IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received June 23, 2020, accepted July 7, 2020, date of publication July 20, 2020, date of current version July 29, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3010028

Prediction of Enzyme Function Based on a Structure Relation Network

MENG LIANG, (Member, IEEE), AND JUNLAN NIE
School of Information Science and Engineering, Yanshan University, Qinhuangdao 066000, China
Corresponding author: Junlan Nie (neijlysu@126.com)

ABSTRACT Traditional biological experimental methods for enzyme function prediction have not been able to meet the increasing number of newly discovered enzymes measured by X-ray crystallography or magnetic resonance. A good computational model and protein feature representation for predicting enzymatic function can quickly annotate the functions of enzymes in chemical reactions. Existing machine learning methods usually compress protein 3D structure information into pictures convenient for convolutional neural networks (CNNs) and discard a large amount of relation information. Therefore, we proposed a method using the relation between amino acids directly to predict enzyme function. First, in addition to common structural features, we introduced a new structural feature, the relative angle of the amino acid ($C-C\alpha-C$) plane. Additionally, all protein structure features were organized into a new representation. Then, a structure relation network (SRN) to learn four features of the enzyme was established. Finally, the proposed model was evaluated on a large dataset containing 42,699 enzymes and achieved 92.08% classification accuracy, showing improvements compared with previous works.

INDEX TERMS Enzyme function prediction, relational network, amino acid object.

I. INTRODUCTION

As the volume of protein databases increases and new protein families are discovered [1], protein function prediction is beneficial not only for understanding proteins but also for proteomics. Additionally, as the number of proteins in the PDB database rapidly grows and far exceeds the ability to manually annotate protein function, an efficient computational approach is important.

Many works use the enzyme commission (EC) number as a fairly complete framework for annotation. The EC number is a numerical classification scheme based on chemical reactions proven by experimental evidence [2]. Thus, protein function prediction can be treated as a multi-label classification problem. There have been many machine learning approaches in the literature [3], [4] for automatic enzyme annotation. Before 2015, most methods used features derived from amino acid sequences and applied some classical machine learning models, e.g., KNN [5]–[7], SVM [8]–[17] and neural networks [18]. In the past few years, deep learning techniques, particularly convolutional neural networks (CNNs), have been used for protein function prediction. The main advantage of these methods is the automatic exploitation of features after data are processed into an appropriate image format. 1D convolution is applied to features related to the amino acid sequence [19], while 2D convolution is related to the position-specific scoring matrix [20] or other feature maps. Some works [21], [22] calculate the torsion angle and distance of each pair of amino acids and process them into a fixed-size feature map for CNN. Learning with structure information represented as a set of multichannel images can exceed 90% accuracy. Although protein function is determined by amino acid sequence and protein structure, the research trend suggests that protein structure, i.e., the 3D configuration of the chain of amino acids, is a more reliable predictor of protein function than amino acid sequence because it is far more conserved in nature [23].

A graph is a kind of data structure that models a set of objects (nodes) and their relationships (edges). Obviously, the relation between amino acids and protein-protein interaction networks [24] are both graphs. The traditional machine learning applications mentioned above deal with graph-structured data by mapping the graph-structured information to a simpler representation. Standard neural networks such as CNN [25]and RNN do not properly handle graphs because they pile the node features in a certain order and only treat the correlation of edges as the characteristics of nodes. One important class of models that can directly

The associate editor coordinating the review of this manuscript and approving it for publication was Ioannis Schizas.

process graph data is called the graph neural network (GNN) [26]. It has many variants, such as gated graph sequence neural networks (GGNNs) [27] and capsule graph neural networks (CapsGNNs) [28]. We consulted some existing frameworks and comprehensive reviews on the graph neural network [29]–[34]. They provide some mechanisms used in graph neural networks, such as gate mechanisms, attention mechanisms, and skip connections, and provide a thorough review of different graph neural network models as well as a systematic taxonomy of the applications. Although GNNs have achieved great success in different fields, it is remarkable that the research on graph neural networks only uses graph structures to represent the input of neural networks without using any knowledge reasoning related to graphs. Moreover, the numbers of amino acids (nodes) and their relationships (edges) in different proteins are usually different. How to deal with graphs with dynamic structures is an exciting challenging problem.

There are also some other reasoning models taking objects and relations as input, to explain their interactions. The interaction network (IN) [35] takes as input a graph that represents a system of objects and relations, instantiates pairwise interaction terms and computes their effects via a relational model, which can explain how objects in complex systems interact, supporting dynamical predictions, as well as inferences about the abstract properties of the system. Relation networks (RNs) [36] equipped on a deep learning architecture as a simple plug-and-play module can implicitly discover and learn to explain entities and their relations. It is a simple and powerful approach for learning to perform rich, structured reasoning in complex, real-world domains.

The relation between amino acids is a typical kind of graph data so that the traditional feature representation loses information and is limited by the fixed size of the model input. Additionally, the currently available data show that there is no method using relational inference models to predict protein function at present. Thus, this paper mainly addresses extracting the relations information of amino acids accepted by the RN model and builds a network model using RN as a plug-in to predict protein function.

This paper is organized as follows: we introduce the method used in the experiment in Section 2 and elaborate on the material and experimental results in Section 3. Finally, we provide some discussions and conclusions in Section 4 and Section 5, respectively.

II. EXPERIMENTAL METHOD

In this section, we describe the method used in the experiments. In the data processing stage, the residue sequence information and structure information in each PDB file of proteins were processed into a state matrix combining both structure and amino acid sequence information. Extracting the features of amino acids in a nonstatistical way was our main work in the early stage. To predict protein function using a novel feature expression method, a machine learning framework with a relational network (RN) inference module was established.

A. FEATURE EXTRACTION BASED ON AMINO ACID SEQUENCE

The state description matrix is a sequence containing structural information, and each row is an amino acid object. We used the enzyme protein collected from the PDB as the initial dataset. Each PDB contains the amino acid sequence of the protein and the specific three-dimensional position of each amino acid atom. In addition to the torsion angles of amino acids and the distance between amino acid pairs commonly used, a relative angle γ was introduced to describe the principal plane ($C - C\alpha - C$). In the obtained state description matrix with dimensionality $[m \times l]$, m is the number of amino acids in the protein, and l corresponds to the length of the object state. Therefore, a state contains four parts, including amino acid name (N), angles ϕ and ψ(A), relative distance (RD) and relative angle γ(RA).

1) EXTRACTING NAME VECTOR

In the feature extraction process, 23 types of amino acids were involved, including 20 standard amino acids, aspartic acid, glutamic acid and other amino acids with undefined residues. First, we extracted the amino acid name sequence from the PDB file to form an original text. After the word frequencies were counted and a word dictionary was generated, the words in the text were numbered based on the dictionary. Word embedding vector can not only be realized by word2ver tools such as skip-gram and continuous bag of words(CBOW), but also can be obtained as an auxiliary product when long short-term memory (LSTM) network predicts the amino acid sequence. We used LSTM augmented with stacked attention modules to train the text. The sequence fragments cut out from 4000 sequences were used as training samples. The hyperparameters of LSTM during training were as follows: the number of cell layers was 2, batchsize was 10, sequence length was 96, learning rate was 0.006, and epochs was 20. The output was optimized with a mean squared errors(MSE) loss function using the Adam optimizer. Amino acid names passed through an LSTM using a learnable lookup embedding for individual words. The dimension of the word vector was generally set to 50 to 300. More dimensions mean that more word information can be stored but require more expensive calculation costs. The dimension of the lookup table was $[23 \times 60]$. The one-hot representation of the amino acid name was a 23-dimensional vector, where 23 corresponds to the class of amino acids, while the word embedding representation was a 60-dimensional vector. The name vector obtained from training contained amino acid sequence information.

2) EXTRACTING TORSION ANGLES

The shape of the protein backbone was expressed by the two torsion angles of the polypeptide chain, which describe the rotations of the polypeptide backbone around the bonds

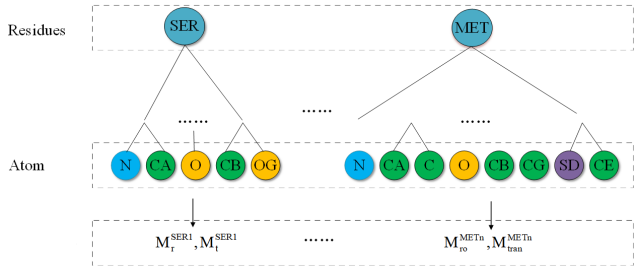

FIGURE 1. Calculate relative ($C-C\alpha-C$) plane angles and distances between residues and standard amino acids.

between $N-C\alpha$ (angle ϕ) and $C\alpha-C$ (angle ψ) $\in [-180, 180]$. In past statistical methods, the torsion angle features were often processed into the form of two-dimensional histograms (also known as Ramachandran diagrams). All amino acids in the protein were grouped according to their type and the density of the torsion angles ϕ and ψ. In contrast, we kept the original angle data and considered it an inherent property of residue, which was used to describe the method of contacting adjacent residues. Therefore, the second part of the state was 2 in length.

3) EXTRACTING RELATIVE INFORMATION

The relative rotation of $(C-C\alpha-C)$ planes of two residues on a sequence can be regarded as the accumulation of the torsion angles of a series of intermediate residues. We presented an expression of the $(C-C\alpha-C)$ plane. The $(C-C\alpha-C)$ plane of an amino acid was processed as a relation between it and a referenced standard $(C-C\alpha-C)$ plane. Due to the self-similarity of amino acids, we first sampled one instance for each type of amino acid as a standard and placed its center at the coordinate origin. Traversing the amino acid sequence of the protein, we calculated the rotation matrix M_r and translation matrix M_t, which describes how the residue can eventually coincide with the corresponding standard residue after translation and rotation. Fig. 1 shows the matrix obtained when processing amino acid sequences, where M_r^{SER1} describes the relative distance between SER1 and the standard SER, while M_t^{SER1} describes the relative rotation between their $(C-C\alpha-C)$ planes. For example, the three atoms contained in the plane of SER1 constitute point set A, which corresponds to the standard amino acid point set B. The two point sets have the same number of elements and can be placed in one-to-one correspondence. Then, they satisfy the following formula:

$$B = M_r^{SER1} \times A + M_t^{SER1} \tag{1}$$

The calculation process based on singular value decomposition usually requires the following three steps:

- Calculate the center point of the point set.

$$\mu_A = \frac{1}{3}\sum_{i=1}^{3} P_A^i, \quad \mu_B = \frac{1}{3}\sum_{i=1}^{3} P_B^i \tag{2}$$

where μ_A and μ_B are the centers of A and B, respectively, and P is the three-dimensional coordinate of the atom.

- Recenter the point set and calculate the optimal rotation matrix M_r. To calculate the rotation matrix M_r, the influence of the transition matrix M_t needs to be eliminated. New point sets A' and B' were generated, and the covariance matrix H between them was calculated by the following formula:

$$A_i' = P_A^i - \mu_A, \quad B_i' = P_B^i - \mu_B \tag{3}$$

The covariance matrix H is:

$$H = \sum_{i=1}^{3} (P_A^i - \mu_A)(P_B^i - \mu_B)^T \tag{4}$$

U, S, and V of the matrix H were obtained by the singular value decomposition (SVD) method:

$$[U, S, V] = SVD(H) \tag{5}$$

the rotation matrix was obtained by U and V:

$$M_r = VU^T \tag{6}$$

- Calculate the translation matrix M_t.

$$M_t = -M_r \times \mu_A + \mu_B \tag{7}$$

The dimension of matrix M_r is $[3 \times 3]$, while M_t is $[3 \times 1]$. Matrix M_r is a representation of the angle γ, while M_t is a representation of distance. Since standard amino acids are uniform, the matrix of amino acids can be used to calculate the relative distance and angle between them. The last two parts of the object state had a total length of 12. Therefore, the total length of an object state was 74.

B. FEATURE EXTRACTION AND CLASSIFICATION BASED ON SRN

In this section, we introduce our framework as shown in Fig. 2 for predicting enzyme functions from the relation network, using relations of amino acids as features. The input of a protein is a set of "objects", $O = \{o_1, o_2 \ldots, o_n\}, o_i \in R^{74}$. Then, the relation of each pair of amino acid objects can be described by the function g as:

$$g_\theta(o_i, o_j, r_{ij}) \tag{8}$$

The relation information r_{ij} such as the relative distance and relative angle are clearly contained in the input data. The role of g_θ is to infer the ways in which two objects are related, and its output is a "relation". The relation of all amino acid pairs constitutes a representation of enzyme function.

In Fig. 2, the framework describes three processes: (1) objects are matched in pairs, and their relation r_{ij} is calculated; (2) the function g_θ of all object pairs is calculated; and (3) the fused results are input to an MLP classifier.

First, common options for the network were used. We designed a set of more common MLP structures depending on experience and other research [36] that verified the

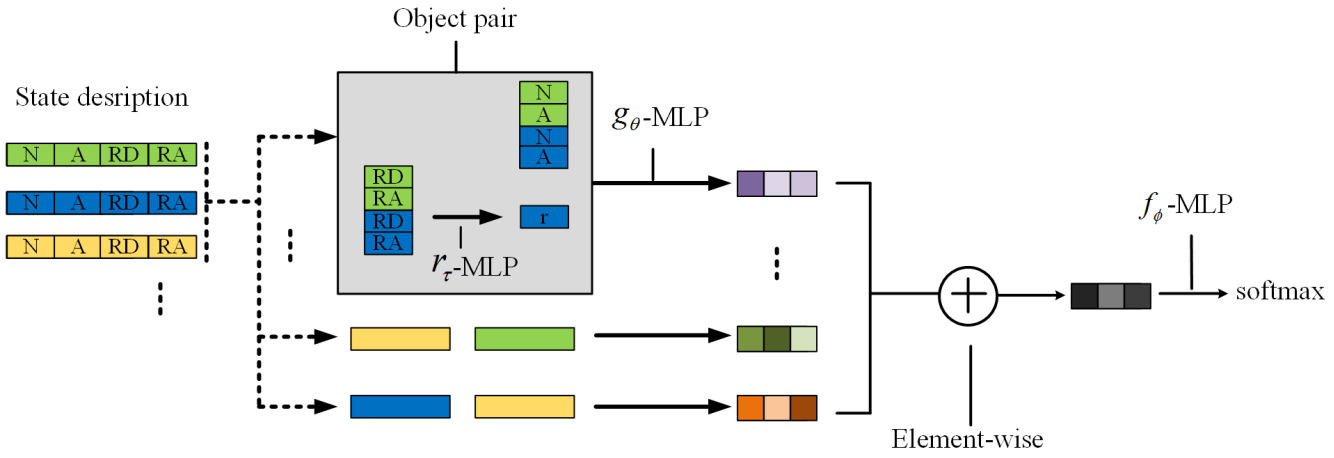

FIGURE 2. Framework of the SRN.

effectiveness of MLP as an RN network structure. Then, we used cross-validation to select the best parameters. r_τ, f_ϕ and g_θ are MLPs with parameters τ, ϕ and θ, respectively. r_τ is a three-layer MLP consisting of 24, 48, and 132 units. g_θ is a four-layer MLP consisting of 256, 256, 512, 256 units with ReLU nonlinearities. f_ϕ is a three-layer MLP consisting of 256, 256, 128, 64 units with ReLU nonlinearities. The final layer was a linear layer that produced logits for a softmax to predict the probability distribution over categories. The softmax output was optimized with a softmax loss function (i.e., the softmax operator followed by the logistic loss) using the Adam optimizer with a learning rate of 0.001. Fifty samples per batch were used for the model, which takes all channels as input. The parameters are learnable synaptic weights, making the model end-to-end differentiable. The model considers all implicit relations and can learn to infer relations; it is not related to the input order of amino acid objects. This independence guarantees that the model can handle amino acid sequences of different lengths.

In the model training process, proteins generally have hundreds of amino acids, so tens of thousands of amino acid pairs need to be processed to update the model weight once. When constructing object pairs, a function P can be introduced to reduce the number of pairs. The main function of P is to measure whether to pair two amino acid objects. The simplest strategy of P is that if the distance between two amino acids exceeds a preset threshold, do not treat them as a pair. The starting point of the strategy is that amino acids have considerably less mutual influence. Similarly, the function P can directly process the object sequence of the state matrix without the need for models such as RNN and LSTM.

We use TensorFlow as our machine learning framework and deploy the program on a device with a 2.4 GHz i7 with 16 GB memory and two Nvidia GeForce GTX 980Ti (6 GB memory card). The training sets were used to train the parameters of the network, including word embedding of amino acid names. After a week of training, this trained classifier could determine the likelihood of belonging to each class for final prediction.

III. RESULTS

The enzyme committee divides enzymes into 6 primary categories: oxidoreductases (EC1), transferases (EC2), hydrolases (EC3), lyases (EC4), isomerases (EC5), and ligases (EC6). We collected 42,699 enzymes performing a single function from the PDB database (http://www.rcsb.org/), excluding enzymes performing multiple reactions and associated with multiple enzymatic functions. The first digit of the enzyme commission (EC) code is used as a single label for protein function prediction. The number of samples per class is shown in Table 1.

TABLE 1. The numbers various single-labeled enzymes used in the experiment.

EC1	EC2	EC3	EC4	EC5	EC6
oxidoreductases	transferases	hydrolases	lyases	isomerases	ligases
16,669	1,893	1,757	3,102	7,968	11310

The dataset was split into five folds. Four folds were used as the training set and one as the test set. Then, 3 folds of the training set were used for training and one for validation. Cross-validation was used to tune the model parameters. After the model parameters were selected, the test performance was measured.

Evangelia's AD-CNN [21] is the most representative method using amino acid structure information and CNN. It has two features X_A, X_D and two architectures (Architecture 1 and Architecture 2), among which the performance of architecture 2 is better. In this experiment, we compared the proposed model with Evangelia's architecture 2. Both methods used the same test and training sets. Due to the number of test samples and the need for comparative experiments, we tested the entire dataset for comparison, as shown in Table 2. The red figures in Table 2 are the true positive rate (TPR), showing the

TABLE 2. TPR and accuracy (in percentage) in predicting main enzymatic function by two models.

Class	Sample	AD-CNN	SRN
EC1	16,669	94.32	95.26
EC2	1,893	75.00	74.88
EC3	1,757	72.55	74.54
EC4	3,102	81.98	83.93
EC5	7,968	92.01	93.17
EC6	11310	92.79	94.47
Total	42699	90.83	92.08

proportion of each enzyme that was successfully predicted. The accuracies of Evangelia's method and SRN were 90.83 % and 92.08 %, respectively.

Based on Table 2, the analytic distribution of samples in each class is shown in the form of confusion matrices in Table 3. In the longitudinal comparison of Table 3, the prediction accuracy of each enzyme in SRN was higher than that in AD-CNN. However, the prediction accuracies of EC2 and EC3 were lower than those of the other classes. As shown in Table 3, the enzyme of EC1 was more likely to be misclassified as EC6. The prediction accuracy of EC3 was not high because the model easily confused EC1 and EC6.

TABLE 3. Confusion matrices for each method.

Method	Class	1	2	3	4	5	6
AD-CNN	EC1	94.32	0.03	0.15	0.18	1.30	4.02
	EC2	9.95	75.00	0.26	1.02	4.34	9.44
	EC3	10.92	0.00	72.55	0.56	3.92	12.04
	EC4	5.26	0.16	0.32	81.98	4.94	7.34
	EC5	4.15	0.00	0.06	0.19	92.01	3.58
	EC6	4.81	0.16	0.20	0.28	1.76	92.79
SRN	EC1	95.86	0.00	0.09	0.03	0.73	3.30
	EC2	9.95	74.23	0.77	0.77	4.08	10.20
	EC3	10.08	0.00	74.51	0.00	2.52	13.00
	EC4	4.63	0.16	0.16	83.25	3.83	7.97
	EC5	3.40	0.00	0.00	0.06	93.84	2.70
	EC6	4.53	0.04	0.08	0.12	0.84	94.39

During the training process, one iteration of each batch of training data updates the model parameters and can obtain the training accuracy at this time. At the end of each training epoch, the test set was only used to evaluate test accuracy without participating in training model parameters. With the increase in the number of training epochs, the training accuracy and test accuracy gradually converged, as shown in Fig. 3. Therefore, model training of 300 epochs was sufficient. The test accuracy representing the generalization ability of the model was always lower than the training accuracy.

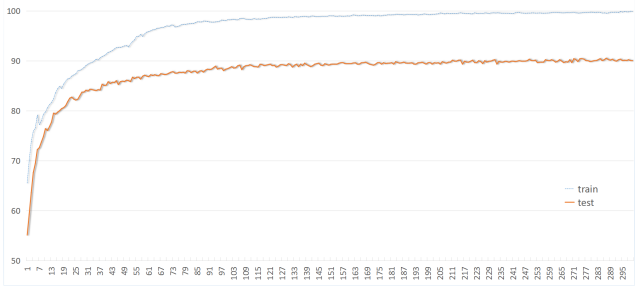

FIGURE 3. The change in two precisions for different numbers of epochs.

In Fig. 3, the red line shows the variation in iteration times and training accuracy. The blue line indicates the number of iterations and the change in training accuracy of the experiment in this paper. The corresponding parameters can only be determined after the data are trained. Then, the accuracy is obtained from the test set. In the training process, the iteration process affected the determination of parameters, which affected the test results. Therefore, the variation in prediction accuracy with the number of iterations is shown in Fig. 3.

The model performance can be measured by a loss function that assigns a penalty to classification errors. The test loss value corresponding to each epoch is shown in Fig. 4.

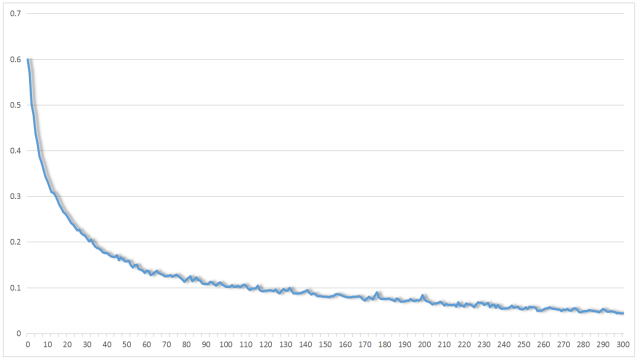

FIGURE 4. Test loss vs epochs.

Receiver operating characteristic (ROC) curves were derived from the results of cross-validation and used to further evaluate the performance of the SRN model. Fig. 5 shows the ROC curves and area under the curve (AUC) value for each type of enzyme. The ordinate represents the true positive rate (TPR), and the abscissa represents the false positive rate (FPR). An AUC value greater than 0.5 indicated that the SRN model has a certain predictive value.

IV. DISCUSSION

The method in this paper used the primary EC number as a label to predict the function of the protein and finally reached an accuracy of 92.08%, which is a considerable improvement over the previous work (90.83% in Evangelia I). More specifically, for EC1, 5, and 6, the models performed better in terms of precision, but the prediction accuracies of the three other enzymes were lower, which was due to the insufficient sample size for EC2, 3, and 4 in the dataset. If the number of training samples increased, the reliability of the predictions improved.

In addition to the display of experimental precision results, we also invited biological researchers to use our model and propose interactive experiences. They thought the machine learning model plays an important role in understanding proteins with unknown functions, and our deep learning reasoning model using amino acid relationships is an innovative method for future research using protein structural information.

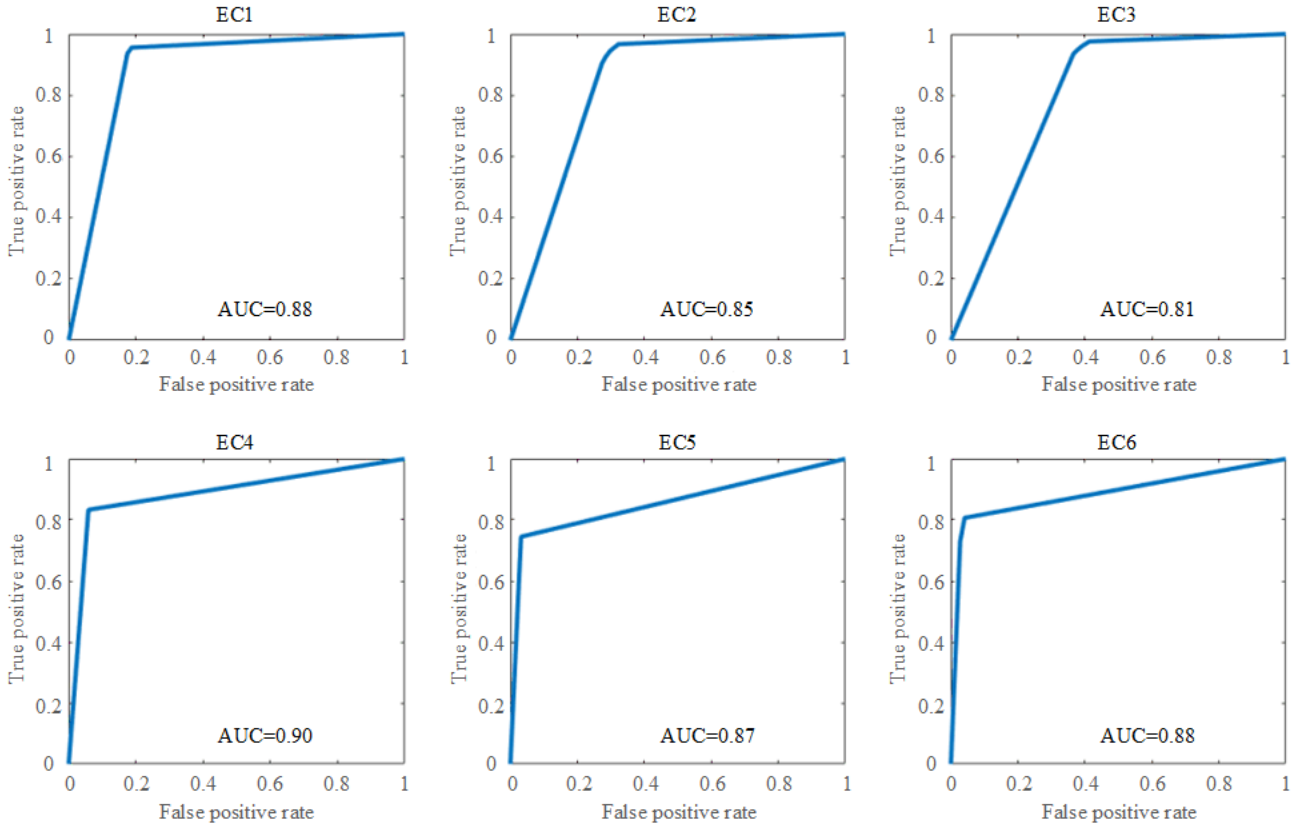

FIGURE 5. ROC curves for each enzymatic class for SRN.

Our current work shows that structural relationship information of amino acids and the relationship inference model can achieve good results in protein functional classification. Nevertheless, SRN still faces some restrictions. First, the model is currently only for single-label classification rather than multilabel classification and only predicts proteins approximately into 6 major classes without further subdivision. Second, the training of the model required considerable time during the entire experiment, so further optimization is necessary to improve performance.

V. CONCLUSIONS

In this paper, a method was presented that extracts sequence features and structural features that are introduced into a structure relation network for enzymatic function prediction.

The experimental results showed that the uncompressed use of relation information leads to more accurate enzyme class prediction. Overall, the presented approach can provide a rapid and accurate enzyme function prediction, and research-based not only on structural relations but also on inference models for function prediction provides more comprehensive ideas for the biological field. Although machine learning is a black-box model and cannot directly express the one-to-one correspondence between protein structure and function, if a protein with a new fold has similar structural characteristics to proteins in the dataset, its functional prediction will have a higher score on a certain class.

In future work, we will attempt to combine the model with more high-performance computing technology to improve the model's ability to face various complex tasks. We will also explore the relationship between the local structure and function of the protein. The prediction of protein binding sites based on the subgraph matching problem in inference models is also worth exploring.

ACKNOWLEDGMENT

The authors wish to thank Professor J. Nie from the DVVA Group, Department of Information Science and Engineering, Yanshan University, for providing the means to complete this study and R. Gao for useful discussion about the algorithm.

REFERENCES

[1] A. Godzik, "Metagenomics and the protein universe," *Current Opinion Struct. Biol.*, vol. 21, no. 3, pp. 398–403, Jun. 2011.

[2] E. Webb, *Recommendations of the Nomenclature Committee of the International Union of Biochemistry and Molecular Biology on the Nomenclature and Classification of Enzymes*. San Diego, CA, USA: Academic, 1992.

[3] M. Sharma and P. Garg, "Computational approaches for enzyme functional class prediction: A review," *Current Proteomics*, vol. 11, no. 1, pp. 17–22, Jul. 2014.

[4] S. K. Yadav and A. K. Tiwari, "Classification of enzymes using machine learning based approaches: A review," *Mach. Learn. Appl., Int. J.*, vol. 2, nos. 3–4, pp. 30–49, Dec. 2015.

[5] W.-L. Huang, H.-M. Chen, S.-F. Hwang, and S.-Y. Ho, "Accurate prediction of enzyme subfamily class using an adaptive fuzzy k-nearest neighbor method," *Biosystems*, vol. 90, no. 2, pp. 405–413, Sep. 2007.
[6] H.-B. Shen and K.-C. Chou, "EzyPred: A top–down approach for predicting enzyme functional classes and subclasses," *Biochem. Biophys. Res. Commun.*, vol. 364, no. 1, pp. 53–59, Dec. 2007.
[7] E. Nasibov and C. Kandemir-Cavas, "Efficiency analysis of KNN and minimum distance-based classifiers in enzyme family prediction," *Comput. Biol. Chem.*, vol. 33, no. 6, pp. 461–464, Dec. 2009.
[8] C. Cai, L. Han, Z. L. Ji, and Y. Z. Chen, "SVM-Prot: Web-based support vector machine software for functional classification of a protein from its primary sequence," *Nucleic Acids Res.*, vol. 7, no. 5, pp. 3692–3697, 2017.
[9] L. Y. Han, C. Z. Cai, Z. L. Ji, Z. W. Cao, J. Cui, and Y. Z. Chen, "Predicting functional family of novel enzymes irrespective of sequence similarity: A statistical learning approach," *Nucleic Acids Res.*, vol. 32, no. 21, pp. 6437–6444, 2004.
[10] P. D. Dobson and A. J. Doig, "Predicting enzyme class from protein structure without alignments," *J. Mol. Biol.*, vol. 345, no. 1, pp. 187–199, Jan. 2005.
[11] C. Chen, Y.-X. Tian, X.-Y. Zou, P.-X. Cai, and J.-Y. Mo, "Using pseudo-amino acid composition and support vector machine to predict protein structural class," *J. Theor. Biol.*, vol. 243, no. 3, pp. 444–448, Dec. 2006.
[12] L. Lu, Z. Qian, Y.-D. Cai, and Y. Li, "ECS: An automatic enzyme classifier based on functional domain composition," *Comput. Biol. Chem.*, vol. 31, no. 3, pp. 226–232, Jun. 2007.
[13] X.-B. Zhou, C. Chen, Z.-C. Li, and X.-Y. Zou, "Using Chou's amphiphilic pseudo-amino acid composition and support vector machine for prediction of enzyme subfamily classes," *J. Theor. Biol.*, vol. 248, no. 3, pp. 546–551, Oct. 2007.
[14] Y.-C. Wang, X.-B. Wang, Z.-X. Yang, and N.-Y. Deng, "Prediction of enzyme subfamily class via pseudo amino acid composition by incorporating the conjoint triad feature," *Protein Peptide Lett.*, vol. 17, no. 11, pp. 1441–1449, Nov. 2010.
[15] J. D. Qiu, J. H. Huang, S. P. Shi, and R. P. Liang, "Using the concept of Chou's pseudo amino acid composition to predict enzyme family classes: An approach with support vector machine based on discrete wavelet transform," *Protein Peptide Lett.*, vol. 17, no. 6, pp. 715–722, 2010.
[16] Y. C. Wang, Y. Wang, Z. X. Yang, and N. Y. Deng, "Support vector machine prediction of enzyme function with conjoint triad feature and hierarchical context," *BMC Syst.*, vol. 5, no. 1, pp. 1–11, 2011.
[17] A. Amidi, S. Amidi, D. Vlachakis, N. Paragios, and E. I. Zacharaki, "A machine learning methodology for enzyme functional classification combining structural and protein sequence descriptors," *Bioinf. Biomed. Eng.*, vol. 17, no. 6, pp. 728–738, 2016.
[18] V. Volpato, A. Adelfio, and G. Pollastri, "Accurate prediction of protein enzymatic class by N-to-1 neural networks," *BMC Bioinf.*, vol. 14, no. 1, pp. 11–18, 2013.
[19] Y. Li and T. Shibuya, "Malphite: A convolutional neural network and ensemble learning based protein secondary structure predictor," in *Proc. IEEE Int. Conf. Bioinf. Biomed. (BIBM)*, Washington, DC, USA, Nov. 2015, pp. 1260–1266.
[20] M. Spencer, J. Eickholt, and J. Cheng, "A deep learning network approach to *ab initio* protein secondary structure prediction," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 12, no. 1, pp. 103–112, Jan. 2015.
[21] E. I. Zacharaki, "Prediction of protein function using a deep convolutional neural network ensemble," *PeerJ Comput. Sci.*, vol. 3, p. e124, Jul. 2017.
[22] R. Gao, M. Wang, J. Zhou, Y. Fu, M. Liang, D. Guo, and J. Nie, "Prediction of enzyme function based on three parallel deep CNN and amino acid mutation," *Int. J. Mol. Sci.*, vol. 20, pp. 2845–2856, Jan. 2019.
[23] K. Illergård, D. H. Ardell, and A. Elofsson, "Structure is three to ten times more conserved than sequence—A study of structural response in protein cores," *Proteins, Struct., Function, Bioinf.*, vol. 77, no. 3, pp. 499–508, Nov. 2009.
[24] A. Fout, J. Byrd, B. Shariat, and A. Ben-Hur, "Protein interface prediction using graph convolutional networks," in *Proc. NIPS*, 2017, pp. 6530–6539.
[25] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.
[26] F. Scarselli, M. Gori, A. Chung Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE Trans. Neural Netw.*, vol. 20, no. 1, pp. 61–80, Jan. 2009.
[27] Y. Li, D. Tarlow, M. Brockschmidt, and R. Zemel, "Gated graph sequence neural networks," in *Proc. ICLR*, 2016.
[28] X. Zhang and L. Chen, "Capsule graph neural network," in *Proc. ICLR*, 2019.
[29] F. Scarselli, M. Gori, A. Chung Tsoi, M. Hagenbuchner, and G. Monfardini, "Computational capabilities of graph neural networks," *IEEE Trans. Neural Netw.*, vol. 20, no. 1, pp. 81–102, Jan. 2009.
[30] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst, "Geometric deep learning: Going beyond Euclidean data," *IEEE Signal Process. Mag.*, vol. 34, no. 4, pp. 18–42, Jul. 2017.
[31] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, "Neural message passing for quantum chemistry," in *Proc. Proc. 34th Int. Conf. Mach. Learn.*, vol. 70. JMLR.org, 2017, pp. 1263–1272.
[32] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proc. CVPR*, Jun. 2018, pp. 7794–7803.
[33] P. W. Battaglia, J. B. Hamrick, V. Bapst, A. Sanchez-Gonzalez, and V. Zambaldi, "Relational inductive biases, deep learning, and graph networks," Jun. 2018, *arXiv:1806.01261*. [Online]. Available: https://arxiv.org/abs/1806.01261
[34] J. Zhou, G. Cui, Z. Zhang, C. Yang, Z. Liu, and L. Wang, "Graph neural networks: A review of methods and applications," Dec. 2018, *arXiv:1812.08434*. [Online]. Available: https://arxiv.org/abs/1812.08434
[35] P. W. Battaglia, R. Pascanu, M. Lai, D. Rezende, and K. Kavukcuoglu, "Interaction networks for learning about objects, relations and physics," in *Proc. NIPS*, 2016, pp. 4502–4510.
[36] A. Santoro, D. Raposo, D. G. T. Barrett, M. Malinowski, R. Pascanu, and P. Battaglia, "A simple neural network module for relational reasoning," in *Proc. NIPS*, 2017, pp. 4967–4976.

MENG LIANG (Member, IEEE) received the B.S. degree in computer science from Yanshan University, Qinhuangdao, China, in 2012, where he is currently pursuing the successive master's-Ph.D. degree in computer science. From 2014 to 2020, he was a Research Assistant with the Data Visualization and Visual Analysis Laboratory. During this period, he wrote six articles. His research interests include visualization, visual analysis of spatiotemporal data, and machine learning. He won the First Prize of Chinavis Visualization Competition, in 2017, and the Second Prize of Chinavis, in 2019.

JUNLAN NIE was born in Wuhan, China, in 1962. She received the B.S. degree in electronics from Hebei University, and the Ph.D. degree from the Hebei University of Technology. She had been teaching computer, before she became a Professor, in 2013. She is the author of more than 70 articles, and 12 National Funds. Her research interests include data visualization, visual analysis, and virtual reality.