN Model

Let X denote protein data after processing and Y denote the essential protein of Saccharomyces cerevisiae. Given the training set $D = \{(X, Y)\}$, $x \in X$, $y \in Y$, then y can be obtained as follows:

$$y = f\left(\sum_{x \in X} \omega x - \theta\right), \tag{10}$$

where $f()$ is the activation function, the *tanh* function is adopted in this paper, ω represents the weight, and θ represents the threshold. Training set D has three descriptive attributes for each input data, $x = [Sd, Gen, Ort]^T$. The output data is a two-dimensional real-valued vector, $y = [0/1, 0/1]$. The number of hidden layers is defined as L, and the number

of nodes of each hidden layer is h. As can be seen in Figure 1, the training model of IYEPDNN consists of three parts: the input layer, X, to the hidden layer, between the hidden layer, and the hidden layer to the output layer, Y. Let Y' be the predicted value of Y. Equations (11)–(A3) can be obtained by combining Equation (10).

The predicted value, $y'_{1,j}$, of the j-th node from the input layer to the first hidden layer is

$$\begin{aligned} y'_{1,j} &= f\left(\sum_{i=1}^{h}[\omega_{i,j,1}, \omega_{i,j,2}, \omega_{i,j,3}] \times \begin{bmatrix} Sd \\ Gen \\ Ort \end{bmatrix} - \theta_j\right), \\ &= f\left(\sum_{i=1}^{h} \omega_{i,j} x - \theta_j\right) \end{aligned} \tag{11}$$

where θ_j represents the threshold of the j-th node of the first hidden layer. The predicted value, $y'_{d,j}$, of the j-th node, $L_{d,j}$, from hidden layer c to hidden layer d is

$$y'_{d,j} = f\left(\sum_{i=1}^{h} w_{i,j} y'_{c,i} - \theta_{d,j}\right). \tag{12}$$

The predicted value, Y'_j, of the j-th node from the last hidden layer to the output layer is given by Appendix A.3.

In IYEPDNN model training, the purpose is to find the model with the least error, and the mean square error is used as the loss function, Mse.

$$Mse = \min\left(\frac{\sum_{i=1}^{size(D)} (y_i - y'_i)^2}{size(D)}\right), \tag{13}$$

where $size()$ is the length of training data of dataset D. Let $\Delta\omega$ be the updated form of the weight, ω, that is,

$$\omega \leftarrow \omega + \Delta\omega. \tag{14}$$

Based on the gradient descent method, given the learning rate, η, parameters are adjusted in the direction of the negative gradient of the target. The weight, ω, is given in Appendix A.4. Similar to Formula (A4), the number of hidden layers, L, the number of hidden layer nodes, h, and the threshold can be obtained, θ. By inserting various parameters into the training model of IYEPDNN, the judgment model of IYEPDNN can be obtained.

4. Simulation and Discussion

4.1. Relationship between the Number of Nodes in Each Layer and the Recognition Accuracy

The number of hidden layers and nodes of DNNs has a strong importance on the prediction accuracy of DNNs. When the number of nodes is too small, the DNN training model cannot learn well, which increases the training times and affects the training accuracy. When the number of nodes is too many and the training time increases each time, the DNN training model is prone to an over-fitting phenomenon. According to the characteristics of yeast protein data, data is randomly selected for testing. The relationship between classification error and the number of nodes in the hidden layer is shown in Figure 2.

Figure 2 shows that there are six hidden layers and the learning rate is 0.001. Under the condition that the number of hidden layers remains unchanged, the number of nodes of each hidden layer is constantly increased to test the accuracy of classification. As can be seen from Figure 2, in the beginning, with the increase of nodes in each layer, the

accuracy of classification becomes higher and higher. However, when the number of nodes in each layer reaches ten, the accuracy of classification does not change regularly. When the number of nodes in each layer reaches 60, the accuracy rate of classification remains at approximately 52%. When the number of nodes reaches 89, the accuracy of classification increases suddenly. It can be seen that the prediction accuracy of the yeast protein is not linear to the number of layers of DNN and nodes of each layer, which is only determined by the data characteristics of the yeast protein. Therefore, in the following experiment, the architecture with the highest classification accuracy of 73% is adopted, that is, the hidden layer consists of six layers, and the node number of each layer is 30.

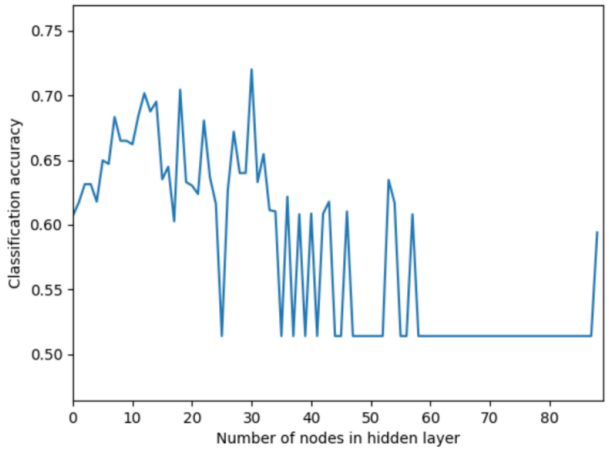

Figure 2. The influence on the accuracy of classification with the number of nodes increases.

4.2. The Relationship between Learning Rate and Recognition Accuracy

The learning rate determines the convergence of DNN, and its value is generally within [0, 1]. When the learning rate is larger, the weight modification is larger and the DNN learning speed is faster. However, if the learning rate is too high, the vibration will occur in the weight learning process. A too-small learning probability makes DNN convergence too slow and weight is difficult to stabilize. When the hidden layer number is six layers and the node number of each layer is 30, the variable learning rate method is adopted to test the influence of the learning rate on the accuracy of the IYEPDNN model. The initial learning rate is 0.001 and decreases by 10% every 1000 iterations. The accuracy impact results of the IYEPDNN model are shown in Figure 3.

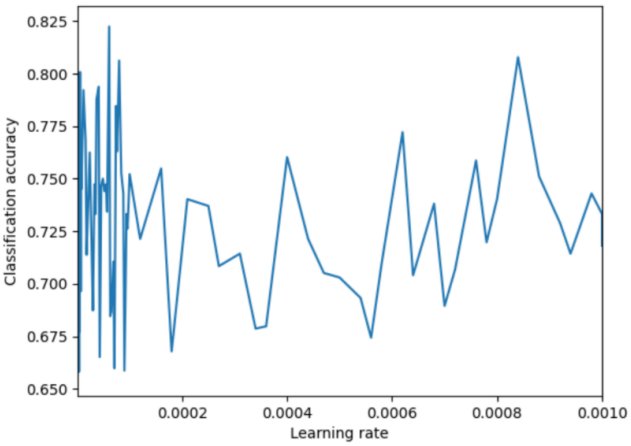

Figure 3. The influence on the accuracy of classification with the learning rate decreases.

Figure 3 shows the influence on the accuracy of classification with the learning rate decreases. As can be seen from Figure 3, when the learning rate is 6.6×10^{-5}, the recognition

rate has an obvious effect and reaches approximately 82.5%. Therefore, the maximum initial value learning rate of the IYEPDNN model is set to 0.1, which improves the DNN convergence speed. As the learning process progresses, the learning rate decreases and is maintained when the learning rate is 6.6×10^{-5}, to improve the stability and recognition rate of DNN.

4.3. Robustness Test

The data of yeast protein in the existing database are determined by biological experiments. However, different laboratory testing conditions may not be the same, and there may be errors in the same testing environment, so the robustness of the information-based means to predict essential proteins becomes a key indicator. We designed three simulations to test the robustness of the IYEPDNN model in an environment with six hidden layers, 30 nodes in each layer, and 6.6×10^{-5} fixed learning rate in the later stage.

In Figures 4, 6 and 8, the horizontal coordinate represents the number of times the model has been trained. In Figures 5, 7 and 9, the horizontal coordinate '1' represents the overall recognition success ratio, that is, the ratio of the sum of the number of essential proteins successfully identified and the number of non-essential proteins successfully identified to the total number of proteins. The horizontal coordinate '2' represents the false-negative ratio, which is the ratio of the number of misidentified non-essential proteins to the number of non-essential proteins. The horizontal coordinate '3' represents the false positive ratio, which is the ratio of the number of misidentified essential proteins to the number of essential proteins. The horizontal coordinate '4' represents the ratio of the number of essential proteins identified to the total protein number. The horizontal coordinate '5' represents the ratio of the number of non-essential proteins identified to the total protein number. In Figures 4–9, the vertical coordinates indicate the correct ratio of tests.

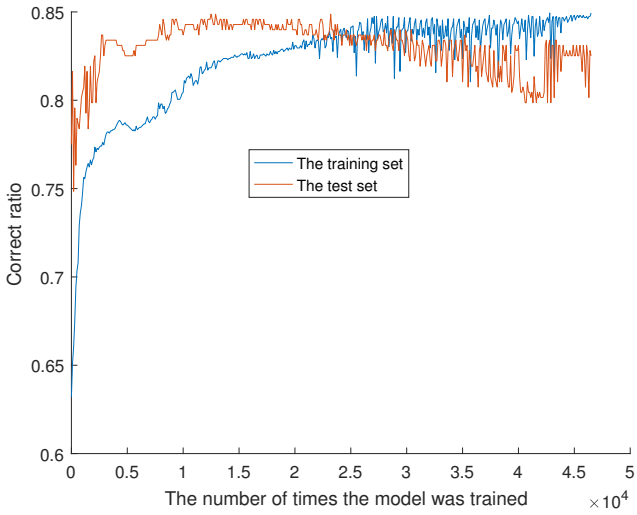

Figure 4. Only training GAVIN data generation model.

4.3.1. Train on the GAVIN Data Only

Eighty percent of essential and non-essential protein data was randomly selected from the GAVIN dataset as training data and the remaining 20% as the test data of the model. With the increase in training times, the recognition accuracy and over-fitting phenomenon are shown in Figure 4. The essential proteins of GAVIN data and DIP data are predicted, and the false positive, false negative, and correct rates are shown in Figure 5.

It can be seen from Figure 4 that the data accuracy of the test set first increases and then decreases gradually with the number of times trained. The lowest accuracy is 74.8466% and the highest accuracy is 84.8761%. The accuracy of the training set increases gradually with the number of times trained. The lowest accuracy is 63.2087% and the highest accuracy is 84.9392%. The fitting point is (x = 22,600, y = 84.0881%).

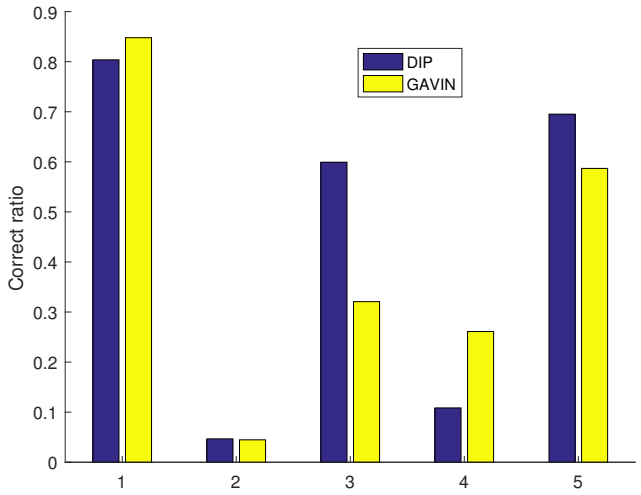

Figure 5. Only training GAVIN data results test.

As can be seen from Figure 5, the overall recognition accuracy of GAVIN is 84.7896%, and that of DIP is 80.3718%. The overall recognition accuracy of GAVIN is higher than that of DIP. The false negative rates are 4.4737% and 4.6659%, respectively. The false positive rates are 32.0728% and 59.9255% respectively. The number of essential proteins correctly identified in the GAVIN dataset is higher than that in the DIP dataset, reaching 26.1057% and 10.8517%, respectively. The number of correctly identified nonessential proteins in the GAVIN dataset is lower than that in the DIP dataset, which are 58.6839% and 69.5201% respectively.

4.3.2. Train on the DIP Data Only

Eighty percent of essential and non-essential protein data was randomly selected from the DIP dataset as training data, and the remaining 20% as the test data of the model. With the increase in training times, the recognition accuracy and over-fitting phenomenon are shown in Figure 6. The essential proteins of GAVIN data and DIP data are predicted, and the false positive, false negative, and correct rates are shown in Figure 7.

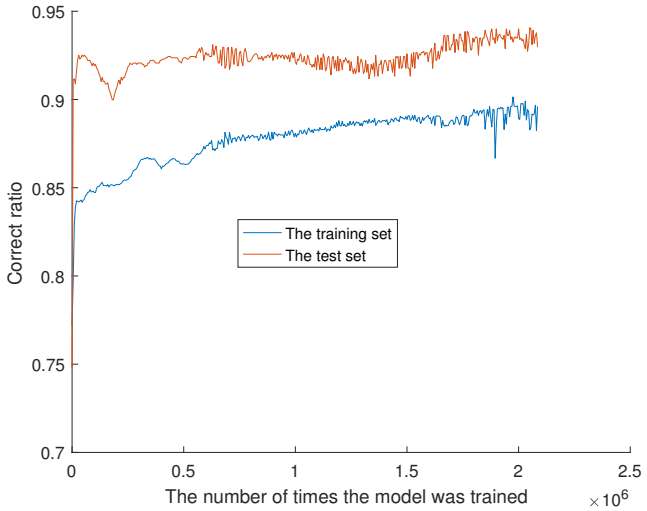

Figure 6. Only training DIP data generation model.

It can be seen from Figure 6 that the data accuracy of the test set and training set increases gradually with the number of training iterations. They only intersected at the beginning, and the model training fails to fit. The number of training sessions has reached 2,085,000, yet the accuracy has been a smooth process. It shows that it is difficult to

achieve a good fit by increasing the number of training times. Therefore, the fitting point (x = 1,346,160, y = 87.3237%) is the closest point between the test set and the training set with good accuracy. The lowest accuracy of the test set is 74.7908% and the highest accuracy of the test set is 94.0759%. The lowest accuracy of the training set is 77.2088% and the highest accuracy of the training set is 90.1467%.

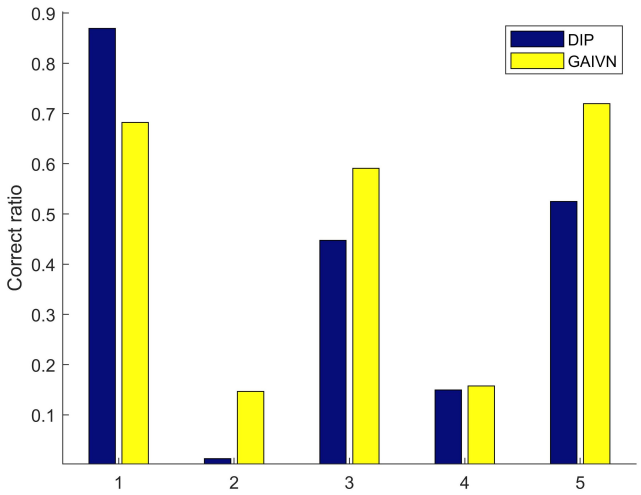

Figure 7. Only training DIP data results test.

It can be seen from Figure 7 that the overall accuracy of GAVIN is 68.2309% and that of DIP is 86.9434%. The overall accuracy of GAVIN is lower than that of DIP. The false negative and false positive aspect of the GAVIN dataset is much higher than the DIP dataset. The false negative of the GAVIN dataset is 14.6491% and the false negative of the DIP dataset is 1.2851%. The false positive of the GAVIN dataset is 59.1036% and the false negative of the DIP dataset is 44.7578%. There is no significant difference between the two datasets in terms of the number of correctly identified essential proteins. The GAVIN is 15.7497% and the DIP is 14.9589%. The number ratio of correctly identified non-essential proteins in the GAVIN dataset is much lower than that in the DIP dataset (52.4811% GAVIN and 71.9844% DIP).

4.3.3. Train on Both DIP and GAVIN

Eighty percent of the data of essential proteins and non-essential proteins was randomly selected from the DIP dataset and GAVIN dataset to synthesize the training dataset, and the other 20% was used as the test data of the model. With the increase in training times, the recognition accuracy and over-fitting phenomenon are shown in Figure 8. The essential proteins of DIP data and GAVIN data are predicted, and the false positive, false negative, and the correct rates are shown in Figure 9.

As can be seen from Figure 9, GAVIN and DIP are very close on all performance metrics. There is a 5% correlation in overall recognition accuracy, a 0.8% correlation in false negatives, a 2% difference in false positives, a 5% difference in essential protein recognition, and a 10% difference in non-essential protein recognition. The correct number of essential proteins was higher for GAVIN than for DIP. The correct number of non-essential proteins is lower for GAVIN than for DIP.

It can be seen from Figure 8 that with the increase in training time, the change of data accuracy of the test set is relatively stable. The lowest accuracy is 80.5666% and the highest accuracy is 86.3745%. The data accuracy of the test set changes greatly and the data accuracy of the training set keeps rising; the fitting point is at (x = 1,178,917, y = 84.0015).

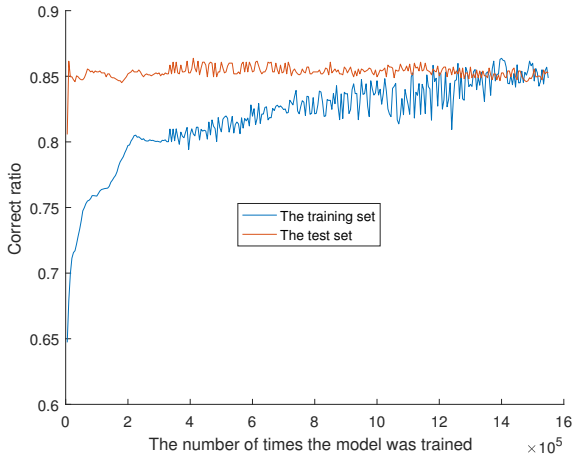

Figure 8. Training DIP+GAVIN data generation model.

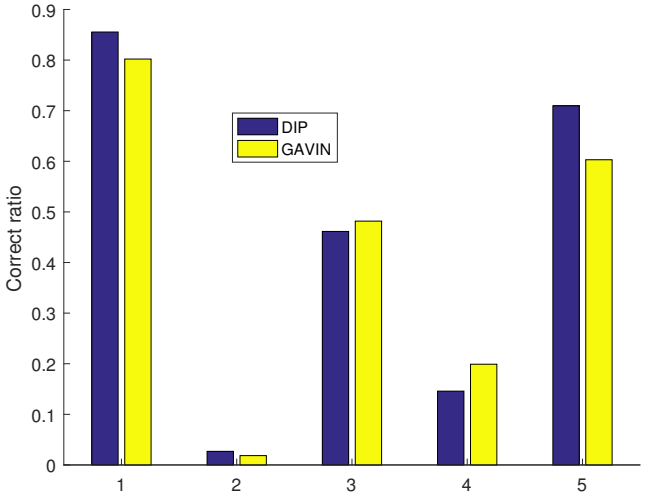

Figure 9. Training DIP+GAVIN data results test.

4.3.4. Discussion

Figure 4 runs a smaller number of times to achieve the fit. This is primarily due to the small number of GAVIN data, the characteristics of the data are relatively distinct, and the fitting point can be found quickly. Figure 6 does not have a fitting point, because DIP has a much larger number of non-essential proteins than essential proteins, with a ratio of 3:1, making it difficult to find corresponding characteristics. Figure 8 synthesizes the GAVIN and DIP data and hits the fit point after 1.1×10^6 times.

When only training the GAVIN dataset, the overall accuracy is more than 80%; when only the DIP dataset is trained, the overall accuracy is more than 70%. After training with the DIP and GAVIN datasets, the overall accuracy is over 80%. No matter the difference in training datasets, the overall accuracy is relatively high, indicating that the IYEPDNN algorithm has good generalization performance. The overall accuracy of Figure 5 is close to that of Figure 9, indicating that the GAVIN dataset has good universality and can better search for relevant features. Figures 4 and 6 also confirm this result. The false negative ratio of the three training models is low and the false positive ratio is relatively high. This is mainly due to the relatively large number of non-essential proteins in the GAVIN and DIP datasets, leading to more accurate identification of non-essential proteins by the model. Through the joint training of DIP and GAVIN data and the addition of missing gene data filling, the overall accuracy is improved and the essential protein recognition ability is enhanced.

It can be seen from Figures 5, 7 and 9 that after the IYEPDNN algorithm is trained on different datasets, there is little correlation between the overall accuracy, false negative ratio, false positive ratio, and correct recognition ratio of essential proteins and correct recognition ratio of non-essential proteins of each model. This indicates that the IYEPDNN algorithm has good robustness and can be applied to predict essential proteins in different scenarios.

4.4. Comparison of Test Results

To analyze the performance of IYEPDNN, we compare it with the network topology-related algorithms, biological characteristics-related algorithms, and artificial intelligence-related algorithms. Network topology related algorithms mainly include DC [10], SC [13], EC [11], IC [15], local average connectivity (LAC) [36], neighborhood centrality (NC) [37], and BC [11] algorithms.

Biological characteristics-related algorithms mainly include WDC (based on weighted degree centrality and gene expression data) [38], PeC (based on the integration of protein–protein interaction and gene expression data) [39], UDoNC (based on protein domains and protein–protein interaction networks) [40], LBCC (based on the combination of local density, BC [11], and DC [10]) [41], RSG (based on RNA-Seq, subcellular localization and GO annotation) [42], DEP-MSB (based on multi-source biological information) [43], OGN (based on integrating orthology information, gene expressions data, and PPI networks) [44], and TEGS (based on integrating network topology, gene expression profile, and GO annotation information, and protein subcellular localization information) [45] algorithms.

Artificial intelligence-related algorithms mainly include RWEP (based on random walk) [46], RWHN (based on randomly walking in the heterogeneous network) [47], EssRank (based on random walk) [48], EPOC (extended Pareto optimality consensus model) [49], ETB-UPPI (based on uncertain networks) [50], EPCS (community significance testing problem) [51], SigEP (local clustering coefficient) [3], RWAMVL (based on local random walk and adaptive multi-view multi-label learning) [6], and AFSO_EP (based on artificial fish swarm optimization) [52] algorithms.

We used line or histogram charts to compare the correlation algorithms. The top 1–25% or the top 100–600 candidate essential protein data of the correlation algorithms was obtained from the original paper. The top 1% in the DIP dataset contains 61 proteins, 5% contains 255 proteins, 10% contains 509 proteins, 15% contains 764 proteins, 20% contains 1019 proteins, and 25% contains 1273 proteins. The top 1% of the GAVIN dataset contains 19 proteins, 5% contains 93 proteins, 10% contains 186 proteins, 15% contains 278 proteins, 20% contains 371 proteins, and 25% contains 464 proteins.

4.4.1. Comparison of PPI Network Topology Related Algorithms

Figure 10 shows the comparison between the IYEPDNN algorithm and PPI network topology-related essential protein recognition algorithms in the DIP dataset. Figure 11 shows the comparison in the GAVIN dataset. With the decrease in the accuracy of candidate essential proteins, the number of essential proteins identified by the PPI network topology correlation algorithm increases linearly. The BC method in Figure 10 is relatively poor, while the LAC and NC methods are effective. Among the first 61 candidate essential proteins, the correct identification accuracy of NC is $32/61 > 50\%$, while the correct identification accuracy of other PPI network topological correlation algorithms is $24/61 = 39.34\%$. LAC identified 552 of the 1273 candidate essential proteins, with an accuracy of $552/1273 < 44\%$. The linear growth rate of PPI network topological correlation algorithms in DIP datasets is lower than $(552 - 29)/(1273 - 61) = 43.15\%$. In Figure 11, the EC method is relatively poor and the LAC method is relatively good. Of the first 19 candidate essential proteins, EC identifies only 6 and LAC identifies 14. EC identifies only 125 of the 464 candidate essential proteins and LAC 254. The linear growth rate of the PPI network topological correlation algorithms in the GAVIN datasets is lower than $(254 - 14)/(464 - 19) = 53.69\%$. The GAVIN dataset is superior to the DIP dataset in basic protein recognition accuracy. It can be seen from Figures 10 and 11 that the number of essential proteins recognized

by the IYEPDNN algorithm is much higher than that of the PPI network topological correlation algorithm. In the DIP dataset, the linear growth rate of the IYEPDNN algorithm is $(1122 - 42)/(1273 - 61) = 89.10\%$. The linear growth rate of the IYEPDNN algorithm in the GAVIN dataset is $(421 - 18)/(464 - 19) = 91.01\%$. At the same time, it can be seen that the number of essential proteins identified by the IYEPDNN algorithm is twice that of the PPI network topological correlation algorithm after 10% candidate essential proteins in the DIP dataset. The number of essential proteins identified by the IYEPDNN algorithm is 1.5 times higher than the PPI network topological correlation algorithm after 10% candidate essential proteins in the GAVIN dataset.

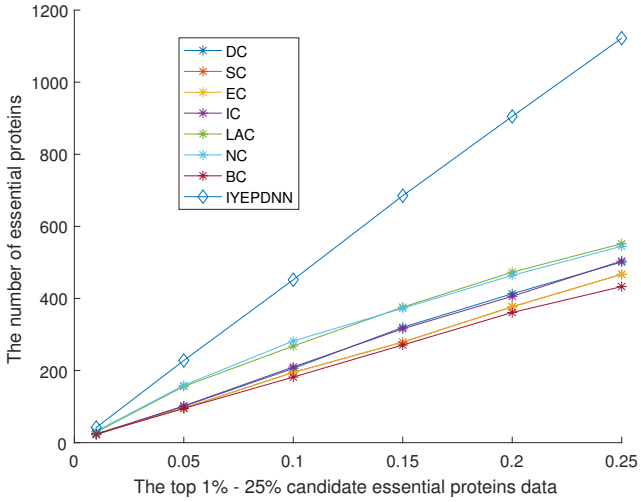

Figure 10. Comparison of PPI network topology related algorithms in DIP dataset.

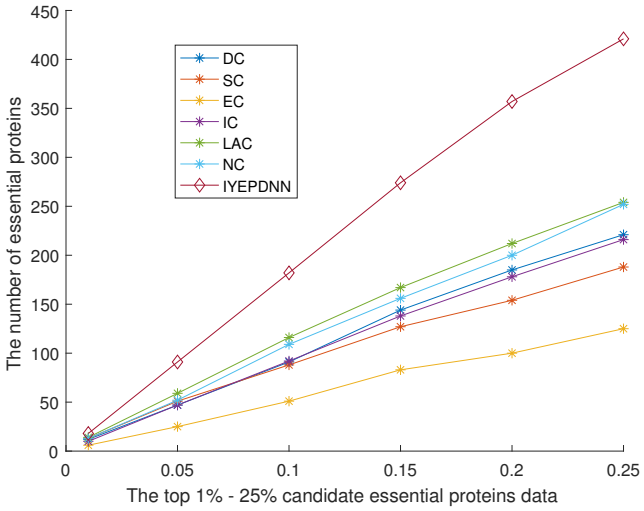

Figure 11. Comparison of PPI network topology related algorithms in GAVIN dataset.

4.4.2. Comparison with Algorithms Related to Biological Characteristics

Figure 12 shows the comparison between the IYEPDNN algorithm and the PPI+ biometric-related basic protein recognition algorithms in DIP datasets. Figure 13 is the comparison in the GAVIN dataset. Based on PPI, the identification accuracy of essential proteins is improved by introducing biometric features. Among the first 61 (1%) candidate essential proteins in Figure 12, the minimum number of recognitions is 36, which is higher than the maximum number of recognitions (32) in Figure 10. In Figure 13, among the first 19 (1%), the minimum number of recognitions is 13, which is close to the maximum number of recognitions (14) in Figure 11. However, with the increasing number of candidate essential proteins, the recognition accuracy of basic proteins improves, but the effect is not obvious

at the later stage. At 1273 (25%) candidate essential proteins in the DIP dataset, the number of LAC essential proteins identified is 552 in Figure 10, the minimum identified essential proteins is 493 in Figure 12, and the maximum identified essential proteins is 669. In the GAVIN dataset of 464 (25%) candidate essential proteins, the number of basic proteins recognized is basically above 220 in Figure 11, and the number of LAC basic proteins recognized reaches 254. Instead, it exceeds the number of basic protein identifications of PPI+ biometric in Figure 13. In the DIP dataset, the IYEPDNN algorithm is lower than DEP-MSB (45) and OGN (44) in the first 61 (1%) candidate essential proteins and far higher than the PPI+ biometric-related basic protein recognition algorithm in other links. The number of essential proteins correctly identified by the IYEPDNN algorithm is 1.67 times higher than the highest number of essential proteins identified by the PPI+ biometric correlation algorithm in 25% of DIP datasets. It reaches 1.72 times at 25% of candidate essential proteins in the GAVIN dataset.

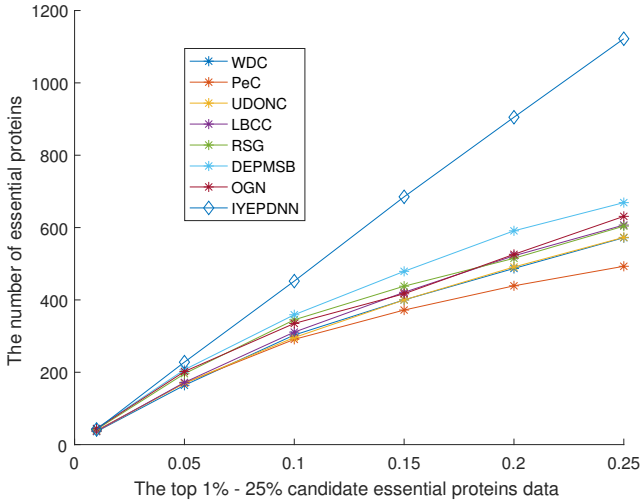

Figure 12. Comparison with algorithms related to biological characteristics in DIP dataset.

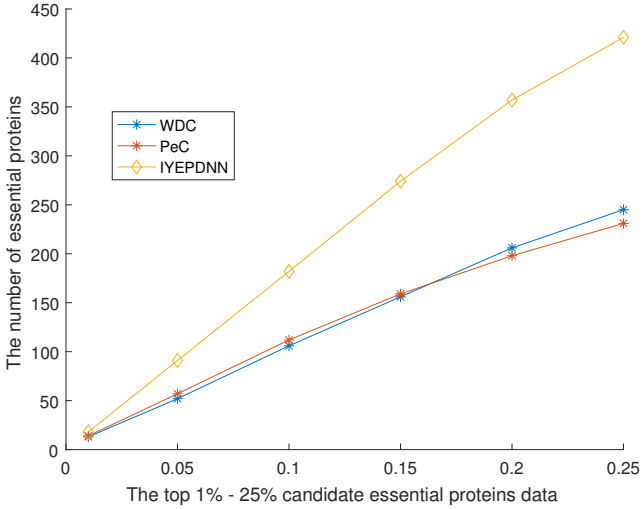

Figure 13. Comparison with algorithms related to biological characteristics in GAVIN dataset.

4.4.3. Comparison with Artificial Intelligence-Related Algorithms

Figure 14 shows the comparison between the IYEPDNN algorithm and the intelligent algorithm of related basic protein recognition in the DIP dataset. Figure 15 is a comparison in the GAVIN dataset. The comparison result of the RWHN [47] algorithm is the test results of the DIP and Gavin datasets fused into one dataset. It can be seen from the results in Figures 10 and 14 that the number of basic proteins identified by the intelligent algorithm

is superior to the PPI network topological correlation algorithm. As can be seen from the results in Figures 12 and 14, the number of basic proteins recognized by the intelligent algorithm is not significantly different from that by the PPI+ biometric correlation algorithm. Figures 11, 13 and 15 also confirm this. In Figures 14 and 15, the results identified by various intelligent algorithms go in the same direction, and each broken line is closer. It is also found that the intelligent algorithm has higher recognition accuracy than other algorithms in the top candidate essential proteins. At the later stage, there is little difference between the PPI+ biometric correlation algorithm and the PPI+ biometric correlation algorithm.

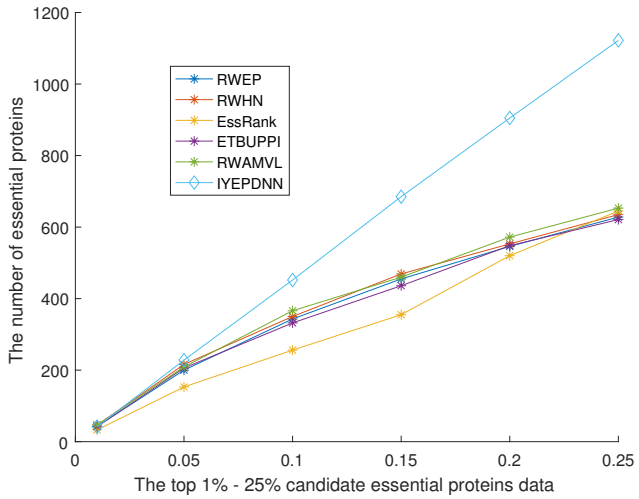

Figure 14. Comparison with artificial intelligence-related algorithms in DIP dataset.

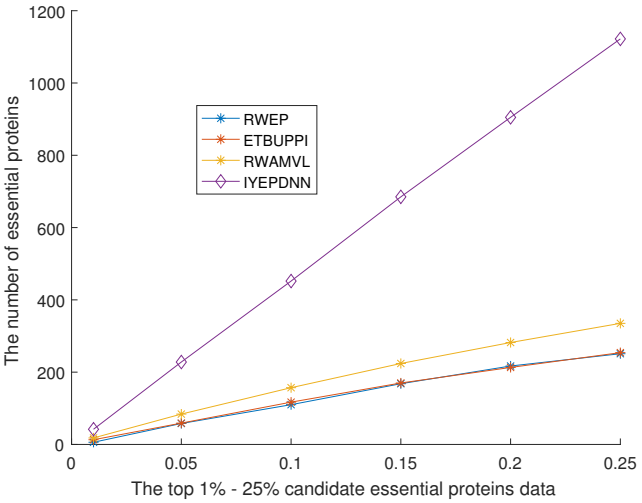

Figure 15. Comparison with artificial intelligence-related algorithms in GAVIN dataset.

Figures 16 and 17 show comparisons between the intelligent algorithm and the PPI+ biometric algorithm of the top 100–600 candidate essential proteins data. Figure 16 shows the comparison results based on the DIP dataset and Figure 17 based on the GAVIN dataset. TEGS is a PPI+ biometric-based algorithm. The results shown in Figures 16 and 17 are basically the same as those in Figures 14 and 15 .

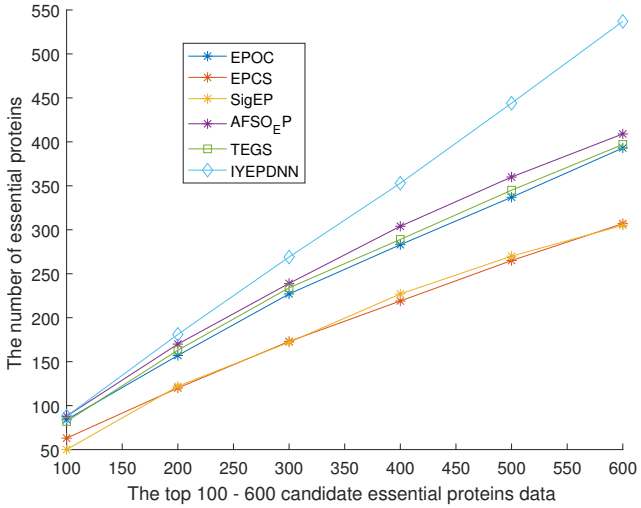

Figure 16. Comparison with artificial intelligence-related algorithms in DIP dataset (100–600).

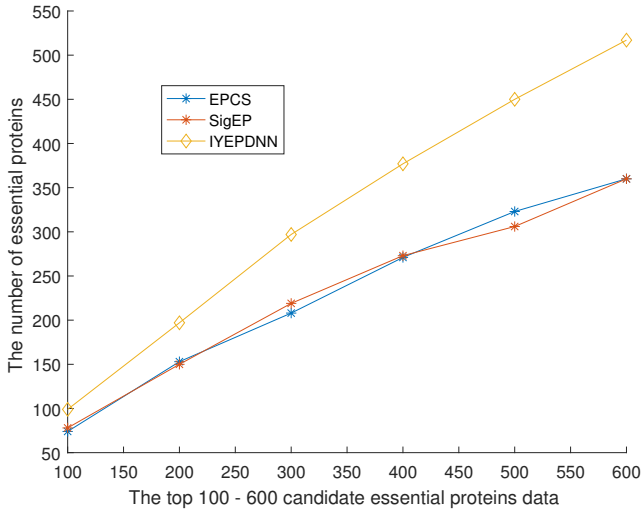

Figure 17. Comparison with artificial intelligence-related algorithms in GAVIN dataset (100–600).

4.4.4. Discussion

The results in Figures 12–15 are better than those in Figures 10 and 11, indicating that the addition of biological characteristics can reduce the false positives and false negatives caused by environmental factors and effectively improve the identification accuracy of essential proteins. The results of Figures 12 and 13 are similar to those in Figures 14 and 15. In particular, the results in Figure 16 show that it is difficult to improve the recognition accuracy by improving the algorithm alone. It is important to search for the relationship between essential and non-essential proteins and the internal correlation between the topology of the PPI network and various biological characteristics. In this paper, the ordinary least squares are used to supplement the missing data, and the deep neural network is used to find the correlation of each feature, which can effectively improve recognition accuracy. Especially in data training, 80% data from known essential proteins and 80% data from non-essential proteins are selected as training data, which can effectively avoid the problem of unbalanced training data.

5. Conclusions

In this paper, yeast protein data were downloaded from the DIP database and the GAVIN database. The genome's similar data of yeast protein were downloaded from the InParanoid database. The gene expression data of yeast protein were downloaded from

the Tu BP database. To solve the problem of incomplete gene expression data, the reverse operation of ordinary least squares is introduced to supplement absent data. Then, PPI network topology, Pearson correlation coefficient, and homologous correlation coefficient were constructed to reduce the convergence rate of DNN. Finally, DNN was used to find the optimal correlation among the node degree, Pearson correlation coefficient, and homology correlation coefficient, to improve the identification accuracy of essential proteins. Numerical studies show that proper selection of training data can effectively avoid the problem of unbalanced training data. At the same time, the correlation of each feature is the key to improving the accuracy of essential protein recognition.

Author Contributions: S.Z., conceptualization, methodology, writing—reviewing; Y.H.: software; W.Y., writing—reviewing. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by the Social Scientific Research Foundation of China (21VSZ126).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: https://github.com/GETywy/IYEPDNN.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

Appendix A.1

$$\begin{aligned}
&\frac{\partial}{\partial\theta}\left[\overline{g_\theta(u)}-g_\theta(u)\right]^T\cdot\left[\overline{g_\theta(u)}-g_\theta(u)\right]\\
&=\frac{\partial}{\partial\theta}\left[\left(\overline{g_\theta(u)}^T-g_\theta(u)^T\right)\left(\overline{g_\theta(u)}-g_\theta(u)\right)\right]\\
&=\frac{\partial}{\partial\theta}\left(\overline{g_\theta(u)}^T\overline{g_\theta(u)}-\overline{g_\theta(u)}^Tg_\theta(u)-g_\theta(u)^T\overline{g_\theta(u)}+g_\theta(u)^Tg_\theta(u)\right)\\
&=\frac{\partial}{\partial\theta}\overline{g_\theta(u)}^T\overline{g_\theta(u)}-\frac{\partial}{\partial\theta}\overline{g_\theta(u)}^Tg_\theta(u)-\frac{\partial}{\partial\theta}g_\theta(u)^T\overline{g_\theta(u)}+\frac{\partial}{\partial\theta}g_\theta(u)^Tg_\theta(u)\\
&=\frac{\partial}{\partial\theta}\overline{g_\theta(u)}^T\overline{g_\theta(u)}-\frac{\partial}{\partial\theta}\overline{g_\theta(u)}^TExp(u)\theta-\frac{\partial}{\partial\theta}Exp(u)^T\theta^T\overline{g_\theta(u)}+\frac{\partial}{\partial\theta}Exp(u)^T\theta^TExp(u)\theta\\
&=Exp(u)^TExp(u)\theta+Exp(u)\theta Exp(u)^T-Exp(u)^T\overline{g_\theta(u)}-Exp(u)^T\overline{g_\theta(u)}\\
&=0
\end{aligned}\quad . \tag{A1}$$

Appendix A.2

$$\begin{aligned}
PCC_{u,v}&=\frac{\mathrm{cov}(u,v)}{\sigma_u\sigma_v}\\
&=\frac{Exp((u-\overline{u})(v-\overline{v}))}{\sigma_u\sigma_v}\\
&=\frac{\sum\limits_{i=t-1}\left(\frac{Exp(u,i)-\overline{Exp(u)}}{\sigma_u}\right)\times\left(\frac{Exp(v,i)-\overline{Exp(v)}}{\sigma_v}\right)}{T-1}
\end{aligned}\quad , \tag{A2}$$

where $\overline{Exp(u)}$ represents the average expression of gene u at all times T, and σ_u and σ_v are the standard variance of expression for gene u at all times T. If $PCC_{u,v}$ has a positive value, then genes u and v are positively correlated; if the value of $PCC_{u,v}$ is negative, then genes u and v are negatively correlated.

Appendix A.3

$$\begin{aligned} Y'_j &= f\left(\sum_{i=1}^{2} w_{i,j} y'_{L,i} - \theta_{Y_j}\right) \\ &= f\left(\sum_{i=1}^{2} w_{i,j} \times f\left(\sum_{i=1}^{h} w_{i,j} y'_{L-1,i} - \theta_{L-1,j}\right) - \theta_{Y_j}\right) \\ &= f\left(\sum_{i=1}^{2} w_{i,j} \times f\left(\sum_{i=1}^{h} w_{i,j} \times f\left(\sum_{i=1}^{h} w_{i,j} \times \cdots \times f\left(\sum_{i=1}^{h} w_{i,j} - \theta_j\right) - \cdots\right) - \theta_{L-1,j}\right) - \theta_{Y_j}\right) \end{aligned} \quad . \tag{A3}$$

Appendix A.4

$$\begin{aligned} &\Delta\omega_{i,j} \\ &= \frac{y - y'}{x} \\ &= -\eta(y - y')x \\ &= -\eta\frac{\partial Mse}{\partial\omega_{i,j}} \\ &= -\eta\frac{\partial Mse}{\partial y'_L} \times \frac{\partial y'_L}{\partial y'_{L-1}} \times \cdots \times \frac{\partial y'_1}{\partial\omega_i} \end{aligned} \quad . \tag{A4}$$

References

1. Akp, A.; Bs, B.; Ag, C. Ortho-Sim-Loc: Essential protein prediction using Orthology and Priority-Based Similarity Approach. *Comput. Biol. Chem.* **2021**, *92*, 107503.
2. Dilucca, M.; Cimini, G.; Forcelloni, S.; Giansanti, A. Co-evolution between Codon Usage and Protein-Protein Interaction in Bacteria. *Gene* **2021**, *778*, 145475. [CrossRef] [PubMed]
3. Liu, Y.; Liang, H.; Zou, Q.; He, Z. Significance-Based Essential Protein Discovery. *IEEE/ACM Trans. Comput. Biol. Bioinform.* **2020**, *19*, 633–642. [CrossRef]
4. Zhang, W.; Xue, X.; Xie, C.; Li, Y.; Liu, J.; Chen, H.; Li, G. 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*, 349–361. [CrossRef]
5. Zhong, J.; Tang, C.; Peng, W.; Xie, M.; Yang, J. A novel essential protein identification method based on PPI networks and gene expression data. *BMC Bioinform.* **2021**, *22*, 248. [CrossRef]
6. Wang, L.; Peng, J.; Kuang, L.; Tan, Y.; Chen, Z. Identification of Essential Proteins Based on Local Random Walk and Adaptive Multi-View Multi-Label Learning. *IEEE/ACM Trans. Comput. Biol. Bioinform.* **2021**, *19*, 3507–3516. [CrossRef]
7. Noori, S.; Al-A'Araji, N.; Al-Shamery, E. SETS: A Seed-Dense-Expanding Model-Based Topological Structure for the Prediction of Overlapping Protein Complexes. *Pertanika J. Sci. Technol.* **2021**, *29*, 1323–1345. [CrossRef]
8. Yadav, A.K.; Shukla, R.; Singh, T.R. Chapter 22—Topological parameters, patterns, and motifs in biological networks. In *Bioinformatics*; Academic Press: Cambridge, MA, USA, 2022; pp. 367–380. [CrossRef]
9. Wang, K.; An, J.; Zhou, M.; Shi, Z.; Shi, X.; Kang, Q. Minority-Weighted Graph Neural Network for Imbalanced Node Classification in Social Networks of Internet of People. *IEEE Internet Things J.* **2023**, *10*, 330–340. [CrossRef]
10. Hahn, M.W.; Kern, A.D. Comparative Genomics of Centrality and Essentiality in Three Eukaryotic Protein-Interaction Networks. *Mol. Biol. Evol.* **2005**, *22*, 803–806. [CrossRef]
11. Joy, M.P.; Brock, A.; Ingber, D.E.; Huang, S. High-Betweenness Proteins in the Yeast Protein Interaction Network. *J. Biomed. Biotechnol.* **2014**, *2005*, 96. [CrossRef]
12. Wuchty, S.; Stadler, P.F. Centers of complex networks. *J. Theor. Biol.* **2003**, *223*, 45–53. [CrossRef]
13. Estrada, E.; Rodríguez-Velázquez, J. Subgraph centrality and clustering in complex hyper-networks. *Phys. A Stat. Mech. Its Appl.* **2006**, *364*, 581–594. [CrossRef]
14. Bonacich, P.; Lloyd, P. Eigenvector-like measures of centrality for asymmetric relations. *Soc. Netw.* **2001**, *23*, 191–201. [CrossRef]
15. Benini, L.; Micheli, G.D. Networks on chip: A new SoC paradigm. *Computer* **2002**, *35*, 70–78. [CrossRef]
16. Wang, H.; Min, L.; Wang, J.; Yi, P. A New Method for Identifying Essential Proteins Based on Edge Clustering Coefficient. *Lect. Notes Comput. Sci.* **2011**, *6674*, 87–98.
17. Amala, A.; Emerson, I.A. An analysis of central residues between ligand-bound and ligand-free protein structures based on network approach. *Protein Pept. Lett.* **2017**, *24*, 517–527. [CrossRef] [PubMed]

18. Du, Y.; Gao, C.; Chen, X.; Hu, Y.; Sadiq, R.; Deng, Y. A new closeness centrality measure via effective distance in complex networks. *Chaos* **2015**, *25*, 033112. [CrossRef] [PubMed]
19. Zhong, J.; Qu, Z.; Zhong, Y.; Tang, C.; Pan, Y. Continuous and Discrete Similarity Coefficient for Identifying Essential Proteins Using Gene Expression Data. *Big Data Min. Anal.* **2023**, *6*, 185–200. [CrossRef]
20. Zhang, H.; Feng, Z.; Wu, C. A Non-local Graph Neural Network for Identification of Essential Proteins. In Proceedings of the 2022 International Joint Conference on Neural Networks (IJCNN), Padua, Italy, 18–23 July 2022; pp. 1–8. [CrossRef]
21. Elbasani, E.; Kim, J.D. Graph and Convolution Recurrent Neural Networks for Protein-Compound Interaction Prediction. In *Advanced Multimedia and Ubiquitous Engineering*; Springer: Singapore, 2021.
22. Dasgupta, S.; Mondal, S.; Khan, A.; Pal, R.K.; Saha, G. Identification of Differentially Expressed Genes Using Deep Learning in Bioinformatics. In Proceedings of the International Conference on Frontiers in Computing and Systems, West Bengal, India, 13–15 January 2020; Springer: Singapore, 2021.
23. Zeng, M.; Li, M.; Fei, Z.; Wu, F.; Li, Y.; Pan, Y.; Wang, J. A deep learning framework for identifying essential proteins by integrating multiple types of biological information. *IEEE/ACM Trans. Comput. Biol. Bioinform.* **2019**, *18*, 296–305. [CrossRef]
24. Ioannis, X.; Lukasz, S.; Duan, X.J.; Patrick, H.; Sul-Min, K.; David, E. DIP, the Database of Interacting Proteins: A research tool for studying cellular networks of protein interactions. *Nucleic Acids Res.* **2002**, *30*, 303–305.
25. Gavin, A.C.; Aloy, P.; Grandi, P.; Krause, R.; Boesche, M.; Marzioch, M.; Rau, C.; Jensen, L.J.; Bastuck, S.; Dümpelfeld, B. Proteome survey reveals modularity of the yeast cell machinery. *Nature* **2006**, *440*, 631–636. [CrossRef] [PubMed]
26. Sonnhammer, E.L.; Östlund, G. InParanoid 8: Orthology analysis between 273 proteomes, mostly eukaryotic. *Nucleic Acids Res.* **2015**, *43*, D234–D239.
27. Östlund, G.; Schmitt, T.; Forslund, K.; Köstler, T.; Messina, D.N.; Roopra, S.; Frings, O.; Sonnhammer, E.L. InParanoid 7: New algorithms and tools for eukaryotic orthology analysis. *Nucleic Acids Res.* **2010**, *38*, D196–D203. [CrossRef]
28. Tu, B.P.; Kudlicki, A.; Rowicka, M.; McKnight, S.L. Logic of the Yeast Metabolic Cycle: Temporal Compartmentalization of Cellular Processes. *Science* **2005**, *310*, 1152. [CrossRef]
29. Mewes, H.W.; Amid, C.; Arnold, R.; Frishman, D.; Güldener, U.; Mannhaupt, G.; Münsterkötter, M.; Pagel, P.; Strack, N.; Stümpflen, V. MIPS: Analysis and annotation of proteins from whole genomes. *Nucleic Acids Res.* **2004**, *34* (Suppl. S1), D169–D172. [CrossRef]
30. Saccharomyces Genome Deletion Project. 2012. Available online: http://www-sequence.stanford.edu/group/yeast-deletion-project/deletion3.html (accessed on 18 July 2023).
31. Ren, Z.; Yan, L. DEG 5.0, a database of essential genes in both prokaryotes and eukaryotes. *Nucleic Acids Res.* **2009**, *37*, D455–D458.
32. Cherry, J.M. Saccharomyces genome database. *Briefings Bioinform.* **2004**.
33. Zhao, B.; Wang, J.; Li, X.; Wu, F.X. Essential protein discovery based on a combination of modularity and conservatism. *Methods* **2016**, *110*, 54–63. [CrossRef]
34. Li, M.; Zhang, J.; Liu, Q.; Wang, J.; Wu, F.X. Prediction of disease-related genes based on weighted tissue-specific networks by using DNA methylation. *BMC Med. Genom.* **2014**, *7*, S4. [CrossRef]
35. Aubry, A.; Braca, P.; Maio, A.D.; Marino, A. Enhanced Target Localization with Deployable Multiplatform Radar Nodes Based on Non-Convex Constrained Least Square Optimization. *IEEE Trans. Signal Process.* **2021**, *70*, 1282–1294. [CrossRef]
36. Li, M.; Wang, J.; Chen, X.; Wang, H.; Yi, P. A local average connectivity-based method for identifying essential proteins from the network level. *Comput. Biol. Chem.* **2011**, *35*, 143–150. [CrossRef] [PubMed]
37. Wang, J.; Min, L.; Wang, H.; Yi, P. Identification of Essential Proteins Based on Edge Clustering Coefficient. *IEEE/ACM Trans. Comput. Biol. Bioinform.* **2012**, *9*, 1070–1080. [CrossRef]
38. Tang, X.; Wang, J.; Zhong, J.; Pan, Y. Predicting Essential Proteins Based on Weighted Degree Centrality. *IEEE/ACM Trans. Comput. Biol. Bioinform.* **2014**, *11*, 407–418. [CrossRef]
39. Min, L.; Zhang, H.; Wang, J.X.; Yi, P. A new essential protein discovery method based on the integration of protein-protein interaction and gene expression data. *BMC Syst. Biol.* **2012**, *6*, 15.
40. Peng, W.; Wang, J.; Cheng, Y.; Lu, Y.; Wu, F.; Pan, Y. UDoNC: An Algorithm for Identifying Essential Proteins Based on Protein Domains and Protein-Protein Interaction Networks. *IEEE/ACM Trans. Comput. Biol. Bioinform.* **2015**, *12*, 276–288. [CrossRef] [PubMed]
41. Chao, Q.; Sun, Y.; Dong, Y. A New Method for Identifying Essential Proteins Based on Network Topology Properties and Protein Complexes. *PLoS ONE* **2016**, *11*, e0161042.
42. Lei, X.; Zhao, J.; Fujita, H.; Zhang, A. Predicting Essential Proteins Based on RNA-Seq, Subcellular Localization and GO annotation datasets. *Knowl. Based Syst.* **2018**, *151*, 136–148. [CrossRef]
43. Liu, W.; Ma, L.; Chen, L.; Chen, B.; Qiang, J. A Novel Scheme for Essential Protein Discovery Based on Multi-Source Biological Information. *J. Theor. Biol.* **2020**, *504*, 110414. [CrossRef]
44. Zhang, X.; Xiao, W.; Hu, X.; Irene, S.N. Predicting essential proteins by integrating orthology, gene expressions, and PPI networks. *PLoS ONE* **2018**, *13*, e0195410. [CrossRef]
45. 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*, 2053–2061. [CrossRef]
46. Lei, X.; Yang, X.; Fujita, H. Random walk based method to identify essential proteins by integrating network topology and biological characteristics. *Knowl. Based Syst.* **2019**, *167*, 53–67. [CrossRef]

47. Zhao, B.; Zhao, Y.; Zhang, X.; Zhang, Z.; Wang, L. An iteration method for identifying yeast essential proteins from heterogeneous network. *BMC Bioinform.* **2019**, *20*, 355. [CrossRef] [PubMed]
48. Xu, B.; Guan, J.; Wang, Y.; Wang, Z. Essential Protein Detection by Random Walk on Weighted Protein-Protein Interaction Networks. *IEEE/ACM Trans. Comput. Biol. Bioinform.* **2019**, *16*, 377–387. [CrossRef] [PubMed]
49. Li, G.; Li, M.; Wang, J.; Li, Y.; Pan, Y. United Neighborhood Closeness Centrality and Orthology for Predicting Essential Proteins. *IEEE/ACM Trans. Comput. Biol. Bioinform.* **2020**, *17*, 1451–1458. [CrossRef] [PubMed]
50. Liu, W.; Ma, L.; Chen, L.; Jeon, B. A New Scheme for Essential Protein Identification Based on Uncertain Networks. *IEEE Access* **2020**, *8*, 33977–33989. [CrossRef]
51. Liu, Y.; Chen, W.; He, Z. Essential Protein Recognition via Community Significance. *IEEE/ACM Trans. Comput. Biol. Bioinform.* **2021**, *18*, 2788–2794. [CrossRef]
52. Lei, X.; Yang, X.; Wu, F.X. Artificial Fish Swarm Optimization Based Method to Identify Essential Proteins. *IEEE/ACM Trans. Comput. Biol. Bioinform.* **2020**, *17*, 495–505. [CrossRef]