Poster Volume

The 2024 Second International Conference on Applied Intelligence

November 22-25, 2024

Zhengzhou, Henan, China

Preface

The second International Conference on Applied Intelligence (ICAI 2024) was held during November 22-25, 2024, Zhengzhou, Henan, China. The conference is started to provide an annual forum dedicated to the emerging and challenging topics in artificial intelligence, machine learning, pattern recognition, bioinformatics, and computational biology. It aims to bring together researchers and practitioners from both academia and industry to share ideas, problems, and solutions related to the multifaceted aspects of Applied Intelligence.

This year, the conference concentrated mainly on the theories and methodologies as well as the emerging applications of Applied Intelligence. Its aim was to unify the picture of contemporary Applied Intelligence techniques as an integral concept that highlights the trends in advanced computational intelligence and bridges theoretical research with applications. Therefore, the theme for this conference was "Advanced Applied Intelligence Technology and Applications". Papers that focused on this theme were solicited, addressing theories, methodologies, and applications in science and technology.

ICAI 2024 received 228 submissions from 8 countries and regions. All papers went through a rigorous peer-review procedure and each paper received at least three review reports. Based on the review reports, the Program Committee finally selected 11 Poster papers from the accepted papers, included in a volume. These volume of Poster papers will be arranged on the open access website http://poster-openaccess.com/.

The organizers of ICAI 2024, including the Society of International Computing, China, made an enormous effort to ensure the success of the conference. We hereby would like to thank the members of the Program Committee and the referees for their collective effort in reviewing and soliciting the papers. In particular, we would like to thank all the authors for contributing their papers. Without the high-quality submissions from the authors, the success of the conference would not have been possible. Finally, we are especially grateful to the International Neural Network Society, and the National Science Foundation of China for their sponsorship.

De-Shuang Huang
ICAI 2024 General Chair

Contents

Neural Networks

Protein Structure and Function Prediction

Identification of Membrane Protein Types via Deep Residual Hypergraph Neural Network

Jiyun Shen, Zhiqiang Hui, Long Cheng

Suzhou University of Science and Technology

Abstract. Conventional computational methods for identifying the species of membrane proteins tend to ignore two issues: high-order correlation among membrane proteins and the scenarios of multi-modal representations of membrane proteins, which leads to information loss. To tackle those two issues, we use a deep residual hypergraph neural network (DRHGNN) to learn the representations of membrane proteins further and to achieve accurate identification of membrane proteins' types eventually.

1 Methods

In order to extract features from membrane proteins' PSSM, we employ Average Blocks (AvBlock), Discrete Cosine Transform (DCT), Discrete Wavelet Transform (DWT), Histogram of Oriented Gradient (HOG), and Pseudo-PSSM (PsePSSM). Each type of PSSM-based feature is used to generate a hypergraph G which can be represented by an incidence matrix H. Then, five types of features and corresponding H are concatenated, respectively, and both are fed into a deep residual hypergraph neural network (DRHGNN) to identify the types of membrane proteins. Figure.1. depicts the schematic diagram.

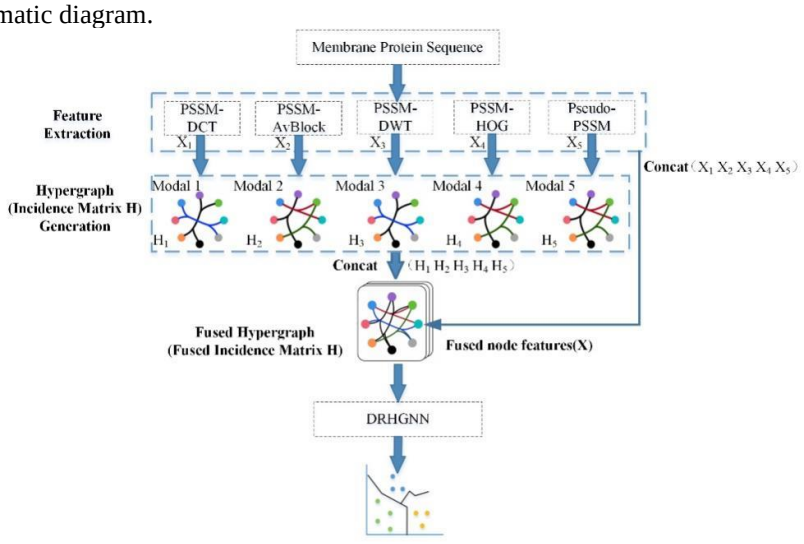


Figure. 1. The schematic diagram of our proposed method.

Figure.2. illustrates the detail of the deep residual hypergraph neural network (DRHGNN). Those multi-types of node features and corresponding incidence matrix H modelling complex high-order correlation are concatenated, respectively, which overcomes the scenarios of multi-modal representations of membrane proteins. Then, concatenated features and incidence matrix are fed into deep residual hypergraph neural network to get nodes output labels and eventually achieve classification task. We build a residual enhanced hypergraph convolution layer. Then we naively stack multiple residual hypergraph convolution blocks to tackle the problem of over-smoothing in HGNN and enjoy an accuracy increase. Additional Linear transforms are incorporated into the model's first and last layer, and the residual hypergraph convolutions are utilized for information propagation. The deep embeddings are finally used for classification tasks.

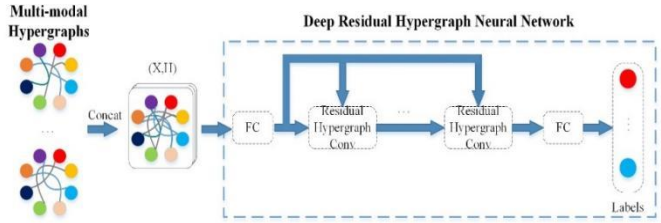


Figure .2. The DRHGNN framework. FC represents a fully connected layer.

2 Datasets

We judge the performance of DRHGNN on the classification of membrane proteins based on four datasets. Table 1. outlines the details of the datasets.

Table 1. The scale of training and testing samples in four different membrane proteins' datasets.

Specific types	Dataset 1		Dataset 2		Dataset 3		Dataset 4	
	Train	Test	Train	Test	Train	Test	Train	Test
Single-span type 1	610	444	388	223	561	245	435	478
Single-span type 2	312	78	218	39	316	7	152	180
Single-span type 3	24	6	19	6	32	9	-	-
Single-span type 4	44	12	35	10	65	17	-	-
Multi-span type 5	1,316	3,265	936	1,673	1,119	2,478	1,311	1,867
Lipid-anchor type 6	151	38	98	26	142	36	51	14
GPI-anchor type 7	182	46	122	24	164	41	110	86
Peripheral type 8	610	444	472	305	674	699	-	-
Overall	3,249	4,333	2,288	2,306	3,073	3,604	2,059	2,625

* - represents not available.

3 Results

As Figure. 3. shows, HGNN using identify mapping can mitigate the problem of over-smoothing a little, and HGNN using initial residual can reduce the over-smoothing problem greatly. Meanwhile, adopting initial residual and identity mapping together can significantly improve performance while effectively reducing the over-smoothing problem. Furthermore, we find that the experimental results of HGNN adopting initial residual and identity mapping together and HGNN using initial residual are very close. However, HGNN adopting both outperforms in terms of accuracy and the macro average of the F1-score and reaches the best result faster than just adopting the initial residual.

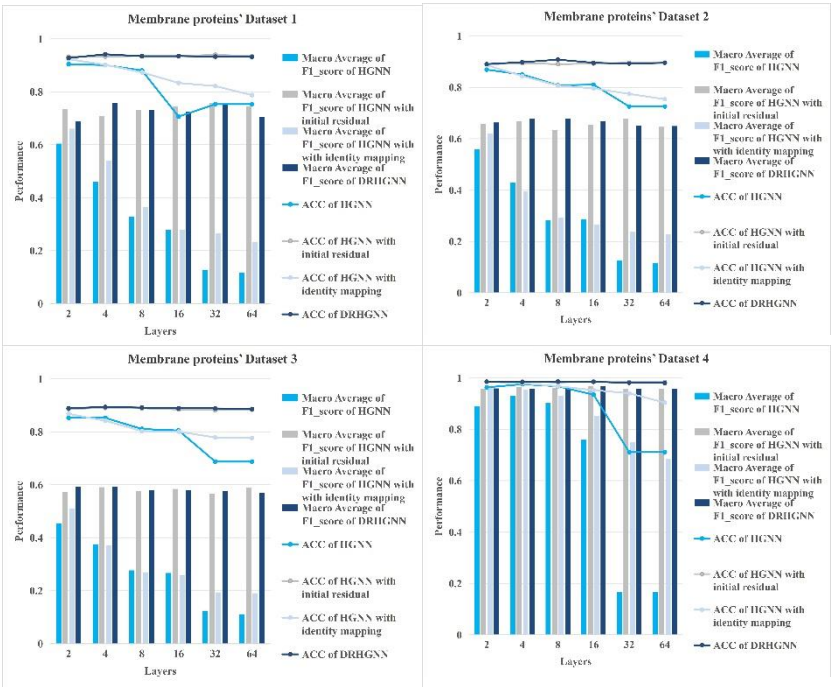


Figure. 3. The performance comparison of DRHGNN, HGNN, HGNN with initial residual, HGNN with identity mapping with different layers on membrane protein classification task.

4 Conclusions

DRHGNN resolves the following issues: the high-order correlation among membrane proteins and the scenarios of multi-modal representations of membrane proteins.

We carry out extensive experiments whose results demonstrate the better performance of DRHGNN on membrane protein classification task. Experiments also show that DRHGNN can handle the over-smoothing issue as the number of model layers increases compared with HGNN.

SGC-based Anomaly Detection for Multivariate Time Series

Kewei Hu, Qiang Tian, Biao Wang, Jiakun Wu, and He Li*

Xidian University, Xi'an 710000, China
heli@xidian.edu.cn

Abstract. In industrial facilities or IT systems, there are lots of multivariate time series generated from various metrics. Anomaly detection in multivariate time series is of great importance in applications such as fault diagnosis and root cause discovery. Recently, some unsupervised methods have made great progress in this task, especially the reconstruction architecture of autoencoders (AEs), learning normal distribution, and producing a significant error for anomalies. Although AEs can reconstruct subtle abnormal patterns well with the powerful generalization ability, it also leads to a high false negative. Moreover, these AE-based models ignore the dependence among variables at different time scales. In this paper, we propose an enhanced anomaly detection framework that builds upon the Multiscale Wavelet Graph Autoencoder (MEGA) by substituting the Graph Convolutional Network (GCN) with Simplified Graph Convolution (SGC) to augment the model's performance. The core idea is to leverage the spectral methods of SGC to process the multivariate time series data obtained by integrating Discrete Wavelet Transform (DWT) into the AE. Experiments have been conducted on three public multivariate time-series anomaly detection datasets. The results indicate that the improved model utilizing SGC performs comparably to MEGA, yet in certain scenarios, it may provide slightly better outcomes.

Keywords: Anomaly detection · discrete wavelet transform (DWT) · simple graph convolution(SGC) · multivariate time series.

1 Introduction

In time-series anomaly detection, identifying outliers from normal data distributions has gained increasing attention from academia and industry. Multivariate time series, which record multiple system indicators, are crucial for applications [24] like system monitoring and troubleshooting. In industrial environments [23], with numerous operational indicators generated continuously, manual monitoring is impractical, making automatic anomaly detection essential.

Traditional machine learning methods like KNN [2] and One-class SVM [13] have been proposed but struggle with high-dimensional and complex data. Recently, deep learning approaches, particularly deep autoencoders (AEs) [16], have

been explored for unsupervised anomaly detection, which model temporal dependence and intervariable dependence to reconstruct time series for anomaly detection [18]. Although these reconstruction-based deep models have achieved good performance in detecting a distinct anomaly, that is, obviously deviated from normal patterns, they fail to detect subtle contextual anomalies that behave normally compared to their neighbors.

Moreover, system-level anomalies often involve inter-variable dependencies [26], which are not adequately captured by directly modeling the original multivariate time series. These time series consist of oscillations at multiple scales, leading to varying inter-variable dependencies.

This paper introduces an enhanced multivariate time-series anomaly detection framework using SGC [22]. SGC reduces training time, computational resources, and model complexity while better capturing long-term dependencies compared to traditional GCN [7]. Our contributions are:

1.Dynamic Graph Structures with SGC:After decomposing the time series into multifrequency components, we construct dynamic graph structures at each scale, using the frequency components as node features.We employ SGC to capture the complex inter-variable dependencies at different scales. SGC's spectral methods provide a more efficient and effective way to process these dependencies compared to GCN.

2.Extensive experiments on three public multivariate time-series anomaly detection datasets demonstrate that our model maintains good performance while significantly reducing computational complexity and training time.

2 Related Work

2.1 Multivariate Time-Series Anomaly Detection

Recent research on anomaly detection in multivariate time series has focused on modeling temporal and variable dependencies.Many approaches use AEs architectures for anomaly detection. OmniAnomaly [18] uses VAE-based networks to capture time dependencies and stochasticity in latent space. USAD [1] introduces adversarial AEs to model time representations and combines generator and discriminator scores for anomaly detection. RAMED [17] addresses temporal error accumulation using a multiresolution ensemble decoder, while MemAE [4] adds a memory module to enhance AE's generalization and detection capability. In modeling variable dependencies, MTAD-GAT [26] applies attention mechanisms for temporal and spatial anomaly detection, and GDN [3] uses graph networks to represent variable relationships and uses a prediction-based approach to accomplish anomaly detection. AddGraph [27] integrates dynamic graph structures with GRUs to capture both short-term and long-term patterns, while MSCRED [25] models correlations through system signature matrices. MTS-DCGAN [11] combines sliding windows and forgetting mechanisms in the anomaly detection phase to focus on the contribution of samples at different distances and achieve good results. CCG-EDGAN [10] combines cross-correlation graphs and encoder–decoder GAN to learn sequential correlation

features among multiple time series and achieve good performance. MEGA [20] combines DWT, AE, and GCN to better detect subtle anomalies and capture multiscale dependencies.

2.2 DWT in Neural Networks

In signal processing, signals like temperature or KPIs are composed of different frequency components.Researchers have recently combined DWT with neural networks to capture multiscale frequency representations. For example, mWDN [21] uses multilevel DWT within deep neural networks for improved frequency learning in time-series analysis. DWT has also been applied successfully in computer vision tasks due to its noise robustness and efficiency. WaveCNet [9] integrates wavelets with CNNs for noise-resistant image classification, MW-GAN [19] enhances video quality using wavelets and GANs, WaveletMonoDepth [15] applies wavelet decomposition for depth prediction, and STMFANet [6] combines wavelets with spatial-temporal networks for video prediction.

2.3 GCNs

GCNs have become a powerful tool for learning on graph-structured data by aggregating information from neighboring nodes through graph convolutions. Traditional GCNs, introduced by Kipf and Welling [7], operate in the spectral domain and excel in tasks like semi-supervised node classification. However, issues like oversmoothing and high computational cost have led to various improvements. SGC [22] simplifies GCNs by removing nonlinearities and collapsing weight matrices, improving efficiency and reducing oversmoothing. GraphSAGE [8] introduces inductive learning to handle dynamic graphs. GCNs are now being applied to time-series anomaly detection. GDN [2] models variable relationships using graph networks, while MTAD-GAT [13] integrates attention mechanisms to capture both temporal and variable dependencies. Our work leverages SGC within an autoencoder framework to enhance detection of subtle dependencies and anomalies in multivariate time series.

3 Methodology

3.1 Overview

Our proposed methodology enhances the MEGA framework by integrating SGC to improve anomaly detection capabilities in multivariate time series. This approach capitalizes on the strengths of SGC to simplify the model architecture while retaining effectiveness.The specific structure is shown in Figure 1.

3.2 Multiscale Discrete Wavelet Decomposition

In the DWT segment of our methodology, we employ a multilevel DWT to decompose the original multivariate time series into a set of frequency components

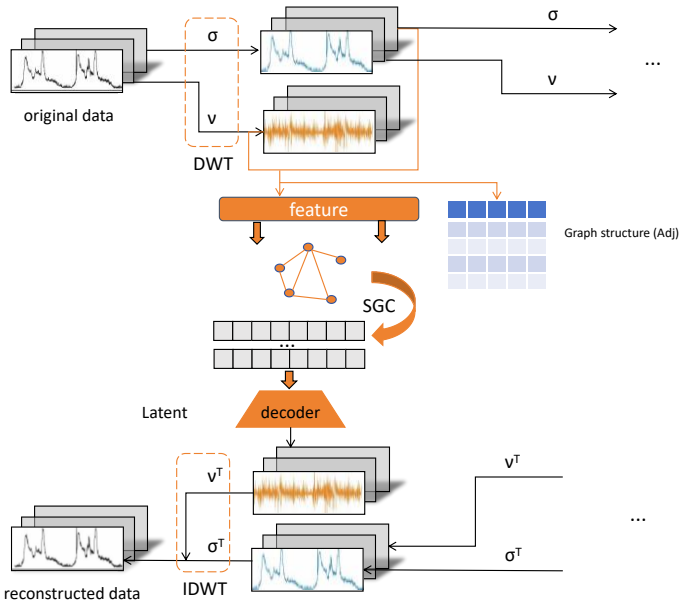


Fig. 1. This is a schematic of one layer of the model in this paper, mainly consisting of three parts, and the other two layers have the same structure as in the schematic.

that represent different scales within the data. This process is crucial for detecting anomalies that may manifest at various frequency bands.

The DWT uses a pair of filters—a low-pass filter (σ) and a high-pass filter (ν)—to decompose the time series into low-frequency components and high-frequency components. The decomposition is performed iteratively, with each iteration halving the length of the time series while doubling the number of frequency components.

The mathematical representation of the DWT process is as follows:

$$f_l^{i+1} = \sigma^i f_l^i \tag{1}$$

$$f_h^{i+1} = \nu^i f_h^i \tag{2}$$

where f_l^i represents the low-frequency components at the i-th level, f_h^i represents the high-frequency components at the i-th level.

We then feed the multiscale frequency components into the encoder to extract their latent representations. These representations are subsequently used by the decoder to regenerate the frequency components. Finally, we reconstruct the original time series using the IDWT.

3.3 Graph Structure Learning

After decomposing the multivariate time series into multiple frequency components using DWT, we construct a dynamic graph structure at each scale. The

relationships between variables are modeled as edges, and the characteristics of each variable are represented as nodes. To efficiently capture these dependencies, we employ the SGC, which simplifies the traditional GCN by removing the nonlinearity and collapsing weight matrices.

The adjacency matrix A is defined as:

$$A = \text{LeakyReLU}\left(\tanh\left(\kappa\left(E_1 E_2^T - E_2 E_1^T\right)\right)\right) \tag{3}$$

where E_1 and E_2 are embeddings learned adaptively during training, and κ is a scaling factor. This formulation ensures the asymmetry of the adjacency matrix, reflecting the unidirectional impact of anomalies among variables.

Given the node features f_i from the DWT, the SGC operation can be expressed as:

$$f_i' = \tilde{A} f_i \tag{4}$$

where $\tilde{A} = \tilde{D}^{-1/2} A \tilde{D}^{-1/2}$ and $\tilde{D}_{ii} = 1 + \sum_j A_{ij}$ is the degree matrix. Here, f_i' represents the updated node features after the SGC layer, capturing the aggregated information from neighboring nodes.

We incorporate the frequency information of the time series into the process of SGC to obtain the latent representation. When anomalies occur, the frequency of the series changes, and the variable dependence of the graph convolution output is affected, which are helpful for anomaly detection. The entire encoding process can be formalized as follows:

$$z = E(f_l, f_h) \tag{5}$$

where E is the encoder function that maps the low-frequency components f_l and high-frequency components f_h to the latent representation z.

3.4 Frequency Generator and IDWT

Once we have the multiscale representation of the latent space, we utilize multiscale frequency generation and synthesis to reassemble the original time series.

Frequency Generator

The frequency generator uses the encoded latent representations z to generate the multiscale frequency components. The generation process can be expressed as:

$$f_l^{i+1} = \sigma^i z \tag{6}$$

$$f_h^{i+1} = \nu^i z \tag{7}$$

where f_l^{i+1} and f_h^{i+1} are the low-frequency and high-frequency components at scale, σ^i and ν^i are the learned low-pass and high-pass filters.

IDWT

The IDWT is used to reconstruct the original time series from the generated multiscale frequency components. The reconstruction process combines the low-frequency and high-frequency components at each scale and can be expressed as:

$$f_l^i = \text{IDWT}(f_l^{i+1}, f_h^{i+1}) \tag{8}$$

where f_l^i is the reconstructed low-frequency component at scale i , IDWT is the inverse discrete wavelet transform operation that combines the low-frequency and high-frequency components to reconstruct the time series at the previous scale.

By integrating the frequency generator and IDWT, the MEGA framework effectively reconstructs the original time series, enabling the detection of anomalies through the comparison of the reconstructed and original time series.

3.5 Anomaly Detection

In the test phase, the loss obtained by feeding samples into the trained model is used as the score for anomaly determination. The loss function can be expressed as:

$$L = \alpha\|X - X'\|_2^2 + \beta\|f_{1h} - f'_{1h}\|_2^2 + \gamma\|f_{2h} - f'_{2h}\|_2^2 + \delta\|f_{3l} - f'_{3l}\|_2^2 + \lambda\|f_{3h} - f'_{3h}\|_2^2 \tag{9}$$

where $\alpha, \beta, \gamma, \delta$, and λ are the weighting coefficients that can be used to adjust the attention for different scale frequencies. X and X' are the original and reconstructed time series, respectively. $f_{1h}, f'_{1h}, f_{2h}, f'_{2h}, f_{3l}, f'_{3l}, f_{3h}$, and f'_{3h} are the high-frequency and low-frequency components at different scales.

If we want to place more emphasis on the reconstruction of frequency components at different scales in the anomaly scores, we can increase the corresponding loss weights during the training phase. In the final step, an anomaly threshold is set to classify samples with scores above the threshold as abnormal and those below the threshold as normal, thereby completing the anomaly detection.

4 Experiment

4.1 Datasets and Evaluation Metrics

In this article, we use three public datasets of multivariate time-series anomaly detection to test the effectiveness of our model, including two real-world datasets Mars Science Laboratory rover (MSL), Soil Moisture Active Passive satellite (SMAP) [5] collected from NASA, each of which has a training and testing datasets. Anomalies in both testing subsets have been labeled. And a five-week-long dataset was collected from a large Internet company, server machine dataset (SMD) [18] which is divided into two subsets of equal size: a training set and a testing set. MSL contains 27 entities, whose dimension is 55, while SMAP contains 55 entities, whose dimension is 25. SMD has 28 entities with 38 metrics of each. The detailed information on these datasets is shown in Table 1.

Table 1. Details of the Datasets

Dataset	Train	Test	Dimensions	Anomalies (%)
MSL	58317	73729	27×55	10.72
SMAP	135183	427617	55×25	13.13
SMD	708405	708420	28×38	4.16

In order to compare the performance between our model and other baselines, we use the same data preprocessing method and evaluation metrics as in previous works [18], [1] including

$$F1 = \frac{2 \times P \times R}{P + R} \tag{10}$$

$$P = \frac{TP}{TP + FP} \tag{11}$$

$$R = \frac{TP}{TP + FN} \tag{12}$$

where TP is the number of true positives, FP is the number of false positives, and FN is the number of false negatives.

We do not focus on the specific strategy for selecting the anomaly threshold. If a model does not provide a method to select thresholds, we use a brute force search to find the best F1 scores, ensuring a uniform threshold search strategy. To maintain consistency in the experimental settings, we apply the point-adjust strategy: for a segment anomaly, any subset that triggers an alert is considered acceptable. Therefore, if the model detects any observation as an anomaly within a ground-truth anomaly segment, we assume the entire segment is correctly detected. For general point anomalies, no adjustment is made.

4.2 Baseline Methods

We compare our model with five unsupervised anomaly detection methods to demonstrate the performance of our model including: IF [12] is a classical anomaly detection algorithm in machine learning that exploits the assumption of low data density at outlier points for data partitioning; AE is the most fundamental reconstruction-based deep model in the field of anomaly detection; DAGMM [28] combines AE and the Gaussian mixture model (GMM) to estimate the density of the representations in the latent space; LSTM-VAE [14] replaces the feed-forward neural network in VAE with an LSTM to capture the temporal dependence; MEGA [3] have been introduced in the related work.

4.3 Experimental Results

Table 2 shows the overall performance of our model and the other baselines. As shown in the table, MEGA and Ours, which use multiscale frequency modeling,

achieve better results in accuracy, the metric evaluated for the anomaly detection task.

Of these models, IF had the worst results.IF attempts to distinguish abnormal data from normal data in the raw feature space, but it struggles with high-dimensional, complex data distributions. DAGMM also performs relatively poorly compared to other benchmarks due to the lack of time-dependent modeling. Since anomalies in time-series data are usually caused by temporal variations, multiscale frequency modeling can capture time dependence more effectively from multiple perspectives.

AE and LSTM-VAE are trained to utilize temporal information in a reconstructive manner. However, they have higher false negatives because their reconstructions are performed only in the original space without proper constraints in the latent space. These models do not specialize in subtle pattern anomalies and do not take into account anomalous relationships between variables, hence their limited performance.

In contrast, Our SGC-based models utilize both temporal and spatial relationships and are highly competitive in terms of performance; MEGA's GCN integrates multi-scale frequency information to capture spatial anomalies, while SGC in Ours simplifies the graph learning process by reducing computational overhead and removing non-critical operations. This allows Ours to improve its efficiency while maintaining strong performance, especially on the MSL dataset, where it even outperforms MEGA in terms of F1 score. although Ours is slightly inferior to MEGA in terms of overall F1 score, the trade-off in computational efficiency makes it a practical alternative.

The effective utilization of multi-scale frequency information and spatial anomaly detection using graph networks is responsible for MEGA's excellent performance. However, our model provides a more computationally efficient model that strikes a balance between performance and resource utilization, especially in high-dimensional time series anomaly detection tasks.

Table 2. Performance of different models

Model	**MSL**			**SMAP**			**SMD**		
	P	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
IF	0.5681	0.6740	0.5984	0.4423	0.5105	0.4671	0.5938	0.8532	0.5866
AE	0.8535	0.9748	0.8792	0.7216	0.9795	0.7776	0.8825	0.8037	0.8280
DAGMM	0.7562	0.9803	0.8112	0.6334	0.9984	0.7124	0.6730	0.8450	0.7231
LSTM-VAE	0.8599	0.9756	0.8537	0.7164	0.9875	0.7555	0.8698	0.7879	0.8083
MEGA	0.8561	0.8223	0.8388	0.9694	0.5564	0.7070	0.9835	0.9992	0.9913
Ours	0.8689	0.8118	0.8394	0.9300	0.5611	0.6999	0.8707	0.9974	0.9297

5 Conclusion

In this paper, we introduce an unsupervised multivariate time-series anomaly detection framework based on multiscale frequency decomposition and generation. We employ SGC instead of traditional Graph Convolutional Networks, which significantly reduces training time and computational resource requirements. Experiments on three representative datasets (MSL, SMD, and SMAP) demonstrate that our model can maintain or even improve accuracy while reducing model complexity.

References

1. Audibert, J., Michiardi, P., Guyard, F., Marti, S., Zuluaga, M.A.: Usad: Unsupervised anomaly detection on multivariate time series. In: Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining. pp. 3395–3404 (2020)
2. Chaovalitwongse, W.A., Fan, Y.J., Sachdeo, R.C.: On the time series k-nearest neighbor classification of abnormal brain activity. IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans **37**(6), 1005–1016 (2007)
3. Deng, A., Hooi, B.: Graph neural network-based anomaly detection in multivariate time series. In: Proceedings of the AAAI conference on artificial intelligence. vol. 35, pp. 4027–4035 (2021)
4. Gong, D., Liu, L., Le, V., Saha, B., Mansour, M.R., Venkatesh, S., Hengel, A.v.d.: Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 1705–1714 (2019)
5. Hundman, K., Constantinou, V., Laporte, C., Colwell, I., Soderstrom, T.: Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding. In: Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining. pp. 387–395 (2018)
6. Jin, B., Hu, Y., Tang, Q., Niu, J., Shi, Z., Han, Y., Li, X.: Exploring spatial-temporal multi-frequency analysis for high-fidelity and temporal-consistency video prediction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4554–4563 (2020)
7. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907 (2016)
8. Li, C., Guo, L., Gao, H., Li, Y.: Similarity-measured isolation forest: Anomaly detection method for machine monitoring data. IEEE Transactions on Instrumentation and Measurement **70**, 1–12 (2021)
9. Li, Q., Shen, L., Guo, S., Lai, Z.: Wavelet integrated cnns for noise-robust image classification. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 7245–7254 (2020)
10. Liang, H., Song, L., Du, J., Li, X., Guo, L.: Consistent anomaly detection and localization of multivariate time series via cross-correlation graph-based encoder–decoder gan. IEEE Transactions on Instrumentation and Measurement **71**, 1–10 (2021)
11. Liang, H., Song, L., Wang, J., Guo, L., Li, X., Liang, J.: Robust unsupervised anomaly detection via multi-time scale dcgans with forgetting mechanism for industrial multivariate time series. Neurocomputing **423**, 444–462 (2021)

12. Liu, F.T., Ting, K.M., Zhou, Z.H.: Isolation forest. In: 2008 eighth ieee international conference on data mining. pp. 413–422. IEEE (2008)
13. Ma, J., Perkins, S.: Time-series novelty detection using one-class support vector machines. In: Proceedings of the International Joint Conference on Neural Networks, 2003. vol. 3, pp. 1741–1745. IEEE (2003)
14. Park, D., Hoshi, Y., Kemp, C.C.: A multimodal anomaly detector for robot-assisted feeding using an lstm-based variational autoencoder. IEEE Robotics and Automation Letters **3**(3), 1544–1551 (2018)
15. Ramamonjisoa, M., Firman, M., Watson, J., Lepetit, V., Turmukhambetov, D.: Single image depth prediction with wavelet decomposition. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 11089–11098 (2021)
16. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning internal representations by error propagation, parallel distributed processing, explorations in the microstructure of cognition, ed. de rumelhart and j. mcclelland. vol. 1. 1986. Biometrika **71**(599-607), 6 (1986)
17. Shen, L., Yu, Z., Ma, Q., Kwok, J.T.: Time series anomaly detection with multiresolution ensemble decoding. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 35, pp. 9567–9575 (2021)
18. Su, Y., Zhao, Y., Niu, C., Liu, R., Sun, W., Pei, D.: Robust anomaly detection for multivariate time series through stochastic recurrent neural network. In: Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining. pp. 2828–2837 (2019)
19. Wang, J., Deng, X., Xu, M., Chen, C., Song, Y.: Multi-level wavelet-based generative adversarial network for perceptual quality enhancement of compressed video. In: European Conference on Computer Vision. pp. 405–421. Springer (2020)
20. Wang, J., Shao, S., Bai, Y., Deng, J., Lin, Y.: Multiscale wavelet graph autoencoder for multivariate time-series anomaly detection. IEEE Transactions on Instrumentation and Measurement **72**, 1–11 (2022)
21. Wang, J., Wang, Z., Li, J., Wu, J.: Multilevel wavelet decomposition network for interpretable time series analysis. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. pp. 2437–2446 (2018)
22. Wu, F., Souza, A., Zhang, T., Fifty, C., Yu, T., Weinberger, K.: Simplifying graph convolutional networks. In: International conference on machine learning. pp. 6861–6871. PMLR (2019)
23. Yu, J., Song, Y., Tang, D., Han, D., Dai, J.: Telemetry data-based spacecraft anomaly detection with spatial–temporal generative adversarial networks. IEEE Transactions on Instrumentation and Measurement **70**, 1–9 (2021)
24. Zeng, Z., Jin, G., Xu, C., Chen, S., Zeng, Z., Zhang, L.: Satellite telemetry data anomaly detection using causal network and feature-attention-based lstm. IEEE Transactions on Instrumentation and Measurement **71**, 1–21 (2022)
25. Zhang, C., Song, D., Chen, Y., Feng, X., Lumezanu, C., Cheng, W., Ni, J., Zong, B., Chen, H., Chawla, N.V.: A deep neural network for unsupervised anomaly detection and diagnosis in multivariate time series data. In: Proceedings of the AAAI conference on artificial intelligence. vol. 33, pp. 1409–1416 (2019)
26. Zhao, H., Wang, Y., Duan, J., Huang, C., Cao, D., Tong, Y., Xu, B., Bai, J., Tong, J., Zhang, Q.: Multivariate time-series anomaly detection via graph attention network. In: 2020 IEEE international conference on data mining (ICDM). pp. 841–850. IEEE (2020)

27. Zheng, L., Li, Z., Li, J., Li, Z., Gao, J.: Addgraph: Anomaly detection in dynamic graph using attention-based temporal gcn. In: IJCAI. vol. 3, p. 7 (2019)
28. Zong, B., Song, Q., Min, M.R., Cheng, W., Lumezanu, C., Cho, D., Chen, H.: Deep autoencoding gaussian mixture model for unsupervised anomaly detection. In: International conference on learning representations (2018)

An Inferential Graph Convolution Network for Explaining Traffic Congestion

Qing Zhai, Jiayi Chen, Yifan Yin, Zi'ang Yang, and He Li*

Xidian University, Xi'an 710000, China
heli@xidian.edu.cn

Abstract. Due to the growth of vehicles, traffic congestion is becoming increasingly serious. However, existing methods are used for predicting traffic congestion, which cannot been applied for evaluating traffic congestion. In this paper, we propose an Interpretable Graph Convolution Network called ShapGCN for explaining the reason of traffic congestion by considering its physical and semantic neighbors. Specifically, we first design the physical neighbor embedding and semantic neighbor embedding to collectively encode complex extern factors as well as the complex traffic cascade pattern. To interprete traffic congestion in a complex traffic cascade environment, we use the approximation of shapley value to comprehensively quantify the discovered regions and their importance score. We conduct extensive experiments on the real traffic dataset. The experiment results show our ShapGCN can well explain the reason of traffic congestion.

Keywords: Traffic Congestion Prediction · Graph Convolution Network(GCN) · Explainable Analysis.

1 Introduction

In the process of urbanization, the number of cars has in creased excessively with the continuous expansion of urban boundaries. Some of the previous road structure cannot meet the current large traffic flow, resulting in serious traffic congestion. For example, the roads near the schools will encounter traffic jams at the morning peak and evening peak. If we can explain the causes of traffic congestion, the government can optimize road structure and traffic strategy to alleviate traffic pressure according to these reasons. Thus, investigating how to interpret the reason of traffic congestion is of paramount importance for city construction and traffic control.

Potential research works can be used for solving the traffic prediction for the detector network. In early days, the task is simple viewed as the prediction of of a multivariate time series. Therefore, time series models[16], converts non-stationary sequences into stationary sequences by differential methods and then makes predictions. But it is more suitable for some stationary sequences. This is far from the actual traffic flow changes, so the effect is not good. In addition,

it tends to view the historical traffic flow of each area in isolation, without considering the spatial-temporal correlation between areas, which is too simplified for the problem, so it is used less now. Some researchers [25, 3, 2, 27] use CNN to capture spatial features. However, the above methods cannot be applied to non-Euclidean structure data. In recent years, some methods based on GNN [28, 8] are applied for traffic prediction.

Explaining traffic congestion is very challenging due to two aspects: i) complex traffic cascade pattern, traffic congestion is caused by the convergence of traffic flow from many roads. ii) complex external factors, the POI information, weather condition will affect traffic status.

However, most methods can only predict the traffic flow of different regiond and can not explain the reason of traffic congestion. In recent years, many scholars focus on explainable network model. The traditional explainable methods generate a sensitivity map for the input data to calculate the importance of the underlying substructures [19]. Gradient-based saliency maps[1], Class Activation Mapping (CAM) [24], and Excitation Backpropagation (EB) [24]. After that, two variants: gradient-weighted CAM (Grad-CAM) [18] and contrastive EB. They are usually applied for two different applications: visual scene graphs and molecular graphs. STANE [14] introduces a spatial-temporal attention mechanism to learn the attention parameters to fulfill the interpretation requirements. KerGNN [6] integrates graph kernels into the message passing process of GNNs and visualize the graph filters to show the important features of input data.

However, the above methods are designed for interpretable network model and cannot explain the reason of traffic congestion. In order to tackle the aforementioned problem, we propose a model called ShapGCN to explain the reason of traffic congestion. But we are still facing the following challenges for in interpreting traffic congestion to increase traffic controller understanding and trust in traffic congestion: i) complex traffic cascade pattern, traffic congestion is caused by the convergence of traffic flow from many roads. ii) complex external factors, the POI information, weather condition will affect traffic status.

In order to address these problems, we propose the Inferential Graph Convolution Network called ShapGCN for Explaining Traffic Congestion. First, we employ spatial-temporal position embedding to encode spatial-temporal position information. In addition, we introduce the feature mask matrix to mask some features to reflct the importance of these features. Finally, the approximation of shapley value is used to comprehensively quantify the discovered regions and their importance score for interpreting traffic congestion. In summary, our contributions in this paper are as follows:

1.We propose a model called ShapGCN, which can explain the reason of traffic congestion well by the the feature mask matrix.

2.We design a spatial-temporal position embedding to encode spatial-temporal position information in the complex traffic cascade environment.

3.We conduct extensive experiments on real-world datasets, showing our ShapGCN can well explain the reason of traffic congestion.

The rest of the paper is organized as follows. Section 2 shows the related work. In Section 3, we describe the definitions and studied problem, and then we present the architecture of ShapGCN in detail in Section 4. Section 5 give extensive experiments to verify the explainable performance. Finally, our work is concluded at the end of this paper.

2 Related Work

2.1 Traffic prediction

Traffic flow forecastTraffic flow forecasting is critical to urban safety, so many researches have emerged in recent years. The method used is also highly relevant to the development of deep learning in recent years.

The classic solutions are mainly traditional time series models or machine learning methods. The most classic of the time series models is Auto-Regressive Integrated Moving Average (ARIMA)[16], which converts non-stationary sequences into stationary sequences by differential methods and then makes predictions, and many models are derived[12, 13]. The classic machine learning methods Vector Auto-Regressions (VAR)[12] and and Support Vector Regression (SVR)[20]. have also been used, and are more suitable for some stationary sequences. This is far from the actual traffic flow changes, so the effect is not good. These methods tend to view the historical traffic flow of each area in isolation, without considering the spatio-temporal correlation between areas, which is too simplified for the problem, so it is used less now.

Deep learning has developed rapidly in recent years and has greatly improved the accuracy of traffic prediction. At the earliest, STDNN[26] mapped trajectories to grid maps and used DNN to make predictions, and achieved good results. After that, many methods of extracting spatio-temporal correlation using convolution emerged, such as STResNet[25], DeepFTP[3], MGSTC[2], MDL[27], etc[9, 15]. On the other hand, the RNN[5], LSTM[10], GRU[4] models, which are widely used in natural language processing, have also been successful. They are very suitable for modeling the temporal dependence of spatio-temporal data. In order to achieve better results, many works combine spatial correlation and temporal dependence, such as DMVST-Net[23], DeepUrbanEvent[23] etc.

Graph neural networks have received the attention of researchers after being paid attention to from graph representation learning[1, 7, 17] and used in the field of transportation. Graphs in the transportation field often refer to spatio-temporal graphs, indicating that the relationship between nodes will be affected by other nodes and time. In common methods, graph convolution is used to replace CNNs, such as T-GCN[28], ASTGCN[8]. DCRNN combines GCN and RNN models for traffic prediction. In addition, after the birth of the transformer, the combination of transformer and GNN has further promoted the solution of problems[22], such as GMAN[29] and STGNN[21].

2.2 Interpretable model

In recent years, many scholars focus on explainable network model. The traditional explainable methods generate a sensitivity map for the input data to calculate the importance of the underlying substructures [19]. Gradient-based saliency maps[19], Class Activation Mapping (CAM) [24], and Excitation Backpropagation (EB) [24]. After that, two variants: gradient-weighted CAM (Grad-CAM) [18] and contrastive EB. They are usually applied for two different applications: visual scene graphs and molecular graphs. STANE [14] introduces a spatial-temporal attention mechanism to learn the attention parameters to fulfill the interpretation requirements. KerGNN [6] integrates graph kernels into the message passing process of GNNs and visualize the graph filters to show the important features of input data.

3 Problem Definition

In this section, in order to describe our approach clearly, we formulate the basic symbol definition and problem statement in the paper.

3.1 Definition 1 (Traffic Network)

The traffic network is regarded as a graph represented by $G = (V, E, A)$ where V is the set of nodes and E is the set of edges. The number of nodes denotes $|V| = N$. The adjacency matrix $A \in \mathbb{R}^{N \times N}$ is used to represent the nodes' proximities. The adjacency matrix only contains 0 and 1. If node V_i and node V_j are adjacent, A_{ij} in A is equal to 1. Otherwise, it is equal to 0.

3.2 Definition 2 (Graph Signal Matrix)

At each time interval t, the traffic network has a graph signal matrix $X_t \in \mathbb{R}^{N \times C}$, where C represents the number of attribute features.

3.3 Problem (Explainable traffic congestion)

The aims of explaining traffic congestion is using a feature mask matrix to mask some features to show the importance of these features.

4 Methodology

The overview of our framework is shown in fig 1, it can be divided into three core parts. 1) spatial-temporal neighbor embedding module 2) featue mask modules. 3) a output layer. In order to interpret model knowledge for a target class, our method first extracts features automatically and then computes an importance score for features of every neighbor according to our proposed ShapGCN. Finally, the importance scores of every neighbor for the whole class are given for class-wise explanation.

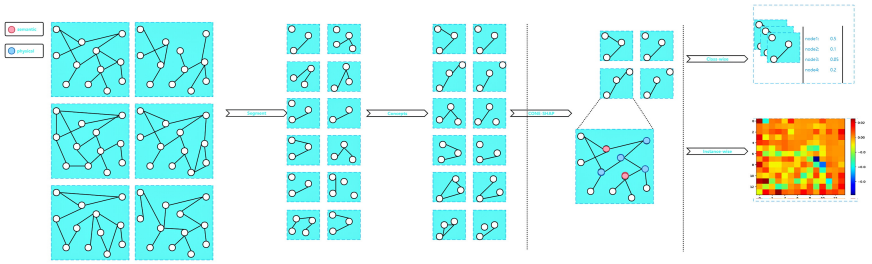


Fig. 1: The framework of our model

4.1 Fearues Discovery

Road feaures are defined as prototypes that are understandable for traffic congestion. Specifically, These feaures can be the position information, temporal information, etc. Since there are no user-defined features in real traffic scenarios, a method to discover traffic features automatically is needed. To extract such kind of traffic features, We denote the features as $C = \{C_1, C_2, ..., C_m\}$, where $C_i = \{c_{i,1}, c_{i,2}, .., C_{i,m}\}$, $c_{i,j}$ denotes the j^{th} feature in the i^{th} region, m is the number of the traffic features. To avoid missing the meaningful features information, we have considered both physical neighbors and semantic neighbors. We evaluate the importance of every feature of physical neighbors and semantic neighbors. We can obtain the physical neighbors by the road structure and semantic neighbors by DTW algorithm.

4.2 Features-based Neighbor Shapley

To measure the contribution of a features in an region for an explained model, we apply a counterfactual method which considers how the prediction of the model will change if this segment is absent. For prediction tasks, let g be the last layer before the softmax operation and g_k represents the logit values of feature k. Similar to CONE-SHAP[11], the value of a node of features k for the model is calculated as:

$$v_k(s) = g_k(x) - g_k(x \setminus \{s\}). \tag{1}$$

For convenience, we denotes $v_k(s)$ as $v(s)$.

Shapley Value. We consider all the N segments in a traffic image as a union, and each of them is a player. For a particular player i, let S be a subset that contains player i and $S \setminus \{i\}$ denotes the subset without the participation of i, then the contribution of i to the subset S is computed as:

$$\triangle v(i, S) = v(S) - v(S \setminus \{i\}). \tag{2}$$

Where $v(\cdot)$ is the utility function, then $\triangle v(\cdot)$ becomes the marginal contribution of the Shapley value. Thus,the Shapley Value of play i is the weighted average

of marginal contribution in all of the subset:

$$\phi_v(i) = \frac{1}{N}\sum_{j=1}^{N}\frac{1}{C_{j-1}^{N-1}}\sum_{S\in S_j(i)}\triangle v(i,S), \tag{3}$$

where $S_j(i)$ denotes the set with size j that contains the i_{th} segment. However, as the number of players increases, the computational complexity of the Shapley value grows exponentially. Since each traffic image contains more than a hundred segments, it is expensive for a computer to compute the true Shapley value. Therefore, recent studies have replaced the true Shapley values with approximations in different cases.

Approximation of Shapley Value. The regions can be treated as the nodes of a fully connected graph, where any two players are connected since they might have correlations during a game. In the application of image classification, the segments of an image can also be treated as nodes, but each node only connects with its neighbors. Here, we define the neighbors $N(i)$ of the i_{th} segment are those segments which are adjacent to it (physical neighbors) or belong to the same concept as it (semantic neighbors). Based on the assumption that participants which are not the neighbors of i hardly affect its contribution for a model's inference procedure, the Shapley Value of i in Equation above can be approximated as:

$$\phi_v^N(i) = \frac{1}{|N(i)|}\sum_{j=1}^{|N(i)|}\frac{1}{C_{j-1}^{|N(i)|-1}}\sum_{i\in S}\triangle v(i,S). \tag{4}$$

Considering that a segment may contain a large amount of neighbors in an instance, we adopt sample-based method to estimate $\phi_v^N(i)$ in order to further reduce the computation costs. Concretely, we first sample k nodes from $N(i)$ and denotes it as $N_k(i)$, and then compute the Shapley Value in the $N_k(i)$. This procedure will repeat M times, and we take the average of these results as the COncept-based NEighbor Shapley Value(CONE-SHAP) of i:

$$\tilde{\phi}_v^N(i) = \frac{1}{M|N_k(i)|}\sum_{t=1}^{M}\sum_{j=1}^{|N_k(i)|}\frac{1}{C_{j-1}^{|N_k(i)|-1}}\sum_{i\in S}\triangle v(i,S). \tag{5}$$

Next, we will introduce how to employ the approximation of Shapley Value from Equation 11 to interpret model knowledge from both instance-wise and class-wise.

4.3 Model Explaining

Instance-wise Explanation. In order to help users understand the basis for a model's reasoning procedure intuitively, we provide concept-based saliency maps to interpret model knowledge on the instance-level. The contribution of each segment of each instance is assigned according to its CONE-SHAP Value

$\tilde{\phi_v^N}(i)$. Compared to perturbation-based methods which explain a model in fine-grained features, our CONE-SHAP focuses on the concept-based explanation, which is more human-friendly.

Class-wise Explanation. To interpret model knowledge on the class level, our method distributes the concept scores to indicate which concept contributes more to the model's prediction on the explained class. A concept is considered important if all of its belongings own a high Shapley Value. Since we have gotten a group of possible concepts in the concept discovery procedure, for a concept C_i, we compute its score by averaging all of the approximate Shapley Values of its segments:

$$CS_i = \frac{1}{|SC_i|} \sum_{c_{i,j} \in C_i} \tilde{\phi}_v^N(i). \tag{6}$$

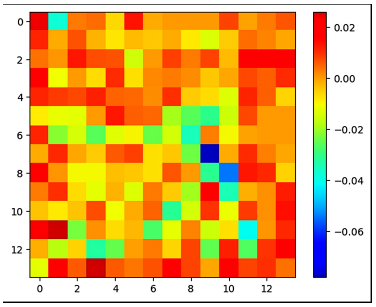


Fig. 2: The Heat Maps

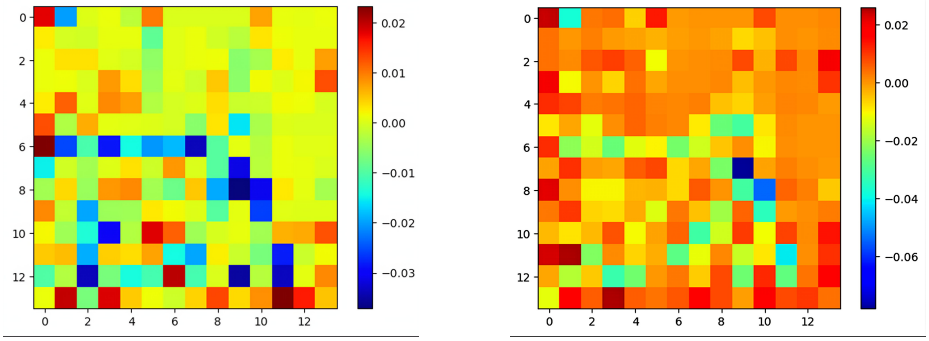
5 EXPERIMENTS

5.1 Experimental Settings

Table 1: Gradually increase the value of M

Metrics	M=1	M=2	M=3	M=4
Mean	0.00405483	0.00202741	0.00135161	0.00101370
Std	0.07392439	0.36962191	0.02464146	0.01848109

Our method can be applied to any task without any further training. To intuitively demonstrate the superiority of our method, we focus on the congestion-level classification of traffic flows in this paper.



(a) (b)

Fig. 3: Comparing two heatmaps at different times .

Table 2: Gradually increase the value of k

Metrics	k=2	k=3	k=4	k=5
Mean	0.00201455	-0.00235587	-0.00124802	-0.00149922
Std	0.06388019	0.01326836	0.00777133	0.00885320

Dataset. The data set we use comes from the Chengdu urban traffic flow data set in November 2016, which contains data of 196 nodes every half hour for 30 days in November. Then we use the average of node's concept scores for class-wise interpretation, and select the data of the first half hour of the third day for instant-wise interpretation.

Settings for neighbors. For each explained instance, we first calculate the inflow and outflow information of each node in different time periods, and then obtain the semantically adjacent neighbors of these nodes by calculating the flow information of these nodes. The meaningful semantic neighbors of nodes are different in different node positions, we use the *DTW* algorithm to find semantic neighbors similar to the node timing. Then, the physical neighbors physically adjacent to the node are calculated through the space-time matrix obtained from the data set. The union of the last two neighbor sets is the node's neighbor set.

5.2 Instance-wise Explanation

Explanation with Heat Maps. We provide the same fine-grained heat map as the input node to indicate which node of the instance is more important to the congestion situation of the traffic flow. Since we get the approximate Shapley value of each node at each timestamp, we treat these values as the score the node gets, and then for each node, we average the approximate Shapley values of the node at all moments to obtain the final scores and display them on a heat map.

Table 3: Physical neighbors and Semantic neighbor

Neighbors	physical	semantic	both
Mean	-0.00201529	0.00020384	0.00235587
Std	0.01425835	0.01326836	0.01326836

Figure 4 shows our CONE-SHAP heatmap for each node, where the importance score for each node is scaled between -1 and 1 by dividing by the absolute value of the largest number, for unit settings with scores below 0 is blue, over zero is set to red. On the graph you can see that our concept-based heatmap is easier for humans to understand.

Different Importance of Concepts on Different Instances. Intuitively, even the same node might have different importance to different instances. Based on the CONE-SHAP value at each timestamp, our method can estimate the importance of each node on each instance as follows: Firstly, we find out all the nodes in the instance. Then, for each node, we calculate the CONE-SHAPE value belonging to the node of Equation 11. Finally, we can estimate concept importance by summing the CONE-SHAPE values of its nodes. Figure 2 illustrates an example of our CONE-SHAPE interpretation of a traffic flow congestion class classification model. At the class level, our method shows that some nodes are very important for the recognition of the overall traffic jam, while in specific instances, the importance of these nodes varies. For example, in the first picture of Figure 3, the most important nodes are nodes 1, 2, 3, but in the second picture, the importance scores of these three points are not the most important.

5.3 Class-wise Explanation

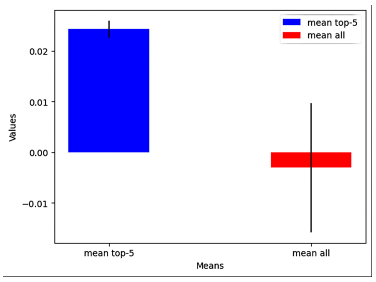


Fig. 4: The top-5 node's Mean

Validating the Performance of Concepts. To measure the most important concepts in the explanatory model, we compare the mean and variance of the five nodes with the highest importance scores in the explanatory model to

the mean variance of the population.As shown in Figure 4:The average value of the five nodes with the highest importance score is much greater than the average value of all nodes.

Analysis of the Hyperparameters for Approximating Shapley Value. We approximate the true Shapley value by sampling from the neighbors, as shown in *Equation* 11. Too large M and k will bring pressure on the computational cost, while too small M and k will lead to inaccurate estimation. So we experimented to find the right M and k. In order to choose M, we first fix the value of k to 3, gradually increase the value of M to observe the change of the population mean, the results shown in Table 1 show that in our setting, setting M to 1 is enough to approximate the Shapley value . Similarly, we set M to 1 and gradually increase k. We found that when k is 2, a higher Shapely value is reached, as shown in Table 2. Therefore, we set M to 1 and k to 2 in our experiments.

Ablations experiment. In our experiments, the neighbor nodes of a node include semantic nodes and physical nodes, so we use ablation experiments to observe whether the two types of neighbor nodes have a great impact on the calculation of the final result. As shown in table 3, when considering two types of neighbors at the same time, the effect is better than only considering one type of neighbor nodes, which shows that the idea of considering two types of neighbor nodes in our method is reasonable.

6 Conclusions

In this paper, we propose a model named ShapGCN to achieve a better explanation of traffic congestion. We design the spatial-temporal position embedding and propose the spatial-temporal convolution module. By using the approximation of shapley value to comprehensively quantify the discovered regions and their importance score, ShapGCN has the ability to interprete traffic congestion in a complex traffic cascade environment. We evaluate our ShapGCN on the real dataset and the results show that our model achieve great interpretability.

References

1. Cao, S., Lu, W., Xu, Q.: Grarep: Learning graph representations with global structural information. In: Proceedings of the 24th ACM international on conference on information and knowledge management. pp. 891–900 (2015)
2. Chen, C., Li, K., Teo, S.G., Zou, X., Li, K., Zeng, Z.: Citywide traffic flow prediction based on multiple gated spatio-temporal convolutional neural networks. ACM Transactions on Knowledge Discovery from Data (TKDD) **14**(4), 1–23 (2020)
3. Chen, Y., Chen, F., Ren, Y., Wu, T., Yao, Y.: Poster: Deeptfp: Mobile time series data analytics based traffic flow prediction. In: Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking. pp. 537–539 (2017)
4. Cho, K.: Learning phrase representations using rnn encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078 (2014)

5. Connor, J.T., Martin, R.D., Atlas, L.E.: Recurrent neural networks and robust time series prediction. IEEE transactions on neural networks **5**(2), 240–254 (1994)
6. Feng, A., You, C., Wang, S., Tassiulas, L.: Kergnns: Interpretable graph neural networks with graph kernels. In: Proceedings of the AAAI conference on artificial intelligence. vol. 36, pp. 6614–6622 (2022)
7. Grover, A., Leskovec, J.: node2vec: Scalable feature learning for networks. In: Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 855–864 (2016)
8. Guo, S., Lin, Y., Feng, N., Song, C., Wan, H.: Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. In: Proceedings of the AAAI conference on artificial intelligence. vol. 33, pp. 922–929 (2019)
9. Guo, S., Lin, Y., Li, S., Chen, Z., Wan, H.: Deep spatial–temporal 3d convolutional neural networks for traffic data forecasting. IEEE Transactions on Intelligent Transportation Systems **20**(10), 3913–3926 (2019)
10. Hochreiter, S.: Long short-term memory. Neural Computation MIT-Press (1997)
11. Li, J., Kuang, K., Li, L., Chen, L., Zhang, S., Shao, J., Xiao, J.: Instance-wise or class-wise? a tale of neighbor shapley for concept-based explanation. In: Proceedings of the 29th ACM International Conference on Multimedia. pp. 3664–3672 (2021)
12. Lippi, M., Bertini, M., Frasconi, P.: Short-term traffic flow forecasting: An experimental comparison of time-series analysis and supervised learning. IEEE Transactions on Intelligent Transportation Systems **14**(2), 871–882 (2013)
13. Liu, H., Tian, H.q., Li, Y.f.: Comparison of two new arima-ann and arima-kalman hybrid methods for wind speed prediction. Applied Energy **98**, 415–424 (2012)
14. Liu, Z., Zhou, D., He, J.: Towards explainable representation of time-evolving graphs via spatial-temporal graph attention networks. In: Proceedings of the 28th ACM international conference on information and knowledge management. pp. 2137–2140 (2019)
15. Ma, X., Zhong, H., Li, Y., Ma, J., Cui, Z., Wang, Y.: Forecasting transportation network speed using deep capsule networks with nested lstm models. IEEE Transactions on Intelligent Transportation Systems **22**(8), 4813–4824 (2020)
16. Min, W., Wynter, L.: Real-time road traffic prediction with spatio-temporal correlations. Transportation Research Part C: Emerging Technologies **19**(4), 606–616 (2011)
17. Perozzi, B., Al-Rfou, R., Skiena, S.: Deepwalk: Online learning of social representations. In: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 701–710 (2014)
18. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-cam: Visual explanations from deep networks via gradient-based localization. In: Proceedings of the IEEE international conference on computer vision. pp. 618–626 (2017)
19. Simonyan, K.: Deep inside convolutional networks: Visualising image classification models and saliency maps. arXiv preprint arXiv:1312.6034 (2013)
20. Smola, A.J., Schölkopf, B.: A tutorial on support vector regression. Statistics and computing **14**, 199–222 (2004)
21. Wang, X., Ma, Y., Wang, Y., Jin, W., Wang, X., Tang, J., Jia, C., Yu, J.: Traffic flow prediction via spatial temporal graph neural network. In: Proceedings of the web conference 2020. pp. 1082–1092 (2020)
22. Xu, M., Dai, W., Liu, C., Gao, X., Lin, W., Qi, G.J., Xiong, H.: Spatial-temporal transformer networks for traffic flow forecasting. arXiv preprint arXiv:2001.02908 (2020)

23. Yao, H., Wu, F., Ke, J., Tang, X., Jia, Y., Lu, S., Gong, P., Ye, J., Li, Z.: Deep multi-view spatial-temporal network for taxi demand prediction. In: Proceedings of the AAAI conference on artificial intelligence. vol. 32 (2018)
24. Zhang, J., Bargal, S.A., Lin, Z., Brandt, J., Shen, X., Sclaroff, S.: Top-down neural attention by excitation backprop. International Journal of Computer Vision **126**(10), 1084–1102 (2018)
25. Zhang, J., Zheng, Y., Qi, D.: Deep spatio-temporal residual networks for citywide crowd flows prediction. In: Proceedings of the AAAI conference on artificial intelligence. vol. 31 (2017)
26. Zhang, J., Zheng, Y., Qi, D., Li, R., Yi, X.: Dnn-based prediction model for spatio-temporal data. In: Proceedings of the 24th ACM SIGSPATIAL international conference on advances in geographic information systems. pp. 1–4 (2016)
27. Zhang, J., Zheng, Y., Sun, J., Qi, D.: Flow prediction in spatio-temporal networks based on multitask deep learning. IEEE Transactions on Knowledge and Data Engineering **32**(3), 468–478 (2019)
28. Zhao, L., Song, Y., Zhang, C., Liu, Y., Wang, P., Lin, T., Deng, M., Li, H.: T-gcn: A temporal graph convolutional network for traffic prediction. IEEE transactions on intelligent transportation systems **21**(9), 3848–3858 (2019)
29. Zheng, C., Fan, X., Wang, C., Qi, J.: Gman: A graph multi-attention network for traffic prediction. In: Proceedings of the AAAI conference on artificial intelligence. vol. 34, pp. 1234–1241 (2020)

Dynamic Group Link Prediction in Continuous-Time Interaction Network

Shijie Luo, He Li*, Xuejiao Li, and Tian Tian

Xidian University, Xi'an 710000, China
heli@xidian.edu.cn

Abstract. Recently, group link prediction has received increasing attention due to its important role in analyzing relationships between individuals and groups. However, most existing group link prediction methods emphasize static settings or only make cursory exploitation of historical information, so they fail to obtain good performance in dynamic applications. To this end, we attempt to solve the group link prediction problem in continuous-time dynamic scenes with fine-grained temporal information. We propose a novel continuous-time group link prediction method CTGLP to capture the patterns of future link formation between individuals and groups. A new graph neural network CTGNN is presented to learn the latent representations of individuals by biasedly aggregating neighborhood information. Moreover, we design an importance-based group modeling function to model the embedding of a group based on its known members. CTGLP eventually learns a probability distribution and predicts the link target. Experimental results on various datasets with and without unseen nodes show that CTGLP outperforms the state-of-the-art methods by 13.4% and 13.2% on average.

Keywords: Group Link Prediction · Continuous-Time Interaction Network · Graph Neural Network (GNN).

1 Introduction

Link prediction, aiming to predict relationships between pairs of entities, has received wide attention as the increasing importance of network data [11, 14, 17]. Since it helps us understand the inherent characteristics and evolutionary mechanisms of real-world networks, link prediction has been widely applied in many practical applications, such as knowledge graph completion [22], biochemical reaction reconstruction [18] and content recommendation [13]. Almost all existing link prediction methods focus only on relationships between pairs of entities [27], but the analysis of relationships between individuals and groups (i.e., group link prediction) also deserves attention since the patterns of relationship formation are not exclusively limited to a pair of entities [24, 23].

Nevertheless, in many real scenarios, our focus is on the prediction of future relationships between individuals and groups, such as organizers of hobby clubs expecting to invite target participants to their future events. The task

of predicting future relationships between individuals and groups is known as *continuous-time group link prediction*. Despite recent efforts, three deficiencies remain in addressing continuous-time group link prediction problem with fine-grained temporal information. First, previous methods rarely discuss future links between individuals and groups, but tend to mine missing ones. The assumption that all members are connected to the group at the same time makes the fine-grained raw temporal information missing. Second, individuals are assumed to be isolated from each other, which neglects the neighborhood information that laterally depicts dynamic link preferences. Third, equal treatment of all group members leads to ignoring the diversity of members' importance in groups.

In this paper, we propose CTGLP, a novel continuous-time group link prediction method, to infer future relationships between individuals and groups in continuous-time dynamic networks with fine-grained temporal information. We first present CTGNN, a new graph neural network (GNN) with a continuous-time neighbor sampling strategy, to learn the embeddings of individuals, where a novel aggregation function is designed to jointly capture neighborhood features and fine-grained temporal information. Second, an importance-based group modeling function is provided to model the latent representation of a group based on the embeddings of existing members. Finally, CTGLP outputs conditional probability distributions by using the embeddings of groups and finds out the link targets.

The contributions of this paper are summarized as follows.

1.We propose a novel continuous-time group link prediction method CTGLP, which learns the patterns of link formation between individuals and groups in continuous-time dynamic networks with fine-grained temporal information and predicts the future links between individuals and groups.

2.We propose a new graph neural network CTGNN to learn the representations of individuals. A continuous-time neighbor sampling strategy is designed to control the computational consumption, and an aggregation function CTAgg is presented to bias the aggregation weights of the features of sampled neighbors.

3.We propose an importance-based group modeling function that models the groups into the latent space by measuring the importance of each member to the group based on the time of link formation.

4.Extensive experiments on various datasets with and without unseen nodes are conducted to validate CTGLP and the experimental results demonstrate that CTGLP outperforms the baselines by a significant margin, with average gains of 13.4% and 13.2%.

2 Related Work

2.1 Link prediction

Based on the network structural similarity, heuristic link prediction methods, such as Common Neighbors (CN) [12] and Adamic-Adar (AA) [1], assume that edges are more likely to exist between nodes with higher similarity scores. However, they only exploit shallow topological features of networks and lack general

applicability. Besides, embedding techniques have also shown great potential in link prediction. Some embedding algorithms, such as DeepWalk [19], node2vec [6], HTNE [28], etc., calculate the possibility of generating links between nodes using the embeddings. Nevertheless, some studies [16, 4] have demonstrated that embedding models may be inferior to well-designed mechanistic methods. Recently, some well-designed methods have shown their superiority in link prediction. TDGNN [21] leverages temporal information in dynamic networks to achieve continuous-time link prediction. GNMFCA [15] predicts future links using global and local information of temporal networks. GC-LSTM [3] applies an embedded Long Short-Term Memory (LSTM) of Graph Convolutional Network to perform dynamic link prediction. Dyngraph2vec [5] integrates longer-term temporal information to learn node embeddings and predict future links. LP-ROBIN [2] utilizes incremental embedding to capture temporal dynamics and predict new connections. Despite the great success of link prediction, existing link prediction methods cannot be directly applied to group link prediction focusing on the relationships between individuals and groups because they only concentrate on the relationships between node pairs.

2.2 Group link prediction

Due to the inevitable limitations of link prediction methods applied to group link prediction, recent attempts have been made to solve the group link prediction problem. An LSTM-based model [24] is elaborately designed to address this problem. Feeding the sum of random vectors of members in a series of groups into LSTM, this model learns the embedding vectors of members and trains a classifier to predict the target. Despite the input of a series of group vectors, it ignores the neighborhood information that potentially expresses link preferences and may introduce information noise. Subsequently, a CVAE-based model [23] is proposed to estimate the probability of link existence by tuning the model parameters in a supervised manner. However, the absence of historical interaction information between individuals and groups prevents the model from addressing the problem of continuous-time group link prediction well. To further leverage historical information, CVAEH [23] additionally introduces a vector that encodes the previous groups. Nevertheless, the rough encoding of historical group information is still not a good solution. Summarizing existing group link prediction methods, they do not consider the fine-grained historical group information and fail to generalize to unseen data. Therefore, how to infer the future relationships between individuals and groups more effectively remains an open question.

3 Preliminaries

3.1 Definitions and Problem

Definition 1 (Continuous-time interaction network). *A continuous-time interaction network* $G = (V, E^T, T)$ *consists of node set* V*, edge set* E^T *and*

time set T, where $v_i \in V$ denotes the node in the network. $e_{ij}^t \in E^T$ denotes the edge/interaction between node v_i and node v_j at time $t \in T$.

Definition 2 (Group). *Individuals jointly participating in a certain event are denoted as a group, i.e., $s_i = \{v_{i,1}^{t_1}, v_{i,2}^{t_2}, \ldots, v_{i,k}^{t_k}\} \subseteq V$, where i denotes the index of a group. $v_{i,k}^{t_k}$ denotes the k-th member node of group s_i, where t_k denotes the link time and $k \geq 2$ indicates that the number of members in a group should not be less than two. $s_i \subseteq V$ indicates that the members of group s_i are from node set V.*

Problem 1 (Continuous-time group link prediction). Given a continuous-time interaction network $G = (V, E^T, T)$, there is a node set $V = \{v_1, v_2, \ldots, v_N\}$ with N nodes and a group set $S = \{s_1, s_2, \ldots, s_M\}$ with M groups. For a group $s_i = \{v_{i,1}^{t_1}, v_{i,2}^{t_2}, \ldots, v_{i,k}^{t_k}\} \subseteq V$, $max(t_1, t_2, \ldots, t_k) \leq t$ with k members observed at current time t, the purpose of continuous-time group link prediction is to predict the target $v_{i,k+1}^{t'} \in V \backslash s_i$ that is most likely to be linked to group s_i at the future time $t'(t' > t)$ based on the k known members in group s_i. Formally, it is defined as:

$$v_{i,k+1}^{t'} = \mathcal{F}(v_{i,1}^{t_1},\ v_{i,2}^{t_2},\ \ldots, v_{i,k}^{t_k}), \tag{1}$$

where $\mathcal{F}$ is the continuous-time group link prediction function. $\{v_{i,1}^{t_1},\ v_{i,2}^{t_2},\ \ldots, v_{i,k}^{t_k}\}$ denotes the k known members of group s_i observed at current time t.

4 Methodology

Figure 1 shows the architecture of CTGLP. It consists of three main components: 1) Individual Representation Learning, 2) Importance-based Group Modeling and 3) Prediction. The Individual Representation Learning component aims to learn the latent embeddings of individuals using our proposed CTGNN. The Importance-based Group Modeling component is to model the latent representations of groups. The Prediction component aims to predict the targets.

4.1 Individual Representation Learning

Given a continuous-time interaction network $G = \{V, E^T, T\}$ with the initial random embeddings of nodes $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}, \mathbf{x}_i \in \mathbb{R}^D$, individual representation learning part obtains the final latent embeddings of members $\mathbf{Z} = \{\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_N\}, \mathbf{z}_i \in \mathbb{R}^d$ using our proposed CTGNN, where D and d are dimensions.

Continuous-Time Neighbor Sampling Whereas previous GNNs simply examine k-hop neighborhood or the sampling schemes are only applicable to static networks [26], the sampling process in all convolutional layers of our CTGNN are continuous-time respected, i.e., the time of the sampled edge should be less than that of the sampled edge of the previous layer. It can be ensured that all sampled neighbors of the central node exist in the past with respect to the central node, thus ensuring that all sampled neighborhood information exists

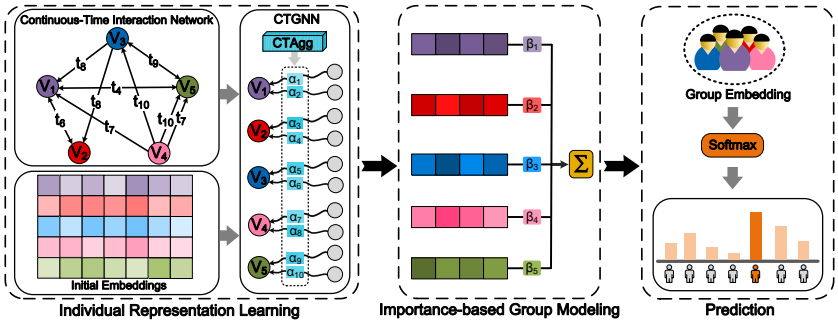


Fig. 1. The overall framework of CTGLP. CTGLP is composed of three main components: individual representation learning, importance-based group modeling and prediction.

prior to the current time during aggregation. We first define the time-limited neighbor set $\Gamma_{\mathcal{T}}(u)$ of node u at time $\mathcal{T}$:

$$\Gamma_{\mathcal{T}}(u) = \{(v,t) \mid e = (u,v,t) \in E^T \cap t < \mathcal{T}\}. \tag{2}$$

Notably, node v may appear multiple times in $\Gamma_{\mathcal{T}}(u)$ as multiple edges may exist between the same pair of nodes.

Then, in one convolutional iteration, we sample a fixed number of neighbors from $\Gamma_{\mathcal{T}}(u)$ for node u:

$$Samp = \begin{cases} \Gamma_{\mathcal{T}}(u), & |\Gamma_{\mathcal{T}}(u)| \leq \theta; \\ r_\theta(\Gamma_{\mathcal{T}}(u)), & |\Gamma_{\mathcal{T}}(u)| > \theta, \end{cases} \tag{3}$$

where $r_\theta(\cdot)$ is the random sampling operation. θ is the neighbor sampling size and θ may be different for each layer.

Performing multiple sampling, CTGNN obtains higher-order neighbors and reduces the number of neighbors involved in the computation. The l-order continuous-time sampled neighbor set of node u at time $\mathcal{T}$ can be obtained by performing neighbor sampling operations l times:

$$\hat{\mathcal{N}}^l_{\mathcal{T}}(u) = Samp_l\left(\Gamma^l_{\mathcal{T}_l}(\ldots Samp_1(\Gamma^1_{\mathcal{T}_1}(u)))\right), \tag{4}$$

where $Samp_l$ represents the sampling operation in the l-th convolutional layer. $\mathcal{T}_{i+1} < \mathcal{T}_i$ for $1 \leq i < l$, and $\mathcal{T}_1 = \mathcal{T}$.

Embedding Update By iteratively aggregating neighborhood features, GNNs learn the embeddings of nodes. A simple but effective aggregation scheme is mean operator [10, 8], which assumes that all neighbors of a central node contribute equally to the update of its new representation. However, mean operator may not be the optimal aggregation scheme for representation learning in continuous-time group link prediction, since the impact of different neighbors on the central node may vary dramatically depending on the time of link formation. Inspired

by a study in event-based social networks [20], we argue that the groups users recently linked to are typically more representative of their preferences than those they linked to earlier. The manifestation of this insight in aggregation is that a newly connected neighbor has a higher contribution to the embedding update of the central node.

We provide an aggregation coefficient α calculated by our aggregation function CTAgg to bias the contribution of each neighbor. Given the edge time t_{ij} between node u_i and u_j as well as the edge time t_{ik} between node u_i and u_k, if $t_{ij} > t_{ik}$, the aggregation coefficient α_{ij}^t of node u_j should be greater than the aggregation coefficient α_{ik}^t of node u_k. At the l-th layer of CTGNN, the embedding update of node u at time t can be denoted as follows:

$$\mathbf{n}_u^{(l)} = \mathrm{AGG}^{(l)}(\{\alpha_{uv}^t \cdot \mathbf{h}_v^{(l-1)} \ , \ v \in \hat{\mathcal{N}}_t^l(u)\}), \tag{5}$$

$$\mathbf{h}_u^{(l)} = \sigma(\mathbf{W}^{(l)} \cdot \mathrm{COM}(\mathbf{h}_u^{(l-1)}, \ \mathbf{n}_u^{(l)}) + \mathbf{w}^{(l)}), \tag{6}$$

where $\mathrm{AGG}(\cdot)$ is a function that aggregates the information of sampled neighbors and $\mathrm{COM}(\cdot)$ is a function that combines information about sampled neighborhoods and the pre-update information of the central node in the previous layer. σ is a nonlinear activation. $\hat{\mathcal{N}}_t^l(u)$ is the l-th hop sampled neighbor set of node u at time t. $\mathbf{W}$ and $\mathbf{w}$ are learnable shared parameter matrices. α_{uv}^t is the aggregation coefficient of neighbor v at time t , and it can be interpreted as the contribution of v to the embedding update of the central node u at time t. The calculation of α_{uv}^t is defined as:

$$\alpha_{uv}^t = \frac{\exp\left(t_{uv} - \ t\right)}{\sum_{v \in \hat{\mathcal{N}}_t^l(u) \cup u} \exp\left(t_{uv} - \ t\right)}, \tag{7}$$

where t_{uv} is the time of the edge between nodes u and v.

After obtaining the embedding $\mathbf{h}_u$ of node u output by the last convolution iteration, a Multiple-layer Perceptron (MLP) with activation functions is employed to attain the final representation $\mathbf{z}_u$ of node u:

$$\begin{aligned} \mathbf{e}_u^{(1)} &= \sigma(\mathbf{U}^{(1)} \cdot \mathbf{h}_u + \mathbf{u}^{(1)}), \\ \mathbf{e}_u^{(2)} &= \sigma(\mathbf{U}^{(2)} \cdot \mathbf{e}_u^{(1)} + \mathbf{u}^{(2)}), \\ &\cdots \\ \mathbf{z}_u &= \sigma(\mathbf{U}^{(j)} \cdot \mathbf{e}_u^{(j-1)} + \mathbf{u}^{(j)}), \end{aligned} \tag{8}$$

where j is the index of neural layers. $\mathbf{U}$ and $\mathbf{u}$ are learnable parameter matrices.

4.2 Importance-based Group Modeling

The practice of previous work [24] is to sum up the vectors of all group members as the vector of the group. While it is intuitively sound, it ignores the fact that the influence of different members on the group may differ greatly. To this end,

we present an importance-based group modeling strategy to represent groups into the latent space.

We first define an importance factor β to measure the importance of each group member. The value of the importance factor depends on the time when members are linked to the group, i.e., more recent members have larger importance factor values as they are intuitively more in line with the group formation trend. Formally, the importance factor of member k on group s_i is defined as:

$$\beta_{ik} = \frac{\frac{1}{\log(T-t_k)}}{\sum_{j=1}^{K} \frac{1}{\log(T-t_j)}}, \tag{9}$$

where K is the number of members in group s_i and T is the prediction time for group s_i.

Given a group s_i with K members (i.e., $s_i = \{v_{i,1}, v_{i,2}, \ldots, v_{i,K}\}$) and the embeddings of these K members (i.e., $\{\mathbf{z}_{i,1}, \mathbf{z}_{i,2}, \ldots, \mathbf{z}_{i,K}\}$), the importance-based group modeling part models the vectors of members as the vector $\mathbf{m}_i \in \mathbb{R}^s$ of group s_i, where s is the size of the group vector. The importance-based group modeling for group s_i is:

$$\mathbf{p}_i = \sum_{j=1}^{K} \beta_{ij} \cdot \mathbf{z}_{i,j}, \tag{10}$$

$$\mathbf{m}_i = \mathbf{C}_2 \cdot \sigma(\mathbf{C}_1 \cdot \mathbf{p}_i + \mathbf{c}), \tag{11}$$

where $\mathbf{C}_1$, $\mathbf{C}_2$ and $\mathbf{c}$ are learnable parameter matrices.

4.3 Prediction

The final outputs of the importance-based group modeling part are fed into an MLP, and a Softmax activation function is employed to generate the link probability distributions between candidate individuals and groups. Formally, given the latent vector $\mathbf{m}_i$ of group s_i, the prediction process is:

$$\begin{aligned} \mathbf{q}_i^{(1)} &= \sigma(\mathbf{G}^{(1)} \cdot \mathbf{m}_i + \mathbf{g}^{(1)}), \\ &\cdots \\ \mathbf{q}_i^{(k)} &= \sigma(\mathbf{G}^{(k)} \cdot \mathbf{q}_i^{(k-1)} + \mathbf{g}^{(k)}), \end{aligned} \tag{12}$$

$$\mathbf{P}_i = \mathrm{Softmax}(\mathbf{Q} \cdot \mathbf{q}_i^{(k)} + \mathbf{q}) \tag{13}$$

where $\mathbf{G}$, $\mathbf{Q}$, $\mathbf{g}$ and $\mathbf{q}$ are learnable parameter matrices. k is the index of hidden layers. $\mathbf{P}_i$ is a link probability distribution whose elements represent the connection possibility between individuals and group s_i. The index corresponding to the element with the largest value in $\mathbf{P}_i$ is the index of the target predicted by CTGLP.

Algorithm 1 Training of CTGLP

Input: Continuous-time interaction network $G = (V, E^T, T)$; set of node initial embeddings $\{\mathbf{x}_v, \forall v \in V\}$; set of groups $S = (s_1, s_2, \ldots, s_M), s_i \subset V$.
Parameter: Convolutional layer depth K; neighbor sampling size θ_k, $\forall k \in \{1, \ldots, K\}$; learnable parameters.
Output: Continuous-time group link prediction function $\mathcal{F}$.

while *model not converge* **do**
  for $i = 1 : M$ **do**
    for $u \in s_i$ **do**
      $\hat{\mathcal{N}}_{\mathcal{T}}^{K}(u) \leftarrow$ Sample-Neighbors$(u, G, K, \mathcal{T}, \theta)$
      $\mathbf{z}_u \leftarrow$ Update-Embedding$(\mathbf{x}_u, \mathbf{x}_{\hat{\mathcal{N}}_{\mathcal{T}}^{K}(u)}, K)$
    end for
    $\mathbf{m}_i \leftarrow$ Group-Modeling$(\{\beta_u \cdot \mathbf{z}_u, u \in s_i\})$
    Calculate link probabilities:
    $\mathbf{P}_i \leftarrow$ Softmax$(\mathbf{m}_i, \mathbf{G}, \mathbf{g}, \mathbf{Q}, \mathbf{q})$
    Obtain the target: $u_i \leftarrow$ argmax$(\mathbf{P}_i)$
  end for
  Update parameters by stochastic gradient descent
end while
return $\mathcal{F}$

4.4 Training

Algorithm 1 shows the overall process of training. Let $\mathbf{y}_i$ denote the one-hot encoding of the target in the i-th training sample and $\mathbf{P}_i$ be the link probability distribution output by CTGLP. Our objective function is formulated as:

$$\mathcal{L} = \frac{1}{M} \sum_{i=1}^{M} \sum_{j=1}^{N} y_{ij} \log\left(P_{ij}\right) \tag{14}$$

where M denotes the number of group samples. N denotes the number of nodes in the node set V. y_{ij} and P_{ij} are the j-th element of $\mathbf{y}_i$ and $\mathbf{P}_i$, respectively.

5 Experiments

5.1 Experimental Setup

Datasets. MovieLens-100K (ML100K for short) [9] and MovieLens-25M (ML25M for short) [9] contains rating data from users on movies. CiaoDVD [7] consists of DVD rating data. We select the rating data and regard the set of users who rate the same item as a group and take out the last member of each group as our prediction target. The number of members in a group are set to be 3 to 20. To construct the datasets without unseen nodes, we remove nodes that appear in validation and testing but not in training, and further clean the data. The statistics of datasets are shown in Table 1.

Datasets	Nodes	Edges	Groups	Unseen*
ML100K	755	59 118	590	42
CiaoDVD	8 714	165 598	4 040	1 195
ML25M	16 065	1 048 836	18 882	1 578
$\text{ML100K}_{\text{w/o}}$	650	22 683	510	0
$\text{CiaoDVD}_{\text{w/o}}$	5 766	85 562	3 341	0
$\text{ML25M}_{\text{w/o}}$	9 998	491 704	16 494	0

* The value indicates the number of nodes that are not presented during training.

Table 1. Statistics of two versions of the three datasets. Note that the subscript w/o denotes the dataset without unseen nodes.

Metrics. To evaluate the performance of methods, we introduce three widely used evaluation metrics: Hit Ratio@K, Normalized Discounted Cumulative Gain@K and Mean Reciprocal Rank@K (denoted as HR@K, NDCG@K and MRR@K respectively). HR@K measures the ability to find the target and emphasizes the accuracy of prediction. NDCG@K and MRR@K measure the ability to rank targets and emphasize the ranking of targets.

Baselines. We compare our CTGLP with the following baselines: (1) Three group link prediction methods. **LSTM-based model** (LSTM for short) [24] employs LSTM to combine the historical information of groups and outputs link probabilities. **CVAE-based model** (CVAE for short) [23] uses CVAE to reconstruct the membership of groups using a vector encoding an entire group and a vector encoding known members. **CVAEH** [23] additionally introduces a vector encoding historical information of previous groups for prediction. (2) Two neural network-based methods. **MLP** [25] utilizes mini-batch gradient descent strategy to update model parameters. **GraphSAGE** (GSAGE for short) [8] uses multi-layer aggregation functions to learn representations of nodes and obtains group vectors to make prediction. (3) Two heuristic link prediction methods. **AA** [1] utilizes correlation coefficients of overlapping neighborhoods between nodes to measure the similarity between nodes. **CN** [12] uses the number of common neighbors between nodes to measure the similarity between nodes. Nodes with higher similarity to members of a group are considered more likely to link to the group. (4) Three network embedding methods. **DeepWalk** (DW for short) [19] uses random walks to generate vectors of nodes. **node2vec** (n2v for short) [6] learns node representations using biased random walks. **HTNE** [28] integrates the Hawkes process and attention mechanism to learn the time-related representations of nodes. The individual-group link scores are obtained by aggregating the similarities between individuals.

Implementation details. For each dataset, we split it into 8:1:1 for training, validation and testing. We implement our CTGLP with PyTorch 1.6.0 and adopt the SGD as the optimizer. We apply batch normalization and dropout strategy with $p = 0.5$ for ML100K and CiaoDVD and $p = 0.1$ for ML25M. The dimension D of initial embeddings, the dimension d of hidden states and the dimension s of group vectors are all tested in {16, 32, 64, 128, 256, 512}. The batch size and learning rate are searched in {32, 64, 128, 256} and {0.005, 0.01, 0.05, 0.1} respectively. Two convolutional layers are employed in CTGNN, and the neighbor sampling sizes are empirically set to 25 and 10 respectively. For the parameters of our method, we initialize it randomly using a uniform distribution with values from 0 to 1. For baselines, we initialize the parameters according to the corresponding paper.

5.2 Performance Comparison

Performance on datasets *with* unseen nodes. From the overall results in Table 2, we observe that CTGLP outperforms most of the competing methods by a comfortable margin. On ML100K, our method obtains better performance than all baselines, especially in ranking targets. Specifically, CTGLP achieves average gains of 7.3%, 31.9% and 43.9% in terms of HR, NDCG and MRR scores. On CiaoDVD, CTGLP outperforms other baselines except for CVAE when $K = 10$. On ML25M, our method shows its great superiority in finding and ranking targets. In terms of three evaluation metrics, CTGLP obtains average gains of 11.45%, 14.7% and 11.1%, respectively.

Performance on datasets *without* unseen nodes. From the overall results in Table 3, we can see that CTGLP always achieves the best performance or the second-best one. On $\text{ML100K}_{\text{w/o}}$, our method is slightly worse than CVAE in HR scores, but it brings average gains of 22.4% and 45.9% in NDCG and MRR scores, which indicates that CTGLP can rank targets better. It is worth noting that most methods, including CTGLP, perform worse on this dataset than on ML100K. We argue that the removal of group members may affect the intrinsic nature of small dataset to a greater extent. On $\text{CiaoDVD}_{\text{w/o}}$, our method always outperforms all competing models. Specifically, in three different metrics, CTGLP outperforms the best comparative method by 10.1%, 26.6% and 36.8% on average. On $\text{ML25M}_{\text{w/o}}$, CTGLP is inferior to heuristic link prediction methods AA and CN in the ability to rank targets, but it is still satisfactory in finding targets (obtains average gain of 13.9%). Besides, task-independent embedding-based methods perform quite poorly, indicating that they are not suitable for continuous-time group link prediction. Compared to the performance on ML25M, the gains obtained by the three well-designed group link prediction methods (LSTM, CVAE and CVAEH) are much smaller than those of our method, suggesting that the lack of fine-grained temporal information and neighborhood features does limit the performance improvement of models.

Summaries. From the above analysis, several conclusions are drawn: (1) Our proposed CTGLP outperforms most baselines in datasets with and without unseen nodes, especially in finding targets. (2) Neighborhood information (the

	Method	HR@K(%)		NDCG@K(%)		MRR@K(%)	
		K=10	K=20	K=10	K=20	K=10	K=20
ML100K	LSTM	15.9	22.7	8.7	10.4	6.6	7.0
	CVAE	28.8±1.5	39.0±1.7	16.1±1.4	18.6±1.4	12.3±2.2	13.0±2.3
	CVAEH	23.7±1.9	32.2±1.0	12.3±1.1	14.5±0.8	8.9±1.3	9.5±1.3
	MLP	14.4±1.1	26.1±1.2	6.3±1.4	9.3±1.2	4.1±1.2	4.7±1.4
	GSAGE	27.1±0.7	35.6±0.4	16.2±0.4	18.3±0.4	12.9±0.6	13.5±0.5
	CTGLP	**30.5±0.9**	**42.4±0.5**	**21.5±0.9**	**24.4±0.8**	**18.6±0.7**	**19.4±0.7**
CiaoDVD	LSTM	10.6	14.7	6.0	7.0	4.5	4.8
	CVAE	**21.6±0.7**	27.1±0.8	**11.9±0.5**	13.4±0.3	**9.0±0.4**	**9.4±0.3**
	CVAEH	16.1±0.8	23.5±1.9	10.0±1.1	11.9±1.3	8.2±1.1	8.7±1.2
	MLP	15.2±1.5	20.5±2.6	7.2±0.6	8.6±0.8	4.8±0.5	5.2±0.4
	GSAGE	17.6±0.8	24.8±0.8	8.8±0.7	10.6±0.5	6.1±0.5	6.5±0.6
	CTGLP	20.8±0.6	**28.7±0.7**	11.7±0.4	**13.7±0.2**	8.8±0.7	**9.4±0.8**
ML25M	LSTM	20.5	25.3	13.9	15.1	11.9	12.2
	CVAE	22.6±2.1	26.9±2.1	16.6±1.6	17.7±1.7	14.7±1.5	15.0±1.5
	CVAEH	19.4±0.8	23.4±0.6	14.3±1.1	15.3±0.8	12.7±1.0	12.9±0.8
	MLP	19.6±1.4	23.4±3.1	14.4±1.5	15.4±2.0	12.8±1.6	13.1±1.8
	GSAGE	27.1±0.4	31.9±0.6	17.1±0.3	18.3±0.4	14.0±0.2	14.3±0.3
	CTGLP	**30.0±0.7**	**35.8±0.8**	**19.6±0.4**	**21.0±0.5**	**16.3±0.6**	**16.7±0.6**

Table 2. Performance of various methods on datasets *with* unseen nodes. Items with the highest values are marked in **bold**.

features of neighbors) is helpful to enhance the model performance. (3) Task-independent embedding-based methods are not suitable for group link prediction directly.

6 Conclusion

In this paper, to address the dynamic group link prediction problem which concentrates on the future relationships between individuals and groups in continuous-time dynamic settings, we propose a novel continuous-time group link prediction method CTGLP. We first build continuous-time interaction networks based on the historical interactions between individuals and groups and present a new graph neural network CTGNN to learn the node embeddings, where a novel aggregation function is designed to jointly capture network structural features and temporal information. Then, we provide an importance-based group modeling

	Method	HR@K(%)		NDCG@K(%)		MRR@K(%)	
		K=10	K=20	K=10	K=20	K=10	K=20
ML100K$_{w/o}$	AA	8.8	18.7	4.3	6.8	3.1	3.8
	CN	7.7	18.7	3.5	6.3	2.2	3.0
	DW	0.0	2.0	0.0	0.5	0.0	0.1
	n2v	0.0	4.0	0.0	1.0	0.0	0.3
	HTNE	0.0	8.0	0.0	2.0	0.0	0.5
	LSTM	4.7	7.0	2.0	2.6	1.2	1.3
	CVAE	**30.0±1.6**	**38.0±1.1**	17.7±1.1	19.6±1.1	13.8±1.3	14.3±1.3
	CVAEH	24.0±3.0	28.0±2.3	12.2±1.9	13.1±1.8	8.4±1.9	8.7±1.9
	CTGLP	28.0±1.0	34.0±0.6	**22.1±0.9**	**23.5±1.0**	**20.3±0.8**	**20.7±0.8**
CiaoDVD$_{w/o}$	AA	20.4	28.7	12.6	14.8	10.3	10.9
	CN	19.3	30.1	12.1	14.8	9.9	10.6
	DW	0.9	1.9	0.3	0.6	0.2	0.3
	n2v	0.5	1.0	0.2	0.4	0.1	0.2
	HTNE	0.5	2.0	0.2	0.5	0.1	0.2
	LSTM	13.0	16.6	8.2	9.1	6.7	7.0
	CVAE	22.1±1.7	26.8±1.0	12.6±1.1	13.8±0.9	9.7±0.9	10.0±0.9
	CVAEH	22.5±0.9	27.7±1.0	13.1±0.7	14.4±0.6	10.2±0.6	10.5±0.6
	CTGLP	**25.5±0.9**	**30.7±1.0**	**17.0±0.5**	**18.3±0.4**	**14.3±0.9**	**14.7±1.0**
ML25M$_{w/o}$	AA	42.5	45.1	31.8	32.4	28.5	**28.7**
	CN	42.7	45	**31.9**	**32.5**	**28.6**	**28.7**
	DW	2.1	5.4	0.7	1.5	0.3	0.5
	n2v	1.3	3.5	0.5	1.0	0.2	0.4
	HTNE	1.3	3.1	0.5	0.9	0.2	0.3
	LSTM	27.3	32.6	19.8	21.1	17.5	17.8
	CVAE	27.8±0.5	34.2±0.6	19.3±0.4	20.9±0.7	16.6±0.7	17.1±0.5
	CVAEH	27.2±0.4	32.1±0.7	19.3±0.5	20.6±0.4	16.9±0.4	17.2±0.4
	CTGLP	**45.8±1.1**	**54.4±0.9**	28.3±0.8	30.5±1.0	22.9±1.1	23.5±1.3

Table 3. Performance of various methods on datasets *without* unseen nodes. Items with the highest values are marked in **bold**.

function to model latent representations of groups, which can differentiate the contribution of members to group formation. Finally, the targets can be predicted using CTGLP based on group vectors. We compare CTGLP with ten baselines on various datasets and the experimental results show that CTGLP outperforms the state-of-the-art method. We also conduct a series of comprehensive experiments to analyze the effects of model components and hyperparameters on performance.

Acknowledgments

The work was supported by STI 2030' Major Projects (2021ZD0201300) and the Concept Grant of Hangzhou Institute of Technology of Xidian University (No.GNYZ2023XJ0409-2).

References

1. Adamic, L.A., Adar, E.: Friends and neighbors on the web. Social Networks **25**(3), 211–230 (2003)
2. Barracchia, E.P., Pio, G., Bifet, A., Gomes, H.M., Pfahringer, B., Ceci, M.: Lprobin: link prediction in dynamic networks exploiting incremental node embedding. Information Sciences **606**, 702–721 (2022)
3. Chen, J., Wang, X., Xu, X.: GC-LSTM: Graph convolution embedded LSTM for dynamic network link prediction. Applied Intelligence **52**(7), 7513–7528 (2022)
4. Ghasemian, A., Hosseinmardi, H., Galstyan, A., Airoldi, E.M., Clauset, A.: Stacking models for nearly optimal link prediction in complex networks. Proceedings of the National Academy of Sciences **117**(38), 23393–23400 (2020)
5. Goyal, P., Chhetri, S.R., Canedo, A.: dyngraph2vec: Capturing network dynamics using dynamic graph representation learning. Knowledge-Based Systems **187**, 104816 (2020)
6. Grover, A., Leskovec, J.: node2vec: Scalable feature learning for networks. In: SIGKDD 2016. pp. 855–864 (2016)
7. Guo, G., Zhang, J., Thalmann, D., Yorke-Smith, N.: Etaf: An extended trust antecedents framework for trust prediction. In: ASONAM 2014. pp. 540–547. IEEE (2014)
8. Hamilton, W., Ying, Z., Leskovec, J.: Inductive representation learning on large graphs. Advances in Neural Information Processing Systems **30** (2017)
9. Harper, F.M., Konstan, J.A.: The movielens datasets: History and context. ACM Transactions on Interactive Intelligent Systems **5**(4), 1–19 (2015)
10. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: ICLR 2017. OpenReview.net (2017), https://openreview.net/forum?id=SJU4ayYgl
11. Kumar, A., Singh, S.S., Singh, K., Biswas, B.: Link prediction techniques, applications, and performance: A survey. Physica A: Statistical Mechanics and its Applications **553**, 124289 (2020)
12. Liben-Nowell, D., Kleinberg, J.: The link-prediction problem for social networks. Journal of the American Society for Information Science and Technology **58**(7), 1019–1031 (2007)
13. Liu, G.: An ecommerce recommendation algorithm based on link prediction. Alexandria Engineering Journal **61**(1), 905–910 (2022)
14. Lü, L., Zhou, T.: Link prediction in complex networks: A survey. Physica A: statistical mechanics and its applications **390**(6), 1150–1170 (2011)
15. Lv, L., Bardou, D., Hu, P., Liu, Y., Yu, G.: Graph regularized nonnegative matrix factorization for link prediction in directed temporal networks using pagerank centrality. Chaos, Solitons and Fractals **159**, 112107 (2022)
16. Mara, A.C., Lijffijt, J., De Bie, T.: Benchmarking network embedding models for link prediction: Are we making progress? In: DSAA 2020. pp. 138–147. IEEE (2020)

17. Martínez, V., Berzal, F., Cubero, J.C.: A survey of link prediction in complex networks. ACM computing surveys (CSUR) **49**(4), 1–33 (2016)
18. Nasiri, E., Berahmand, K., Rostami, M., Dabiri, M.: A novel link prediction algorithm for protein-protein interaction networks by attributed graph embedding. Computers in Biology and Medicine **137**, 104772 (2021)
19. Perozzi, B., Al-Rfou, R., Skiena, S.: Deepwalk: Online learning of social representations. In: SIGKDD 2014. pp. 701–710 (2014)
20. Pham, T.A.N., Li, X., Cong, G., Zhang, Z.: A general graph-based model for recommendation in event-based social networks. In: ICDE 2015. pp. 567–578. IEEE (2015)
21. Qu, L., Zhu, H., Duan, Q., Shi, Y.: Continuous-time link prediction via temporal dependent graph neural network. In: WWW 2020. pp. 3026–3032 (2020)
22. Rossi, A., Barbosa, D., Firmani, D., Matinata, A., Merialdo, P.: Knowledge graph embedding for link prediction: A comparative analysis. ACM Transactions on Knowledge Discovery from Data (TKDD) **15**(2), 1–49 (2021)
23. Sha, H., Al Hasan, M., Mohler, G.: Group link prediction using conditional variational autoencoder. In: ICWSM 2021. vol. 15, pp. 656–667 (2021)
24. Stanhope, A., Sha, H., Barman, D., Hasan, M.A., Mohler, G.: Group link prediction. In: Big Data 2019. pp. 3045–3052 (2019)
25. Taud, H., Mas, J.: Multilayer perceptron (MLP). In: Geomatic Approaches for Modeling Land Change Scenarios, pp. 451–455. Springer (2018)
26. Zhou, J., Cui, G., Hu, S., Zhang, Z., Yang, C., Liu, Z., Wang, L., Li, C., Sun, M.: Graph neural networks: A review of methods and applications. AI Open **1**, 57–81 (2020)
27. Zhou, T.: Progresses and challenges in link prediction. Iscience **24**(11), 103217 (2021)
28. Zuo, Y., Liu, G., Lin, H., Guo, J., Hu, X., Wu, J.: Embedding temporal network via neighborhood formation. In: SIGKDD 2018. pp. 2857–2866 (2018)

Application of DNA-Binding Protein Prediction Based on Graph Convolutional Network and Contact Map

Zhiqiang Hui, Nan Zhou

Suzhou University of Science and Technology

Abstract: DNA contains the genetic information for the synthesis of proteins and RNA, and it is an indispensable substance in living organisms. DNA-binding proteins are an enzyme, which can bind with DNA to produce complex proteins, and play an important role in the functions of a variety of biological molecules. With the continuous development of deep learning, the introduction of deep learning into DNA-binding proteins for prediction is conducive to improving the speed and accuracy of DNA-binding protein recognition. In this study, the features and structures of proteins were used to obtain their representations through graph convolutional networks. A protein prediction model based on graph convolutional network and contact map was proposed. The method had some advantages by testing various indexes of PDB14189 and PDB2272 on the benchmark dataset.

Keywords: DNA-Binding Proteins, Graph Convolutional Network, Contact Map, Protein Prediction.

1. Introduction

With the development of gene sequencing, various sequencing studies have left many DNA and proteins, including DNA-binding proteins[1]. In order to improve the accuracy of structure and prediction, combining with the current developing trend of the technology of deep learning, a DNA binding protein prediction[2] model based on GCN[3] and contact map was proposed[4].

The protein graph depends on the sequence of the results of the comparison, so first

introducing the preprocess of the dataset, including sequence comparison and filtering; the part of the output is used to calculate the features, and the other part as the input of Pconsc4 model[5], which is used to predict protein contact map, so the inputs of the model are feature matrix and adjacency matrix[6]. We use them for training and prediction. The research content of this paper is shown in Figure 1.

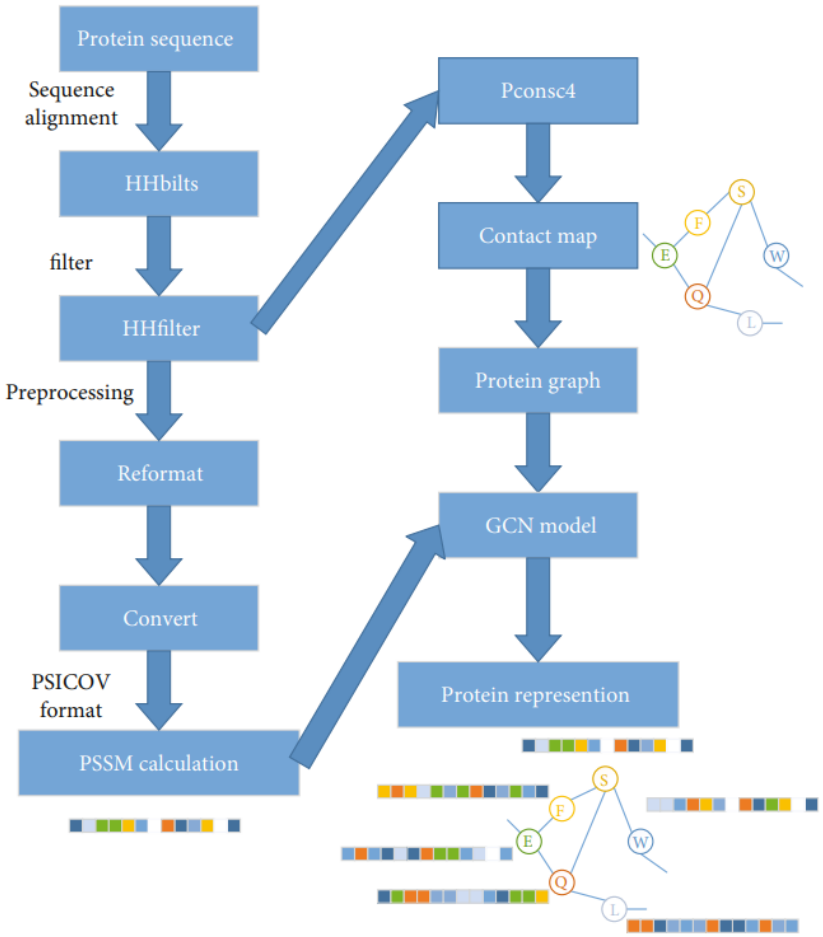


Figure 1: The processing of proteins, including the preprocessing of sequence, the generation of graph structures, and feature extraction, Pconsc4 was used to extract protein structural information. Finally, protein graph was generated higher-level feature graph through GCN.

2. Methods

We have proposed a DNA binding protein prediction model based on graph convolutional network (GCN) and contact graph. This model obtains protein features and structural representations through graph convolutional networks, and extracts protein structural information using contact maps. The specific steps include preprocessing the dataset, using the Pconsc4 model to predict protein structure information[7], extracting protein features, and training and predicting DNA binding protein data. Figure 2 shows the architecture of the model.

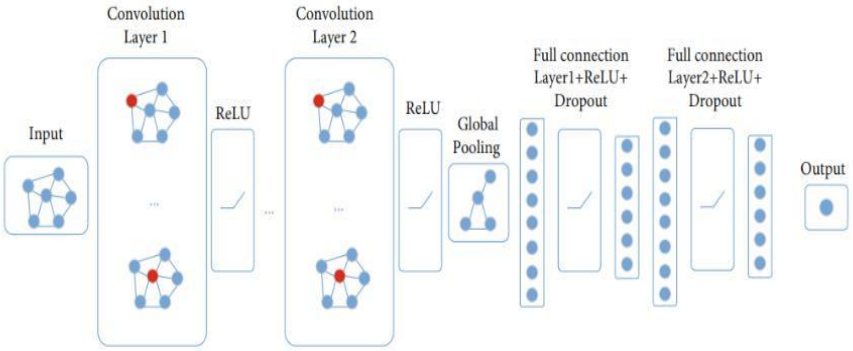


Figure 2: The structure of the GCN network, graphs of DNA-binding proteins through the GCN to get their representation.

Predicting the structure of a protein from its sequence is the purpose of introducing contact map. Specifically, assuming that the length of protein sequence is M, the size of its contact map is M∗M.M(i,j) represents the probability of contact between the ith residue and the jth residue. If the value is less than the threshold value, it can be considered that they are in contact. Pconsc4 is a fast and efficient method to predict contact map. Since its output is a probability value between 0 and 1, the threshold value of 0.5 was set for the obtained contact maps, and the probability value greater than or equal to 0.5 was set as 1.The rest were set as 0, so that the structural information of the protein could be well extracted, corresponding to the adjacency matrix as the input GCN network .[8].

Figure 3 shows a protein contact map.

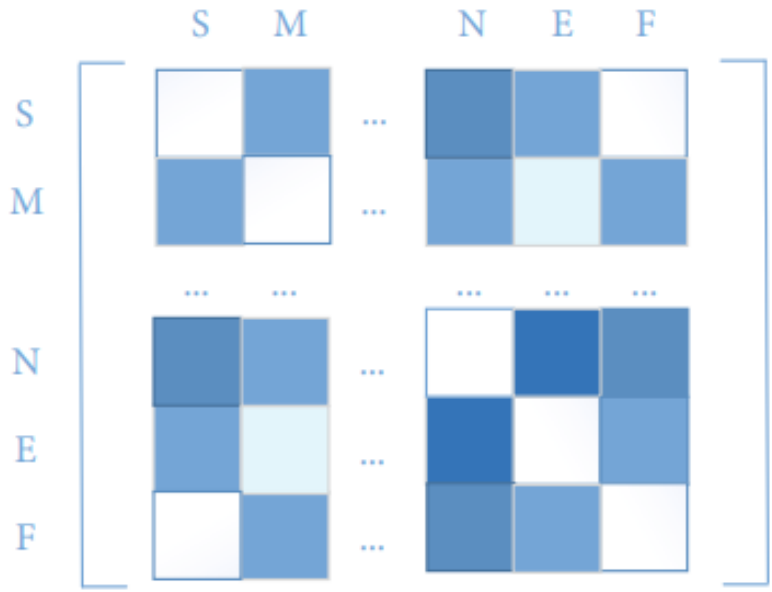


Figure 3: The contact map of protein.

The next step is the extraction of protein features. Since residues are used as nodes, the properties of residues are selected as features. Due to the differences in the R group, different features are displayed, including aromaticity, polarity, and explicit valence [9]. Position-specific scoring matrix (PSSM) is a commonly used representation of protein features, in which the results of each element depend on the results of sequence comparison, and these results represent the feature of proteins [10]. Other features were also used, such as the primary thermal coding of the remaining symbols, whether the residue was aromatic, whether the residue was acidic charged, and whether it was extremely neutral, etc. [11], as shown in Table 1. In summary, the total number of features is 54, so the protein's feature matrix dimension is (M, 54)

For PSSM, the basic position frequency matrix (PFM) [12] is calculated by the number of occurrence of residues at each position in the sequence of sequence alignment results.

Table 1: Node features.

Label	Feature	Size
1	One-hot encoding of the residue symbol	21
2	Position-specific scoring matrix (PSSM)	21
3	Whether the residue is aliphatic	1
4	Whether the residue is aromatic	1
5	Whether the residue is polar neutral	1
6	Whether the residue is acidic charged	1
7	Whether the residue is basic charged	1
8	Residue weight	1
9	The negative of the logarithm of the dissociation constant for the –COOH group	1
10	The negative of the logarithm of the dissociation constant for the –NH3 group	1
11	The negative of the logarithm of the dissociation constant for any other group in the molecule	1
12	The pH at the isoelectric point	1
13	Hydrophobicity of residue (pH = 2)	1
14	Hydrophobicity of residue (pH = 7)	1
	due (pH = 7) 1	54

3. Datasets

The DNA-binding protein dataset selected is the internationally common dataset. PDB14189 and PDB2272 were established by Gomes et al[13]. Among them, the PDB14189 dataset was divided into 7129 DNA-binding protein sequences and 7060 DNA-unbinding protein sequences, and the PDB2272 dataset was divided into 1153 DNA-binding proteins and 1119 nonbinding proteins. PDB14189 was taken as the training set and PDB2272 as the test set. The dataset is detailed in Table 2 below. Among them, positive represents DNA-binding proteins, while negative represents non-DNA-binding proteins.

Table 2: Introduction to the dataset.

Number\dataset	PDB14189	PDB2272
Positive	7129	1153
Negative	7060	1119
Total	14189	2272

4. Results

The experiment was built on PyTorch [14], an open source deep learning framework. The GCN model was based on its PyG implementation [15], PDB14189 was used for testing to find the optimal super parameters, and PDB2272 was used to test model performance.

The Evaluation Index. Accuracy (ACC), Matthews correlation coefficient (MCC), sensitivity (SN), and specificity (SP) were used as the evaluation indexes of the model [16], these indexes were widely used in the studies of biological sequences

In the independent test dataset, PDB14189 was used as the training dataset to train the model, and PDB2272 was used as the test dataset. According to the optimal experimental parameters, the final DNA-binding protein classification model was constructed: the number of GCN[17] layers were three, dropout was 0.2, PSSM was selected as the feature, the input and output dimensions of each layer were (54, 54),

(54,108), and (108,216). Other methods were compared with the method, and the method reached ACC (78.49%), SN (92.59%), SP (64.15%), and MCC (59.27%). Under certain conditions, the method has certain advantages compared with the existing methods， as shown in Table 3.

Table 3: Comparison between the proposed method and existing methods on PDB2272.

Methods	ACC (%)	MCC (%)	SN (%)	SP (%)
Qu et al.[18]	48.33	3.34	48.31	48.35
Local-DPP[19]	50.57	4.56	8.76	93.66
Pse-DNA-Pro[20]	61.88	24.30	75.28	48.08
DPP-Pse-AAC[21]	58.10	16.25	56.63	59.61
Ms-DBP[22]	66.99	33.97	70.69	63.18
GCN-method	78.49	59.27	92.59	64.15

To evaluate the impact of different dropout values, Figure 4 shows the performance of the model according to different dropout values. When the dropout is 0.2, the model has the highest performance compared to other parameters.

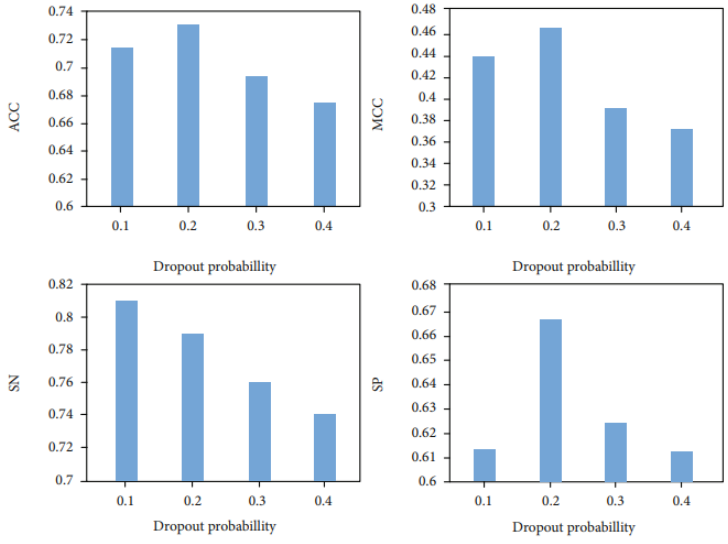


Figure 4: Comparison of prediction performance of different dropout probabilities.

5. Conclusions

DNA-binding proteins are enzymes, which can bind with DNA to produce complex proteins and play important roles in the functions of a variety of biological molecules. In order to improve the accuracy of prediction of DNA-binding protein, a DNA-binding protein prediction model based on GCN and contact map was proposed. In this model, the dataset was preprocessed by sequence alignment; then, the structural information is extracted by Pconsc4 model; PSSM and some biological characteristics are used as features. Finally, the GCN model was constructed to train and predict DNA-binding protein data. The protein graph contained information about the interactions and positions of each residue pair, which was important for feature learning and predicting binding proteins. The protein graph was input into the GCN to extract the features, and the prediction included two full connection layers. Using GCN to map proteins to the representation of rich features has also become a method of protein feature extraction. Through training and parameter tuning, the performance of GCN model was better than some existing methods. It also provides some thoughts for other fields of biological information.

In the future, we plan to carry out a research on feature extraction and network model to improve the accuracy of DNA-binding proteins and related prediction. Different biological features can be combined, and methods such as attention mechanism can be considered to improve the model, in order to achieve the goal of improving the prediction effect and other indicators.

References

1. M. S. Nogueira and O. Koch, “The development of targetspecific machine learning models as scoring functions for docking-based target prediction,” Journal of Chemical Information and Modeling, 2019.
2. Y. Wang, Y. Ding, F. Guo, L. Wei, and J. Tang, “Improved detection of DNA-binding proteins via compression technology on PSSM information,” PLoS One, vol. 12, no. 9, 2017.
3. J. Hanson, T. Litfin, K. Paliwal, and Y. Zhou, “Identifying molecular recognition features in intrinsically disordered regions of proteins by transfer learning,” Bioinformatics, vol. 36, no. 4, 2019.
4. A. S. Rifaioglu, H. Atas, M. J. Martin, R. Cetin-Atalay, V. Atalay, and T. Doğan, “Recent applications of deep learning and machine intelligence on in silico drug discovery: methods, tools and databases,” Briefings in Bioinformatics, vol. 20, no. 5, pp. 1878–1912, 2019.
5. L. Jiang, S. Wang, B. Zhang et al., “"A more probable explanation" is still impossible to explain GN-z11-flash: in response to Steinhardt et al. (arXiv:2101.12738),” 2021, https://arxiv.org/ abs/2102.01239.
6. K. Liu, X. Sun, L. Jia et al., “Chemi-net: a Molecular graph convolutional network for accurate drug property prediction,” International Journal of Molecular Sciences, vol. 20, no. 14, p. 3389, 2019.
7. S. Wang, S. Sun, Z. Li, R. Zhang, and J. Xu, “Accurate de novo prediction of protein contact map by ultra-deep learning model,” PLoS Computational Biology, vol. 13, no. 1, article e1005324, 2017.
8. V. Le, T. P. Quinn, T. Tran, and S. Venkatesh, “Deep in the bowel: highly interpretable neural encoder-decoder networks predict gut metabolites from gut microbiome,” BMC Genomics, vol. 21, no. S4, 2020.
9. Z. Hakime, Z. Arzucan, and O. Elif, “DeepDTA: deep drugtarget binding affinity prediction,” Bioinformatics, vol. 17, p. 17, 2018.
10. M. Sun, S. Zhao, C. Gilvary, O. Elemento, J. Zhou, and F. Wang, “Graph convolutional networks for computational drug development and discovery,” Briefings in Bioinformatics, vol. 21, no. 3, pp. 919–935, 2020.
11. T. Wen and R. B. Altman, “Graph convolutional neural networks for predicting drug-target interactions,” Journal of Chemical Information and Modeling, vol. 59, no. 10, pp. 4131–4149, 2019.

12. T. Nguyen, H. Le, and S. Venkatesh, “GraphDTA: prediction of drug-target binding affinity using graph convolutional networks,” BioRxiv, vol. 2019, p. 684662, 2019.
13. J. Gomes, B. Ramsundar, E. N. Feinberg, and V. S. Pande, “Atomic convolutional networks for predicting proteinligand binding affinity,” https://arxiv.org/abs/1703.10603, 2017.
14. A. Paszke, S. Gross, S. Chintala et al., Automatic differentiation in PyTorch, 2017.
15. S. Akbar, S. Khan, F. Ali, M. Hayat, M. Qasim, and S. Gul, “iHBP-DeepPSSM: identifying hormone binding proteins using PsePSSM based evolutionary features and deep learning approach,” Chemometrics and Intelligent Laboratory Systems, vol. 204, article 104103, 2020.
16. T. Song, S. Wang, D. Liu et al., “SE-OnionNet: a convolution neural network for protein–ligand binding affinity prediction,” Frontiers in Genetics, vol. 11, article 607824, 2021.
17. K. Xu, W. Hu, J. Leskovec, and S. Jegelka, “How powerful are graph neural networks?,” 2018, https://arxiv.org/abs/1810 .00826.
18. Y. Qu, J. A. Fitzgerald, H. Rauter, and N. Farrell, “Approaches to selective DNA binding in polyfunctional dinuclear platinum chemistry. The synthesis of a trifunctional compound and its interaction with the mononucleotide 5'-guanosine monophosphate,” Inorganic Chemistry, vol. 40, no. 24, pp. 6324–6327, 2001.
19. L. Wei, J. Tang, and Q. Zou, “Local-DPP: an improved DNA-binding protein prediction method by exploring local evolutionary information,” Information Sciences, vol. 384, pp. 135–144, 2017.
20. B. Liu, J. Xu, S. Fan, R. Xu, J. Zhou, and X. Wang, “PseDNApro: DNA-binding protein identification by combining Chou's PseAAC and physicochemical distance transformation,” Molecular Informatics, vol. 34, no. 1, pp. 8–17, 2015.
21. Y. D. Khan, M. Jamil, W. Hussain, N. Rasool, S. A. Khan, and K. C. Chou, “pSSbond-PseAAC: prediction of disulfide bonding sites by integration of PseAAC and statistical moments,” Journal of Theoretical Biology, vol. 463, pp. 47–55, 2019.
22. X. du, Y. Diao, H. Liu, and S. Li, “MsDBP: exploring DNAbinding proteins by integrating Multiscale sequence information via Chou's Five-Step rule,” Journal of Proteome Research, vol. 18, no. 8, pp. 3119–3132, 2019.

Identification of Membrane Protein Types Based Using Hypergraph Neural Network

Zhiqiang Hui, Meiling Qian

Suzhou University of Science and Technology

Abstract. The problem in membrane protein classification and prediction is an important topic of membrane proteomics research because the function of proteins can be quickly determined if membrane protein types can be discriminated. most current methods to classify membrane proteins are labor-intensive and require a lot of resources. In this study, the hypergraph neural network model (HGNN) was used to predict membrane protein types.

1 Methods

To address the above issues, we have proposed an innovative hypergraph neural network model (HGNN). This model constructs a multi feature fusion hypergraph correlation matrix by combining various feature extraction methods, including Average Block Method (AvBlock), Discrete Cosine Transform (DCT), Discrete Wavelet Transform (DWT), Directional Gradient Histogram (HOG), and Pseudo Position Specific Matrix (PsePSSM). Finally, by inputting these feature matrices and hypergraph correlation matrices into the HGNN model, the classification and prediction of membrane protein types were achieved. The proposed method in this paper is shown in Figure 1.

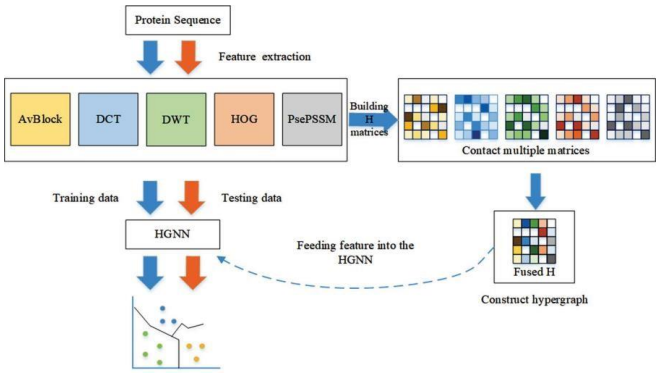


Figure 1. Schematic diagram of our proposed method.

2 Datasets

In this study, we used four datasets to test the performance of the proposed hypergraph neural network model (Table 1). Dataset 1: From Chou and Shen's research, it contains 3249 training sequences and 4333 testing sequences, totaling 8 types of membrane proteins. Dataset 2: Obtained based on Dataset 1 after removing redundant data, containing 2288 training sequences and 2306 testing sequences .Dataset 3: Expanded the dataset size to include 3073 training sequences and 3604 testing sequences .Dataset 4: Research from Chou and Elrod, containing 2059 training sequences and 2625 testing sequences.

Table 1. Statistics of different types of membrane proteins on 4 datasets.

Specific Types	Dataset 1		Dataset 2		Dataset 3		Dataset 4	
	Train	Test	Train	Test	Train	Test	Train	Test
Single-span type 1	610	444	388	223	561	245	435	478
Single-span type 2	312	78	218	39	316	7	152	180
Single-span type 3	24	6	19	6	32	9	-	-
Single-span type 4	44	12	35	10	65	17	-	-
Multi-span type 5	1,316	3,265	936	1,673	1,119	2,478	1,311	1,867
Lipid-anchor type 6	151	38	98	26	142	36	51	14
GPI-anchor type 7	182	46	122	24	164	41	110	86
Peripheral type 8	610	444	472	305	674	699	-	-
Overall	3,249	4,333	2,288	2,306	3,073	3,604	2,059	2,625

3 Results

The model proposed in this paper (HGNN) and the Memtype-2L model were compared on datasets 1, 2, and 3, respectively. The results of the test set comparison are shown in Table 7. It was found that the overall accuracy of HGNN was better than the other methods on the three datasets (92.8%, 88.6%, and 88.2%). Compared with MemType-2L (91.6%, 85.3%, 78.3%), the overall accuracy of HGNN was improved by 1.2%, 3.3%, and 9.9% (Table 2).

Table 2. Prediction accuracy of different classifiers on the dataset.

Specific Types	LR(%)	RF(%)	DNNE(%)	Our method(%)
Single-span type 1	67.6(300/444)	85.6(380/444)	92.6(411/444)	92.6(411/444)
Single-span type 2	62.8(49/78)	61.5(48/78)	76.9(60/78)	79.5(62/78)

Single-span type 3	0(0/6)	0(0/6)	0(0/6)	16.7(1/6)
Single-span type 4	66.7(8/12)	41.7(5/12)	41.7(5/12)	75.0(9/12)
Multi-span type 5	97.0(3166/3265)	92.1(3006/3265)	92.6(3024/3265)	94.8(3094/3265)
Lipid-anchor type 6	39.5(15/38)	31.6(12/38)	34.2(13/38)	44.7(17/38)
GPI-anchor type 7	8.3(36/46)	43.5(20/46)	67.4(31/46)	82.6(38/46)
Peripheral type 8	52.9(235/444)	75.2(334/444)	80.9(359/444)	87.4(388/444)
Overall	87.9(3809/4333)	87.8(3805/4333)	90.1(3903/4333)	92.8(4020/4333)

Finally, on dataset 4, the proposed method model in this paper was used to compare with other already existing method models. The comparison methods include the following: CDA, CDA and PseAA , Fourier-spectrum, PseAA, Wavelet, Dipeptide, CPSR, and Two-stage SVM. The overall accuracies of these methods on dataset 4 were 79.4%, 87.5%, 87.0%, 90.3%, 91.4%, 90.1%, 95.2%, and 96.7%, respectively. Compared with weighted SVM using PseAA (90.3%) and Two-stage SVM (96.7%), our proposed method (99.0%) was more effective, obtaining gains of 8.7% and 2.3%, respectively.

4 Conclusions

We used five methods, AvBlock, DCT, DWT, HOG, and PsePSSM, to extract the protein features. The constructed hypergraph neural network model achieved better results on different datasets. The fusion of different features, driven by multimodal data, further improved the accuracy of membrane protein identification. Therefore, HGNN has the advantages of strong scalability for multimodal features and flexibility of hyper-edge generation.

Boosting Drug-Target Binding Affinity Predictions with a Novel Three-Branch Convolutional Neural Network Approach

Yaoyao Lu [1] and Hongjie Wu [1]

[1] Suzhou University of Science and Technology
hongjie.wu@qq.com

Abstract. The process of discovering new drugs is costly and time-consuming, with safety concerns often arising. Deep learning has become a mainstream approach in computer-aided drug design, with convolutional neural networks (CNN) and graph neural networks (GNN) playing a significant role in drug-target affinity (DTA) prediction. This paper introduces a novel method for predicting DTA using a combination of graph convolutional networks and a three-branch multiscale CNN, leading to significant improvements in prediction accuracy.

Keywords: Drug-Target Binding Affinity Predictions.

1 Introduction

Proteins are involved in all aspects of cellular life activities and play a crucial role in human immunity. The ability to accurately predict drug-target binding affinity is a key focus in the discovery and repositioning of new drugs. Traditional experimental methods have evolved but are limited by being time-consuming and labor-intensive. Computer-aided drug design methods have been developed to save time and labor costs effectively.

Proteins involve all aspects of cellular life activities, and they play a vital role in human immunity [1]. Many diseases are caused by the biochemical dysfunction of protein allogeneic. Specific drugs can alter the way native proteins in the body work, resulting in the desired therapeutic effect [2]. In the discovery and repositioning of new drugs, the ability to accurately predict the drug–target binding affinity becomes the focus of research [3].While experimental methods in wet labs have evolved to screen and characterize chemical molecules, large-scale identification of potential compounds is time-consuming and labor-intensive [4].

In order to save time costs and labor costs, and to make efficient use of resources, many methods of computer-aided drug design have been developed [5]. Virtual screening is one of the main methods. It involves the prediction of potential drugs by many computational models to screen out the drug candidate ligands of interest receptor proteins from large-scale compound ligand libraries. Virtual screening can greatly reduce the number of candidate ligands, significantly reduce the experimental cycle, and thus

accelerate drug discovery [6]. Virtual screening methods can be divided into two categories: receptor-based virtual screening and ligand-based virtual screening methods. Receptor-based virtual screening mainly studies the three-dimensional structure of proteins and seeks interaction with small molecule compounds from the three-dimensional structure [7-9], so it is also called structure-based virtual screening. Common structure-based virtual screening, such as molecular docking [10, 11] and molecular dynamics simulations [12], has been extensively studied.

Although these methods are highly explanatory, their practical application is limited, because they rely heavily on the high-quality threedimensional structure of proteins, and are computationally expensive and inefficient. Ligand-based virtual screening usually starts from the ligand, analyzes the molecular structure and activity information of the known inhibitor, and summarizes the structural characteristics that have an important contribution to the binding ability of the compound by induction. This learned knowledge is then used to screen new ligands to find the compound molecules that meet the requirements [13].

Virtual screening methods are usually based on predictingdrug–target interactions or DTA. The main manifestation is that the input is a vector or graph after the drugs and proteins are encoded, and the output is a classification problem or a regression problem. However, the interactions can be understood as a series of consecutive values used to express the strength of the differentdrug–target interactions. Previously, there were quite a few research ideas that measured drug–target interactions as binary classification tasks [14-18]. In this paper, we focus on DTA prediction. In recent years, deep learning methods have shown excellent performance in many fields [19, 20], and researchers have proposed various data-driven methods based on deep learning [21-24] to study drug targeted binding [25-29].

For example, the deep learning-based DTA prediction model DeepDTA [30], uses a simplified molecular input line entry specification (SMILES) as a drug signature and a protein amino acid as a protein signature. Two features are input into two convolutional neural networks (CNNs) for extraction, and a regression module is then used for prediction through a fully connected layer. GANsDTA [31] is based on a semisupervised generative adversarial network (GAN), which consists of two parts, two GANs for feature extraction and one regression network for prediction. WideDTA [32] takes into account chemical and biological information, using deep learning from four CNNs to predict DTA. DeepAffinity [33] feeds sequences and protein structural properties with drugs into recurrent neural networks (RNNs) for learning.

Deep learning excels in the DTA prediction space [34] and has achieved many achievements. However, in deep learning models [35-37], most experiments express drugs in the form of strings, and the form of one-dimensional sequences is not the natural way molecules are expressed. When we use strings, the structure information of the numerator is lost. The use of graph convolutional networks has also been shown to be more beneficial for computational drug discovery. PADME uses molecular map convolution to predict drug-target interactions, which suggests the potential of GNN in drug development [38]. GraphDTA [39] applied the graph to small molecules to build predictive DTA models for the first time and showed good performance. Although both

CNN-based and graph neural network (GNN)-based approaches have shown good performance at

DTA predictions, there are still some problems that have not been well addressed. First of all, most deep learning methods have only a few CNN layers, and after stacking through convolutional layers, the entire feature information is compressed into a small part, but some local features of the original data are lost. Second, simply using a graph convolutional network (GCN) [9, 39] to graphically express features, does not take into account that the characteristics of each node have different effects on their adjacent and farther nodes, and the closer the node, the greater the impact. To solve the above problems, we propose a method based on the combination of GCN and CNN, putting SMILES into the GCN in the graph, considering the neighboring node weights, and using the attention mechanism based on the GCN. Global and local signatures of proteins are obtained using a three-branched multiscale convolutional neural network (MCNN) [40] at the same time, after which molecular and protein signatures are fused and fed into the prediction module. The prediction module contains three fully connected layers that finally output DTA values.

2 Method

Our approach involves constructing drug molecules into graph representation vectors and learning feature expressions through graph attention networks (GAT) and graph convolutional networks (GCN). A three-branch CNN learns the local and global features of protein sequences, and the two feature representations are merged into a regression module to predict DTA.

In this study, we used one model to deal with drug molecules and another to deal with protein data, for the regression problem of DTA prediction. First, we process the SMILES of drug molecules into graph form with RDKit [41]. The GNN starts with a GAT layer that takes the graph as input and then passes a convolutional feature matrix to the subsequent GCN layers. Each layer is activated by a rectified linear unit (ReLU) function. The final graph representation vector is then computed by concatenating the global max pooling layer and global average pooling layer output by the GCN layer. We represent proteins with amino acid sequences, encode the protein sequences and input them into the embedding layer and then into our CNN. Here, we use a three-branch CNN to extract the local and global signatures of protein amino acid sequences. The three branches use CNNs with different layers and extract different ranges of protein features, which we named the local branch, the middle branch, and the global branch, respectively. After passing through a max pooling layer, the outputs are combined as a protein representation vector. Finally, the molecular representation vector and the protein representation vector are combined and input into the regression module. We use three fully connected layers and set a dropout layer and a ReLU layer after each fully connected layer, and finally output the predicted value of DTA.

2.1 Framework

The framework includes two separate models for drug molecules and protein data. Drug SMILES strings are converted into graph form, and features are extracted using GAT and GCN layers. For proteins, amino acid sequences are encoded and input into a three-branch CNN to extract local and global features. The final molecular and protein vectors are combined and input into the regression module for DTA prediction.

2.2 Model

We evaluate our model on the Davis and KIBA datasets, which are widely used benchmarks for protein and drug binding affinity predictions.

Drugs are represented using SMILES strings converted into graph format RDKit, and proteins are encoded using a 25-tag system based on amino acid properties.

The GAT layer applies a shared linear transformation and calculates attention coefficients for each node. The GCN processes the graph structure, and a multiscale CNN extracts features from protein sequences.

We use the consistency index (CI) and mean squared error (MSE) as metrics to evaluate model performance.

We evaluate our model on two DTA datasets: Davis [42] and KIBA [43]. These two datasets are widely used as benchmark datasets for protein and drug binding affinity predictions. The Davis dataset contains data for selective analysis of kinase protein families and related inhibitors, using dissociation constant (Kd) values [44]. The KIBA dataset combines Kd, the inhibition constant (Ki) [45], or the semi-maximum inhibitory concentration (IC50) [46], using the KIBA value as an affinity. Table 1 summarizes the statistics for both datasets.

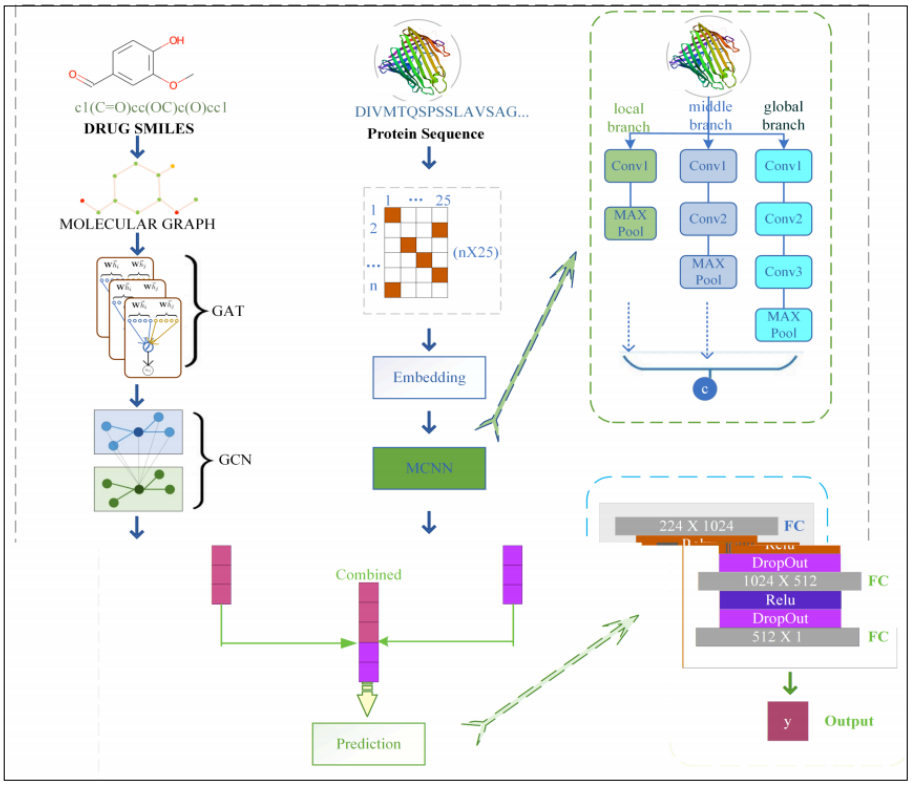
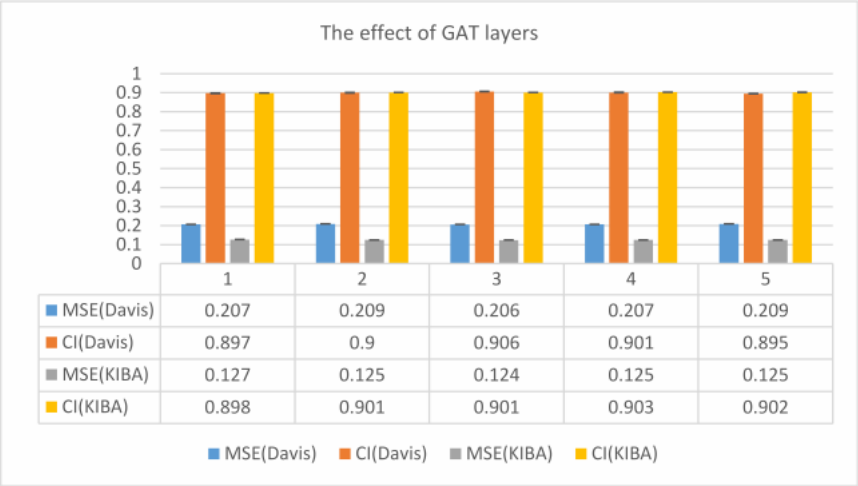


Fig. 1. Architecture.

3 Results

Our model demonstrated a 2.5% improvement in CI and a 21% increase in accuracy as measured by MSE on the Davis dataset compared to DeepDTA. It also outperformed other models including GANsDTA, WideDTA, GraphDTA, and DeepAffinity.



	1	2	3	4	5
MSE(Davis)	0.207	0.209	0.206	0.207	0.209
CI(Davis)	0.897	0.9	0.906	0.901	0.895
MSE(KIBA)	0.127	0.125	0.124	0.125	0.125
CI(KIBA)	0.898	0.901	0.901	0.903	0.902

Fig. 2. Results.

The integration of multiscale CNNs and graph representations for drug molecules and protein sequences, respectively, yielded superior results in DTA prediction. Our model offers a promising approach for accelerating drug discovery.

Acknowledgments. This research was supported by the National Natural Science Foundation of China and other institutions.

Disclosure of Interests. The authors declare no conflict of interest.

REFERENCES

[1] Cao L, Coventry B, Goreshnik I, *et al.* Design of protein-binding proteins from the target structure alone. Nature 2022; 605(7910): 551-60. http://dx.doi.org/10.1038/s41586-022-04654-9 PMID: 35332283
[2] Gonzalez MW, Kann MG. Chapter 4: Protein interactions and disease. PLOS Comput Biol 2012; 8(12): e1002819. http://dx.doi.org/10.1371/journal.pcbi.1002819 PMID: 23300410
[3] Yu JL, Dai QQ, Li GB. Deep learning in target prediction and drug repositioning: Recent advances and challenges. Drug Discov Today 2021; 1359-6446. PMID: 34718208
[4] Deng L, Zeng Y, Liu H, Liu Z, Liu X. DeepMHADTA: Prediction of drug-target binding affinity using multi-head self-attention and convolutional neural network. Curr Issues Mol Biol 2022; 44(5): 2287-99. http://dx.doi.org/10.3390/cimb44050155 PMID: 35678684
[5] Aminpour M, Montemagno C, Tuszynski JA. An overview of molecular modeling for drug discovery with specific illustrative examples of applications. Molecules 2019; 24(9): 1693. http://dx.doi.org/10.3390/molecules24091693 PMID: 31052253
[6] Scior T, Bender A, Tresadern G, *et al.* Recognizing pitfalls in virtual screening: A critical review. J Chem Inf Model 2012; 52(4): 867-81. http://dx.doi.org/10.1021/ci200528d PMID: 22435959
[7] Damale MG, Patil RB, Ansari SA, *et al.* Molecular docking, pharmacophore based virtual screening and molecular dynamics studies towards the identification of potential leads for the management of *H. pylori*. RSC Adv 2019; 9(45): 26176-208. http://dx.doi.org/10.1039/C9RA03281A PMID: 35531003

[8] Loo JSE, Emtage AL, Murali L, Lee SS, Kueh ALW, Alexander SPH. Ligand discrimination during virtual screening of the CB1 cannabinoid receptor crystal structures following cross-docking and
microsecond molecular dynamics simulations. RSC Adv 2019; 9(28): 15949-56. http://dx.doi.org/10.1039/C9RA01095E PMID: 35521393
[9] Jana S, Ganeshpurkar A, Singh SK. Multiple 3D-QSAR modeling, e-pharmacophore, molecular docking, and *in vitro* study to explore novel AChE inhibitors. RSC Adv 2018; 8(69): 39477-95. http://dx.doi.org/10.1039/C8RA08198K PMID: 35558010
[10] Stanzione F, Giangreco I, Cole JC. Use of molecular docking computational tools in drug discovery. Prog Med Chem 2021; 60: 273- 343. http://dx.doi.org/10.1016/bs.pmch.2021.01.004 PMID: 34147204
[11] Rajasekhar S, Karuppasamy R, Chanda K. Exploration of potential inhibitors for tuberculosis *via* structure-based drug design, molecular docking, and molecular dynamics simulation studies. J Comput
Chem 2021; 42(24): 1736-49. http://dx.doi.org/10.1002/jcc.26712 PMID: 34216033
[12] Salo-Ahen OMH, Alanko I, Bhadane R, *et al.* Molecular dynamics simulations in drug discovery and pharmaceutical development. Processes 2020; 9(1): 71. http://dx.doi.org/10.3390/pr9010071
[13] Singh P, Mishra M, Agarwal S, Sau S, Iyer AK, Kashaw SK. Exploring the role of water molecules in the ligand binding domain of PDE4B and PDE4D: Virtual screening based molecular docking of some active scaffolds. Curr Computeraided Drug Des 2019; 15(4): 334-66. http://dx.doi.org/10.2174/1573409914666181105153543 PMID: 30394213
[14] Lim J, Ryu S, Park K, Choe YJ, Ham J, Kim WY. Predicting drug– target interaction using a novel graph neural network with 3D structure-embedded graph representation. J Chem Inf Model 2019; 59(9): 3981-8. http://dx.doi.org/10.1021/acs.jcim.9b00387 PMID: 31443612
[15] Peng J, Wang Y, Guan J, *et al.* An end-to-end heterogeneous graph representation learning-based framework for drug–target interaction prediction. Brief Bioinform 2021; 22(5): bbaa430. http://dx.doi.org/10.1093/bib/bbaa430 PMID: 33517357
[16] Shin B, Park S, Kang K, *et al.* Self-attention based molecule representation for predicting drug-target interaction. arXiv:190806760 2019.
[17] Huang K, Xiao C, Glass LM, Sun J. MolTrans: Molecular interaction transformer for drug–target interaction prediction. Bioinformatics 2021; 37(6): 830-6. http://dx.doi.org/10.1093/bioinformatics/btaa880 PMID: 33070179
[18] Zhao T, Hu Y, Valsdottir LR, Zang T, Peng J. Identifying drug– target interactions based on graph convolutional network and deep neural network. Brief Bioinform 2021; 22(2): 2141-50. http://dx.doi.org/10.1093/bib/bbaa044 PMID: 32367110
[19] Zhang Q, He Y, Wang S, *et al.* Base-resolution prediction of transcription factor binding signals by a deep learning framework. PLoS Comput Biol 2022; 18(3): e1009941. http://dx.doi.org/10.1101/2021.11.01.466840
[20] Shen Z, Zhang Q, Han K, Huang DS. A deep learning model for RNA-protein binding preference prediction based on hierarchical LSTM and attention network. IEEE/ACM Trans Comput Biol Bioinformatics 2022; 19(2): 753-62. PMID: 32750884
[21] Yuan L, Huang DS. A network-guided association mapping approach from DNA methylation to disease. Sci Rep 2019; 9(1): 5601. http://dx.doi.org/10.1038/s41598-019-42010-6 PMID: 30944378
[22] He Y, Shen Z, Zhang Q, Wang S, Huang DS. A survey on deep learning in DNA/RNA motif mining. Brief Bioinform 2021; 22(4): bbaa229. http://dx.doi.org/10.1093/bib/bbaa229 PMID: 33005921
[23] Wang L, You ZH, Huang YA, Huang DS, Chan KCC. An efficient approach based on multi-sources information to predict circRNA – disease associations using deep convolutional neural network. Bioinformatics 2020; 36(13): 4038-46. http://dx.doi.org/10.1093/bioinformatics/btz825 PMID: 31793982

[24] Wang L, You ZH, Huang DS, Zhou F. Combining high speed ELM learning with a deep convolutional neural network feature encoding for predicting protein-RNA interactions. IEEE/ACM Trans Comput Biol Bioinformatics 2020; 17(3): 972-80. http://dx.doi.org/10.1109/TCBB.2018.2874267 PMID: 30296240
[25] Abbasi K, Razzaghi P, Poso A, Ghanbari-Ara S, Masoudi-Nejad A. Deep learning in drug target interaction prediction: current and future perspectives. Curr Med Chem 2021; 28(11): 2100-13. http://dx.doi.org/10.2174/1875533XMTA5qNzU62 PMID: 32895036
[26] Cherkasov A, Muratov EN, Fourches D, *et al.* QSAR modeling: Where have you been? Where are you going to? J Med Chem 2014; 57(12): 4977-5010. http://dx.doi.org/10.1021/jm4004285 PMID: 24351051
[27] Zhang S, Golbraikh A, Tropsha A. Development of quantitative structure-binding affinity relationship models based on novel geometrical chemical descriptors of the protein-ligand interfaces. J Med Chem 2006; 49(9): 2713-24. http://dx.doi.org/10.1021/jm050260x PMID: 16640331
[28] Politi R, Rusyn I, Tropsha A. Prediction of binding affinity and efficacy of thyroid hormone receptor ligands using QSAR and structure-based modeling methods. Toxicol Appl Pharmacol 2014; 280(1): 177-89. http://dx.doi.org/10.1016/j.taap.2014.07.009 PMID: 25058446
[29] Wang S, Jiang M, Zhang S, *et al.* Mcn-cpi: Multiscale convolutional network for compound–protein interaction prediction. Biomolecules 2021; 11(8): 1119. http://dx.doi.org/10.3390/biom11081119 PMID: 34439785
[30] Öztürk H, Özgür A, Ozkirimli E. DeepDTA: Deep drug–target binding affinity prediction. Bioinformatics 2018; 34(17): i821-9. http://dx.doi.org/10.1093/bioinformatics/bty593 PMID: 30423097
[31] Zhao L, Wang J, Pang L, Liu Y, Zhang J. GANsDTA: Predicting drug-target binding affinity using GANs. Front Genet 2020; 10: 1243. http://dx.doi.org/10.3389/fgene.2019.01243 PMID: 31993067
[32] Öztürk H, Ozkirimli E, Özgür A. WideDTA: prediction of drugtarget binding affinity. arXiv:190204166 2019.
[33] Karimi M, Wu D, Wang Z, Shen Y. DeepAffinity: Interpretable deep learning of compound–protein affinity through unified recurrent and convolutional neural networks. Bioinformatics 2019; 35(18): 3329-38. http://dx.doi.org/10.1093/bioinformatics/btz111 PMID: 30768156
[34] Mayr A, Klambauer G, Unterthiner T, *et al.* Large-scale comparison of machine learning methods for drug target prediction on ChEMBL. Chem Sci 2018; 9(24): 5441-51. http://dx.doi.org/10.1039/C8SC00148K PMID: 30155234
[35] Yi HC, You ZH, Huang DS, Li X, Jiang TH, Li LP. A deep learning framework for robust and accurate prediction of ncRNAprotein interactions using evolutionary information. Mol Ther Nucleic Acids 2018; 11: 337-44. http://dx.doi.org/10.1016/j.omtn.2018.03.001 PMID: 29858068
[36] Chuai G, Ma H, Yan J, *et al.* DeepCRISPR: optimized CRISPR guide RNA design by deep learning. Genome Biol 2018; 19(1): 80. http://dx.doi.org/10.1186/s13059-018-1459-4 PMID: 29945655
[37] Shen Z, Zhang YH, Han K, *et al.* miRNA-disease association prediction with collaborative matrix factorization. Biomolecular Networks for Complex Diseases 2017; 2017
[38] Feng Q, Dueva E, Cherkasov A, *et al.* Padme: A deep learningbased framework for drug-target interaction prediction. arXiv:180709741 2018.
[39] Nguyen T, Le H, Quinn TP, Nguyen T, Le TD, Venkatesh S. GraphDTA: Predicting drug–target binding affinity with graph neural networks. Bioinformatics 2021; 37(8): 1140-7. http://dx.doi.org/10.1093/bioinformatics/btaa921 PMID: 33119053

[40] Yang Z, Zhong W, Zhao L, Yu-Chian CC. MGraphDTA: deep multiscale graph neual network for explainable drug–target binding affinity prediction. Chem Sci 2022; 13(3): 816-33. http://dx.doi.org/10.1039/D1SC05180F PMID: 35173947
[41] Bento AP, Hersey A, Félix E, *et al.* An open source chemical structure curation pipeline using RDKit. J Cheminform 2020; 12(1): 51. http://dx.doi.org/10.1186/s13321-020-00456-1 PMID: 33431044
[42] Davis MI, Hunt JP, Herrgard S, *et al.* Comprehensive analysis of kinase inhibitor selectivity. Nat Biotechnol 2011; 29(11): 1046-51. http://dx.doi.org/10.1038/nbt.1990 PMID: 22037378

Predicting DNA-Binding Proteins through Advanced Deep Transfer Learning Techniques

Jun Yan [1] and Hongjie Wu [1]

[1] Suzhou University of Science and Technology
hongjie.wu@qq.com

Abstract. DNA-binding proteins (DBPs) are crucial in gene-related life activities. Traditional methods for DBP prediction are labor-intensive and costly. We present a novel method using deep transfer learning to predict DBPs efficiently. Our approach extracts sequence and PSSM features, employs transfer learning algorithms to construct datasets, and uses an attention mechanism-equipped neural network for prediction.

Keywords: DNA-binding proteins.

1 Introduction

DBPs play a key role in DNA replication, transcription, regulation, and other cellular processes. Traditional experimental methods for DBP identification are resource-intensive. Computational methods offer a more efficient alternative. We focus on developing a computational approach using deep transfer learning to predict DBPs accurately.

Protein is very important for the human body. Some of these proteins can interact with DNA and are called DNA-binding proteins (DBPs). These are very important for gene-related life activities. For example, in DNA replication and repair functions, origins of replication sites [1] is the location where genomic DNA replication begins, and is important for the study of the DNA replication process. In Mathematical Biosciences and Engineering Volume 19, Issue 8, 7719-7736.transcription and regulatory functions, RNA is an important molecule in the cell. Messenger RNA passes genetic information to DNA and acts as a template for protein synthesis, while only 2% of RNA molecules in proteins act as templates, the rest being a molecule called MicroRNA, which plays an important regulatory role in biological processes. Identifying molecules of MicroRNA [2] helps to understand the whole regulatory process, while some other functions are single-stranded DNA binding and separation functions, chromatin formation functions and cell development functions [3,4]. In addition, research into drug target proteins [5,6] and DNA expression genetics are also quite popular, as drug target proteins are closely related to human diseases, while DNA expression genetics include

DNA N4-methylcytosine [7,8], histone modification, RNA interference, etc. The main study in this paper is DNA binding proteins. Identification of DBPs can help us better understand how proteins interact with DNA, thus promoting the development of life science.

Although the traditional method based on biological experiments can obtain high-precision results, it needs large quantities of time and human effort. In addition, with the advent of the postgenome era, Web-lab methods cannot keep up with the growth rate of protein sequences. By contrast, computational approach reduces the resources and manpower required and enables simple and efficient identification of DBPs from many protein sequences. Thus, for the development of bioinformatics, the use of computational methods to predict DBPs is of great value.

In the past decade, machine learning based algorithms are already getting a lot of attention, and researchers have also proposed several research algorithms. In general, DNA-binding proteins can be identified by two computational methods, one based on structure and the other on sequence. Gao et al. [9] proposed a knowledge-based method called DBD-Hunter. This method uses protein structural alignment and statistical potential energy assessment to predict DBPs. Nimrod et al. [10] used the 3D structure of proteins to predict DBPs. They used a random forest classifier to determine whether a protein was a DBP based on features obtained from the protein's evolutionary profile. Zhao et al. [11] Identification of DBPs proteins using 3D structures generated based on HHblits [12]. However, structure-based approaches rely on predicted or natural 3D protein structures, and obtaining these structures is difficult. As a result, many sequence-based methods have been developed. Kumar et al. [13]developed a random forest approach called DNA-Prot to identify DBPs from protein sequences. Liu et al. [14] developed a predictor called iDNAPro-PseAAC, which relies only on protein sequence

information. They applied PseAAC [15,16] to support vector machines to identify DBPs. Wei et al. [17]used the features extracted from the local PSE-PSSM (pseudo location-specific scoring matrix) in combination with a random forest classifier and to identify DBPs. Mishra et al. [18] proposed a method called StackDPPred, which uses features extracted from PSSM and residue-specific contact energy to help train a stacking-based machine learning method that can effectively predict DNA-binding proteins.

Nanni et al. [19] in order to build an optimal and most general classification system for DNA-binding proteins, features were experimentally extracted from proteins and trained and evaluated in a separate support vector machine, while the matrix of proteins was fine-tuned using convolutional neural networks with different parameter settings, and the decisions were fused with the support vector machine using weights and rules for predicting DBPs. In recent years, deep learning has proven to be very effective in image and natural language processing. Therefore, researchers gradually began to apply deep learning in bioinformatics. Deep learning methods need only to input raw data and do not need to manually extract features, as does machine learning. For example, Qu et al. [20] used a combination of LSTM and CNN and extracted features from protein sequences to predict DBPs.Shadab et al. [21] proposed two methods, DeepDBP-ANN and DeepDBP-CNN, by using deep

2 Method

Our method combines transfer learning with deep learning. We used two transfer learning algorithms, DDC and TrAdaBoost, to expand the dataset and improve prediction

performance. An attention mechanism was integrated into the deep neural network to enhance prediction accuracy.

2.1 Transfer Learning Framework

We utilized transfer learning to extract related datasets and train our model. The framework includes sequence and PSSM feature extraction followed by deep learning model training using an attention mechanism.

We applied DDC to reduce the distribution distance between source and target domains. TrAdaBoost was used for data migration, adjusting weights during iterations to focus on misclassified samples.

An attention mechanism was added to improve model efficiency and accuracy, allowing the model to focus on critical features similarly to human focus.

We used one-hot coding for sequence representation and PSSM for evolutionary information capture.

In the experiments of this study, the main approach to prediction was to use a conjunction of transfer learning and deep learning. First, the transfer learning algorithm was used to extract the data set S, which was related to the target sample, but not completely distributed based on sample similarity.

Then the sequence and PSSM [25] features of data set S were extracted, in a deep network with an attention mechanism, the features are input and trained.

In the deep learning part of this method, the sequence and PSSM features were entered into LSTM [26] and CNN [27] respectively. In subsequent improvements, ResNet [28] was used to replace CNN, and better results were obtained. The final prediction results of these two parts also need to go through the fully connected layer. Figure 1 shows an overall prediction framework, mainly based on the DBP [29,30] prediction framework of deep transfer learning

2.2 Train

Our model was trained using the Adam optimizer in PyTorch with cross-entropy loss over 40 epochs.

We used Accuracy, MCC, Sensitivity, and Specificity as our evaluation metrics.

We used the PDB186 dataset for testing and compared our method with other existing methods. Our method showed superior performance.

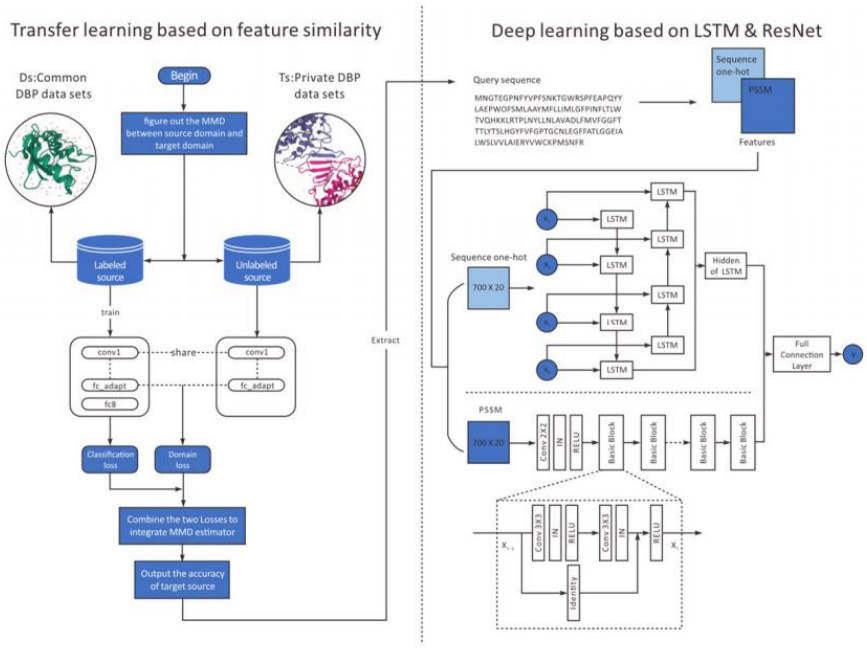


Fig. 1. Architecture.

We compared various neural network models and found that deeper models like ResNet improved performance.

Our transfer learning approach showed significant improvement in model performance with fewer labeled samples. Our deep transfer learning method outperformed traditional machine learning methods, demonstrating better prediction accuracy and robustness.

3 Results

Our deep transfer learning approach for DBP prediction is efficient and accurate. It overcomes the limitations of traditional methods by reducing the need for extensive resources. Future work will focus on improving prediction accuracy by addressing noisy samples.

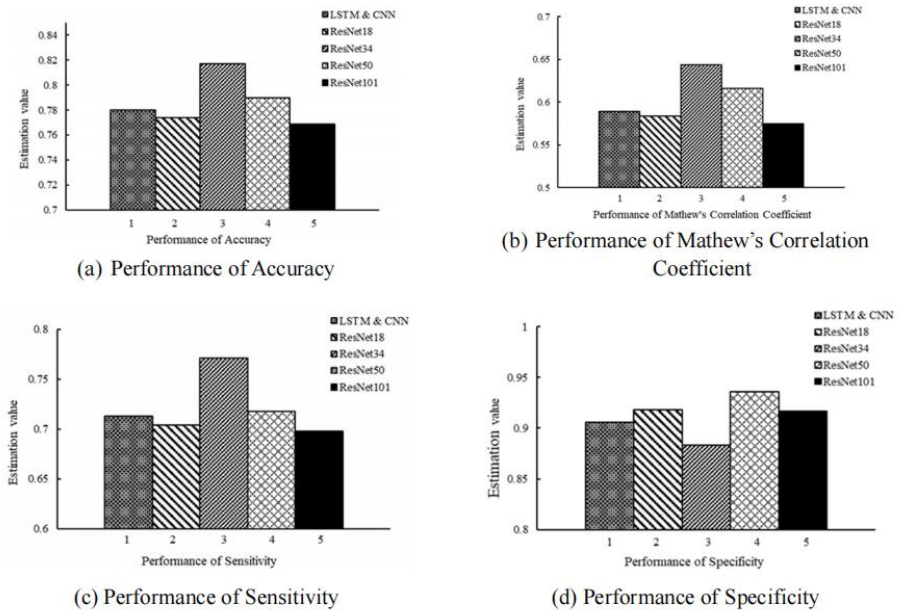


Fig. 2. Results.

The integration of multiscale CNNs and graph representations for drug molecules and protein sequences, respectively, yielded superior results in DTA prediction. Our model offers a promising approach for accelerating drug discovery.

Acknowledgments. This research was supported by the National Natural Science Foundation of China and other institutions.

Disclosure of Interests. The authors declare no conflict of interest.

References

L. Wei, W. He, A. Malik, R. Su, L. Cui, B. Manavalan, Computational prediction and interpretation of cell-specific replication origin sites from multiple eukaryotes by exploiting stacking framework, *Briefings Bioinf.*, **22** (2021). https://doi.org/10.1093/bib/bbaa275
2. L. Wei, M. Liao, Y. Gao, R. Ji, Z. He, Q. Zou, Improved and promising identification of human MicroRNAs by incorporating a high-quality negative set, *IEEE/ACM Trans. Comput. Biol. Bioinf.*,
11 (2014), 192–201. https://doi.org/10.1109/TCBB.2013.146
3. D. H. Ohlendorf, W. F. Anderson, R. G. Fisher, Y. Takeda, B.W. Matthews, The molecular basis
of DNA-protein recognition inferred from the structure of cro repressor, *Nature*, **298** (1982), 718–
23. https://doi.org/10.1038/298718a0

4. W. H. Hudson, E. A. Ortlund, The structure, function and evolution of proteins that bind DNA and
RNA, *Nat. Rev. Mol. Cell Biol.*, **15** (2014), 749–760. https://doi.org/10.1038/nrm3884
5. Y. Ding, J. Tang, F. Guo, Q. Zou, Identification of drug-target interactions via multiple kernel based triple collaborative matrix factorization, *Briefings Bioinf.*, **23** (2022), bbab582. https://doi.org/10.1093/bib/bbab582
6. Y. Ding, J. Tang, F. Guo, Identification of drug–target interactions via dual laplacian regularized
least squares with multiple kernel fusion, *Knowl.-Based Syst.*, **204** (2020), 106254. https://doi.org/10.1016/j.knosys.2020.106254
7. Y. Ding, P. Tiwari, Q. Zou, F. Guo, H. M. Pandey, C-loss based Higher-order Fuzzy Inference Systems for identifying DNA N4-methylcytosine Sites, *IEEE Trans. Fuzzy Syst.*, 2022. https://doi.org/10.1109/TFUZZ.2022.3159103
8. Y. Ding, W. He, J. Tang, Q. Zou, F. Guo, Laplacian regularized sparse representation based classifier for identifying DNA N4-methylcytosine Sites via L2,1/2-matrix norm, *IEEE/ACM Trans. Comput. Biol. Bioinf.*, 2021. https://doi.org/10.1109/TCBB.2021.3133309
9. M. Gao, J. Skolnick, DBD-Hunter: a knowledge-based method for the prediction of DNA-protein
interactions, *Nucleic Acids Res.*, **36** (2008), 3978–3992. https://doi.org/10.1093/nar/gkn332
10. G. Nimrod, M. Schushan, A. Szilagyi, C. Leslie, N. Ben-Tal, iDBPs: a web server for the identification of DNA binding proteins,
Bioinformatics,
26 (2010), 692–693.
https://doi.org/10.1093/bioinformatics/btq019
11. H. Zhao, J. Wang, Y. Zhou, Y. Yang, Predicting DNA-binding proteins and binding residues by
complex structure prediction and application to human proteome, PLoS One, (2014), e96694. https://doi.org/10.1371/journal.pone.0096694
12. M. Remmert, A. Biegert, A. Hauser, J. Soding, HHblits: lightning-fast iterative protein sequence
searching by HMM-HMM alignment, Nat. Methods, 9 (2011), 173–175. https://doi.org/10.1038/nmeth.1818
13. K. K. Kumar, G. Pugalenthi, P. N. Suganthan, DNA-Prot: identification of DNA binding proteins
from protein sequence information using random forest, J. Biomol. Struct. Dyn., 26 (2009), 679–686. https://doi.org/10.1080/07391102.2009.10507281
14. B. Liu, S. Wang, X. Wang, DNA binding protein identification by combining pseudo amino acid
composition and profile-based protein representation, Sci. Rep., 5 (2015), 15479. https://doi.org/10.1038/srep15479
15. K. C. Chou, Some remarks on protein attribute prediction and pseudo amino acid composition, J.
Theor. Biol., 273 (2011), 236–247. https://doi.org/10.1016/j.jtbi.2010.12.024
16. K. C. Chou, Prediction of protein cellular attributes using pseudo-amino acid composition, Proteins, 43 (2001), 246–255. https://doi.org/10.1002/prot.1035
17. L. Wei, J. Tang, Q. Zou, Local-DPP: an improved DNA-binding protein prediction method by
exploring local evolutionary information, Inf. Sci., 384 (2017), 135–144. https://doi.org/10.1016/j.ins.2016.06.026

18. A. Mishra, P. Pokhrel, M. T. Hoque, StackDPPred: a stacking based prediction of DNA-binding
protein from sequence, Bioinformatics, 35 (2019), 433–441.
https://doi.org/10.1093/bioinformatics/bty653
19. L. Nanni, S. Brahnam, Robust ensemble of handcrafted and learned approaches for DNA-binding
proteins, Appl. Comput. Inf., 2021. https://doi.org/10.1108/ACI-03-2021-0051
20. Y. H. Qu, H. Yu, X. J. Gong, J. H. Xu, H. S. Lee, On the prediction of DNA-binding proteins only
from primary sequences: a deep learning approach, PLoS One, (2017), e0188129.
https://doi.org/10.1371/journal.pone.0188129
21. S. Shadab, T. A. Khan, N. A. Neezi, S. Adilina, S. Shatabda, DeepDBP: deep neural networks for
identification of DNA-binding proteins, Inf. Med. Unlocked, 19 (2020), 100318.
https://doi.org/10.1016/j.imu.2020.100318
22. S. Ahmad, A. Sarai, PSSM-based prediction of DNA binding sites in proteins, BMC Bioinf., 6
(2005), 33. https://doi.org/10.1186/1471-2105-6-33
23. J. Zhang, Q. Chen, B. Liu, DeepDRBP-2L: a new genome annotation predictor for identifying DNA-binding proteins and RNA-binding proteins using convolutional neural network and long short-term memory, IEEE/ACM Trans. Comput. Biol. Bioinf., 18 (2021), 1451–1463.
https://doi.org/10.1109/TCBB.2019.2952338
24. J. Zhang, Q. Chen, B. Liu, iDRBP_MMC: identifying DNA-binding proteins and RNA-binding
proteins based on multi-label learning model and motif-based convolutional neural network, J. Mol. Biol., 432 (2020), 5860–5875. https://doi.org/10.1016/j.jmb.2020.09.008
25. G. Li, X. Du, X. Li, L. Zou, G. Zhang, Z. Wu, Prediction of DNA binding proteins using local
features and long-term dependencies with primary sequences based on deep learning, PeerJ, 9 (2021), e11262. https://doi.org/10.7717/peerj.11262
26. K. Greff, R. K. Srivastava, J. Koutnik, B. R. Steunebrink, J. Schmidhuber, LSTM: a search space
odyssey, IEEE Trans. Neural Networks Learn. Syst., 28 (2017), 2222–2232.
https://doi.org/10.1109/TNNLS.2016.2582924
27. T. Roska, L. O. Chua, The CNN universal machine: an analogic array computer, IEEE Trans. Circuits Syst. II, 40 (1993), 163–173. https://doi.org/10.1109/82.222815
28. C. Szegedy, S. Ioffe, V. Vanhoucke, A. A. Alemi, Inception-v4, inception-resnet and the impact of residual connections on learning, in Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, (2017), 4278–4284. Available from: https://dl.acm.org/doi/10.5555/3298023.3298188.
29. B. Liu, J. Xu, X. Lan, R. Xu, J. Zhou, X. Wang, et al., iDNA-Prot|dis: identifying DNA-binding proteins by incorporating amino acid distance-pairs and reduced alphabet profile into the general pseudo amino acid composition, PLoS One, (2014), e106691. https://doi.org/10.1371/journal.pone.0106691
30. Y. Wang, Y. Ding, F. Guo, L. Wei, J. Tang, Improved detection of DNA-binding proteins via
compression technology on PSSM information, PLoS One, (2017), e0185587.
https://doi.org/10.1371/journal.pone.0185587

31. R. Caruana, A. Niculescu-Mizil, An empirical comparison of supervised learning algorithms, in
Proceedings of the 23rd International Conference on Machine Learning, (2006), 161–168.
https://doi.org/10.1145/1143844.1143865
32. K. Weiss, T. M. Khoshgoftaar, D. Wang, A survey of transfer learning, J. Big Data, 3 (2016), 9.
https://doi.org/10.1186/s40537-016-0043-6
33. S.J. Pan, Q. Yang, A survey on transfer learning, IEEE Trans. Knowl. Data Eng., 22 (2010), 1345–
1359. https://doi.org/10.1109/TKDE.2009.191
34. M. Oquab, L. Bottou, I. Laptev, J. Sivic, Learning and transferring mid-level image representations using convolutional neural networks, in 2014 IEEE Conference on Computer Vision and Pattern Recognition, (2014), 1717–1724. https://doi.org/10.1109/CVPR.2014.222
35. W. Dai, Q. Yang, G. Xue, Y. Yu, Boosting for transfer learning, Machine Learning, in Proceedings
of the 24th International Conference on Machine Learning, (2007), 193–200.
https://doi.org/10.1145/1273496.1273521
36. S. R. Bowman, L. Vilnis, O. Vinyals, A. M. Dai, R. Jozefowicz, S. Bengio, Generating sentences
from a continuous space, in Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning, (2016), 10–21. https://doi.org/10.18653/v1/K16-1002
37. E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, T. Darrell, Deep domain confusion: Maximizing for
domain invariance, preprint, arXiv:1412.3474.
38. H. Yan, Y. Ding, P. Li, Q. Wang, Y. Xu, W. Zuo, Mind the class weight bias: weighted maximum mean discrepancy for unsupervised domain adaptation, in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), (2017), 945–954. https://doi.org/10.1109/CVPR.2017.107
39. W. Qin, X. Cui, C. A. Yuan, X. Qin, L. Shang, Z. K. Huang, et al., Flower species recognition system combining object detection and attention mechanism, in International Conference on Intelligent Computing, Springer, 2019. https://doi.org/10.1007/978-3-030-26766-7_1
40. K. Cho, B. V. Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, et al., Learning
phrase representations using RNN encoder-decoder for statistical machine translation, in Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), (2014), 1724–1734. https://doi.org/10.3115/v1/D14-1179
41. T. Mikolov, S. Kombrink, L. Burget, J. Černocký, S. Khudanpur, Extensions of recurrent neural
network language model, in 2011 IEEE International Conference on Acoustics, Speech and Signal
Processing (ICASSP), (2011), 5528–5531. https://doi.org/10.1109/ICASSP.2011.5947611
42. L. Wei, C. Zhou, H. Chen, J. Song, R. Su, ACPred-FL: a sequence-based predictor using effective
feature representation to improve the prediction of anti-cancer peptides, Bioinformatics, 34 (2018),
4007–4016. https://doi.org/10.1093/bioinformatics/bty451
43. Y. Ding, J. Tang, F. Guo, Protein crystallization identification via fuzzy model on linear neighborhood representation, IEEE/ACM Trans. Comput. Biol. Bioinf., 18 (2021), 1986–1995. https://doi.org/10.1109/TCBB.2019.2954826

44. Y. Ding, J. Tang, F. Guo, Human protein subcellular localization identification via fuzzy model
on kernelized neighborhood representation, Appl. Soft Comput., 96 (2020), 106596.
https://doi.org/10.1016/j.asoc.2020.106596
45. S. K. Knapp, Accelerate FPGA macros with one-hot approach, Electron. Des., 1990.
46. J. Soding, Protein homology detection by HMM-HMM comparison, Bioinformatics, 21 (2005),
951–960. https://doi.org/10.1093/bioinformatics/bti125
47. K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), (2016), 770–778.
https://doi.org/10.1109/CVPR.2016.90
48. V. Nair, G. E. Hinton, Rectified linear units improve restricted boltzmann machines, in Proceedings of the 27th International Conference on International Conference on Machine Learning, (2010), 807–814. Available from: https://dl.acm.org/doi/10.5555/3104322.3104425.
49. A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, et al., Automatic differentiation
in pytorch, 2017. Available from: https://paperswithcode.com/paper/automatic-differentiation-inpytorch.
50. D. P. Kingma, J. Ba, Adam: a method for stochastic optimization, CoRR, 2015. Available from:
https://www.semanticscholar.org/paper/Adam%3A-A-Method-for-Stochastic-OptimizationK-ingma-Ba/a6cb366736791bcccc5c8639de5a8f9636bf87e8.
51. W. Lou, X. Wang, F. Chen, Y. Chen, B. Jiang, H. Zhang, Sequence based prediction of DNAbinding proteins based on hybrid feature selection using random forest and Gaussian naive Bayes,
PLoS One, (2014), e86703. https://doi.org/10.1371/journal.pone.0086703
52. P. W. Rose, A. Prlic, C. Bi, W. F. Bluhm, C. H. Christie, S. Dutta, et al., The RCSB Protein Data
Bank: views of structural biology for basic and applied research and education, Nucleic Acids Res.,
43 (2015), D345–D356. https://doi.org/10.1093/nar/gku1214
53. X. Du, Y. Diao, H. Liu, S. Li, MsDBP: Exploring DNA-binding proteins by integrating multiscale
sequence information via Chou's five-step rule, J. Proteome Res., 18 (2019), 3119–3132.
https://doi.org/10.1021/acs.jproteome.9b00226

Leveraging Local Protein Structures for Enhanced Drug-Target Binding Affinity Predictions Using Deep Learning Techniques

Runhua Zhang [1] and Hongjie Wu [1]

[1] Suzhou University of Science and Technology
hongjie.wu@qq.com

Abstract. The traditional drug discovery process is both time-consuming and costly. Utilizing artificial intelligence to predict drug-target binding affinity (DTA) has become a crucial approach for accelerating new drug discovery. This study introduces a novel deep learning-based method that incorporates both the primary and secondary structures of proteins to better represent the local and global features of proteins. We employ convolutional neural networks (CNNs) and graph neural networks (GNNs) to model proteins and drugs separately, capturing their interactions more effectively. Our method demonstrated improved performance in predicting DTA compared to state-of-the-art methods on two benchmark datasets.

Keywords: Drug-Target Binding Affinity Prediction.

1 Introduction

Developing a new drug that reaches the market costs approximately $2.6 billion, with a low approval rate of less than 12%. Therefore, computer-aided drug development has become a hot research topic. Accurately identifying drug-target interactions is essential in the computational stages of drug development. Our method focuses on predicting DTA by representing proteins using both their primary and secondary structures.

Developing a new drug that can be brought to market costs approximately $2.6 billion, and the approval rate of new drugs that enter clinical trials is less than 12% [1,2]. Moreover, developing a new drug requires a significant amount of time [3]. Therefore, computer-aided drug development has become a hot research topic in recent years [4]. Accurately identifying drug-target interactions is an essential step in the computational stages of drug development [5]. Currently, there are mainly two categories of computational methods used for predicting drug-target interactions. The first type treats interaction prediction as a binary classification task [6], that is, determining whether a drug and a target interact or not. The other type treats it as a regression task for predicting the binding affinity between the drug and the target. Binding affinity can measure the strength of drug-target interactions, and is usually expressed using inhibition constant (Ki), dissociation constant (Kd), or the half maximal inhibitory concentration (IC50) [7].

Our method focuses mainly on predicting drug-target binding affinity (DTA).

There are several computational methods used for predicting DTA. One approach is the ligand-based method which compares a query ligand to known ligands based on its target protein. However, if the number of known ligands for the target protein is insufficient [8], the predictions may be unreliable [9]. Another approach is molecular docking [10], which models the binding of compounds and proteins in conformational space based on their 3D structures. However, preparing 3D protein-ligand complexes can be quite challenging [11].

Predicting DTA using computational methods typically involves three main steps.

First, drug and target protein data are converted into computationally ready vectors or graphs using various encoding methods [12]. The commonly used representation forms of drugs mainly include simplified molecular linear input specification (SMILES) [13], molecular fingerprint and graph. Proteins are usually represented using one-hot encoding to capture their primary sequences. Second, different feature extraction methods are applied to obtain representative features of drugs and proteins, which are then used to replace their original input features. Finally, a regression process is performed to combine the respective representations and predict binding affinities.

In recent years, deep learning (DL) has made significant progress in the field of computer-aided drug design [14], particularly in the prediction of DTA. Many DLbased methods have been developed to improve DTA prediction performance. One of the earliest DL-based DTA prediction models, DeepDTA [15] uses one-dimensional (1D) convolutional neural networks (CNN) to extract sequence features of drugs and proteins, it uses the protein primary sequence and the SMILES string of the drug ligand as input, without incorporating any additional input information. WideDTA [16] improves prediction performance by incorporating protein domain information. However, expressing drugs as SMILES strings leads to the loss of their original graph structure, motivating the use of graph neural networks (GNN). GraphDTA [17] represents drugs as graphs, using multiple GNN variants such as the graph convolutional network (GCN)[18], the graph attention network (GAT) [19], and the graph isomorphism network (GIN) [20], and retaining CNN to represent proteins. This model outperformed existing 1D methods, highlighting the importance of structural information. However, these models only consider the overall interaction between drugs and proteins. MGraphDTA[21] introduces dense connections into the GNN and builds an ultra-deep network structure consisting of 27 layers of GCN. This architecture enables the simultaneous capture of local and global structures of compounds, improving the prediction performance of DTA. Additionally, MGraphDTA proposes a new visualization method to better understand the role of GNN in DTA prediction. DeepAffinity [22] introduces an attention mechanism to learn the binding site information between compounds and proteins, improving model interpretability. These approaches have demonstrated the success of using CNNs for feature extraction from protein sequences. GraphDTA, on the other hand, uses a graph structure to represent drugs and applies GCN for feature extraction, leading to improved prediction performance. This indicates that graph structures can be effectively utilized in DTA prediction. The above method mainly uses the primary structure of the protein, that is, the amino acid sequence to represent and input,

and can only extract the global features of the protein, ignoring the local features of the protein in a segment.

In this paper, we propose a novel deep learning-based method for predicting DTA that integrates both global and local features of proteins. The entire model comprises three distinct modules: the global protein features module, the local protein featuresmodule, and the ligand module. The protein data is one-dimensional and consists of the amino acid sequence structure and secondary structure of the protein, while the drug ligand is represented using graph data. We use CNN to learn the representation of protein primary and secondary sequences, employ GAT and GCN to learn the graph data representation of drugs, and finally concatenate the features obtained from the convolution and maximal pooling layers of the three modules and fed them into the classification component.

2 Method

We evaluated our model on two public datasets: Davis and KIBA. Protein secondary structure information was predicted using MLRC methods and incorporated into our model. Primary protein sequences were represented using one-hot encoding, and secondary structures were represented using 8D one-hot vectors. Drugs were represented as graphs using SMILES strings converted with RDKit.

Our model predicts drug-target interactions as a regression task, aiming to predict specific binding affinities. The proposed model architecture consists of three functional modules: global protein features, local protein features, and ligand features.

2.1 Global Protein Features Module

This module uses a CNN to learn the representation of protein primary sequences. The primary sequence is represented as a one-dimensional sequence of amino acids using a 20D one-hot encoding scheme. The CNN consists of three convolutional layers with an increasing number of filters, followed by max pooling layers. This allows the model to capture global features of the protein sequence.

Local Protein Features Module. This module also uses a CNN, but it focuses on learning the representation of protein secondary structures. Secondary structures are represented using an 8D one-hot vector for each amino acid type. The CNN here also consists of three convolutional layers followed by max pooling layers, capturing the local features of the protein.

Ligand Features Module. Drug compounds are represented as graphs with nodes representing atoms and edges representing chemical bonds. We use the Graph Attention Network (GAT) and Graph Convolutional Network (GCN) to learn the graph data representation of drugs. The GAT layer learns the importance of each node using a self-attention mechanism, while the GCN layers capture the connectivity relationship between graph nodes.

Classification Component. The features from the max pooling layers of the three modules are concatenated and fed into a classification component. This component

consists of three fully connected layers with 1024, 512, and 512 nodes, respectively. Dropout layers with a rate of 0.2 are used after the first two fully connected layers to prevent overfitting. The output layer predicts the binding affinity.

2.2 Train

The model was trained for 1000 epochs with a batch size of 512 and a learning rate of 0.0005. The Adam optimization algorithm and Rectified Linear Unit (ReLU) activation function were used to train the network. Mean Squared Error (MSE) was used as the loss function.

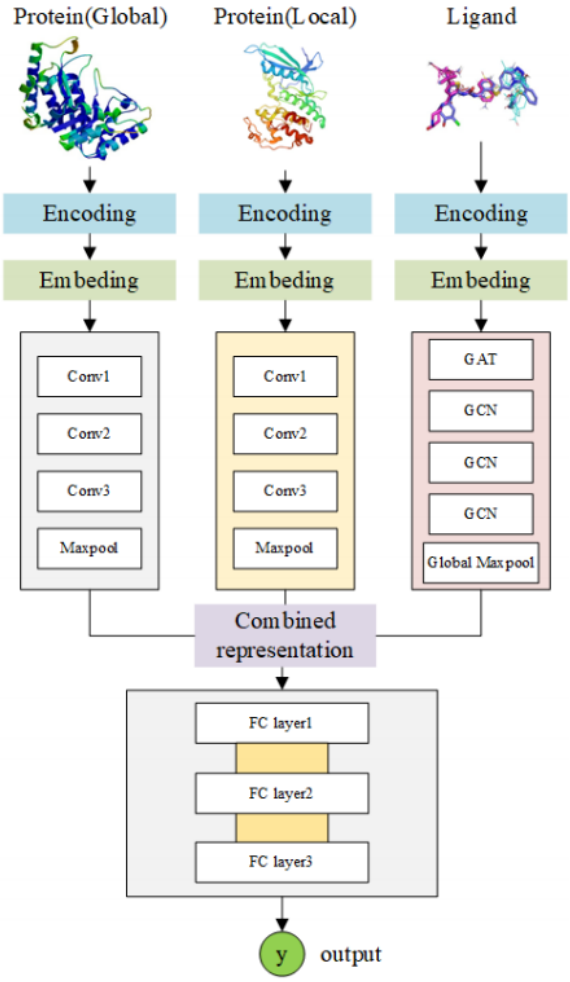


Fig. 1. Architecture.

3 Results

Our model showed superior performance compared to DeepDTA, WideDTA, GraphDTA, and AttentionDTA models on both the Davis and KIBA datasets. The inclusion of protein local features improved prediction accuracy, as indicated by the lower MSE and higher CI scores.

Our deep learning model, which incorporates primary and secondary protein structures, predicts drug-target binding affinity more accurately. The enriched datasets can be used in future experiments.

Acknowledgments. This research was supported by the National Natural Science Foundation of China and other institutions.

Disclosure of Interests. The authors declare no conflict of interest.

References

1. DiMasi J A, Grabowski H G, Hansen R W. Innovation in the pharmaceutical industry: new estimates of R&D costs[J]. Journal of health economics, 2016, 47: 20-33.
2. Mullard A. New drugs cost US $2.6 billion to develop[J]. Nature reviews. Drug discovery, 2014, 13(12): 877.
3. Ding Y, Tang J, Guo F. Identification of drug–target interactions via dual laplacian regular
ized least squares with multiple kernel fusion[J]. Knowledge-Based Systems, 2020, 204: 106254.
4. Sun M, Tiwari P, Qian Y, et al. MLapSVM-LBS: Predicting DNA-binding proteins via a multiple Laplacian regularized support vector machine with local behavior similarity[J]. Knowledge-Based Systems, 2022, 250: 109174.9
5. Ding Y, Tang J, Guo F. Identification of drug–target interactions via fuzzy bipartite local model[J]. Neural Computing and Applications, 2020, 32: 10303-10319.
6. Yamanishi Y, Kotera M, Kanehisa M, et al. Drug-target interaction prediction from chemical, genomic and pharmacological data in an integrated framework[J]. Bioinformatics, 2010, 26(12): i246-i254.
7. Tang J, Szwajda A, Shakyawar S, et al. Making sense of large-scale kinase inhibitor bioactivity data sets: a comparative and integrative analysis[J]. Journal of Chemical Information and Modeling, 2014, 54(3): 735-743.
8. Yang H, Ding Y, Tang J, et al. Drug–disease associations prediction via multiple kernelbased dual graph regularized least squares[J]. Applied Soft Computing, 2021, 112: 107811.
9. Ding Y, Tang J, Guo F. Human protein subcellular localization identification via fuzzy model on kernelized neighborhood representation[J]. Applied Soft Computing, 2020, 96: 106596.
10. Wu H, Ling H, Gao L, et al. Empirical potential energy function toward ab initio folding G protein-coupled receptors[J]. IEEE/ACM Transactions on Computational Biology and Bioinformatics, 2020, 18(5): 1752-1762.
11. Karimi M, Wu D, Wang Z, et al. Explainable deep relational networks for predicting compound–protein affinities and contacts[J]. Journal of chemical information and modeling, 2020, 61(1): 46-66.

12. Ding Y, Tang J, Guo F. Identification of drug-target interactions via multi-view graph regularized link propagation model[J]. Neurocomputing, 2021, 461: 618-631.
13. Weininger D. SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules[J]. Journal of chemical information and computer sciences, 1988, 28(1): 31-36.
14. Ding Y, Tang J, Guo F. Identification of drug-side effect association via semisupervised model and multiple kernel learning[J]. IEEE journal of biomedical and health informatics, 2018, 23(6): 2619-2632.
15. Öztürk H, Özgür A, Ozkirimli E. DeepDTA: deep drug–target binding affinity prediction[J]. Bioinformatics, 2018, 34(17): i821-i829.
16. Öztürk H, Ozkirimli E, Özgür A. WideDTA: prediction of drug-target binding affinity[J]. arXiv preprint arXiv:1902.04166, 2019.
17. Nguyen T, Le H, Quinn T P, et al. GraphDTA: predicting drug–target binding affinity with graph neural networks[J]. Bioinformatics, 2021, 37(8): 1140-1147.
18. Kipf T N, Welling M. Semi-supervised classification with graph convolutional networks[J]. arXiv preprint arXiv:1609.02907, 2016.
19. Veličković P, Cucurull G, Casanova A, et al. Graph attention networks[J]. arXiv preprint arXiv:1710.10903, 2017.
20. Xu K, Hu W, Leskovec J, et al. How powerful are graph neural networks?[J]. arXiv preprint arXiv:1810.00826, 2018.
21. Yang Z, Zhong W, Zhao L, et al. Mgraphdta: deep multiscale graph neural network for explainable drug–target binding affinity prediction[J]. Chemical science, 2022, 13(3): 816- 833.
22. Karimi M, Wu D, Wang Z, et al. DeepAffinity: interpretable deep learning of compound– protein affinity through unified recurrent and convolutional neural networks[J]. Bioinformatics, 2019, 35(18): 3329-3338.
23. Davis M I, Hunt J P, Herrgard S, et al. Comprehensive analysis of kinase inhibitor selectivity[J]. Nature biotechnology, 2011, 29(11): 1046-1051.10
24. Tang J, Szwajda A, Shakyawar S, et al. Making sense of large-scale kinase inhibitor bioactivity data sets: a comparative and integrative analysis[J]. Journal of Chemical Information and Modeling, 2014, 54(3): 735-743.
25. Guermeur, Yann, et al. "Improved performance in protein secondary structure prediction by inhomogeneous score combination." Bioinformatics (Oxford, England) 15.5 (1999): 413- 421.
26. Combet, Christophe, et al. "NPS@: network protein sequence analysis." Trends in biochemical sciences 25.3 (2000): 147-150.
27. Wang H, Tang J, Ding Y, et al. Exploring associations of non-coding RNAs in human diseases via three-matrix factorization with hypergraph-regular terms on center kernel alignment[J]. Briefings in Bioinformatics, 2021, 22(5): bbaa409.
28. Kabsch W, Sander C. Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features[J]. Biopolymers: Original Research on Biomolecules, 1983, 22(12): 2577-2637.
29. Wan L, Zeiler M, Zhang S, et al. Regularization of neural networks using dropcon nect[C]//International conference on machine learning. PMLR, 2013: 1058-1066.
30. Kingma D P, Ba J. Adam: A method for stochastic optimization[J]. arXiv preprint arXiv:1412.6980, 2014.

31. Nair V, Hinton G E. Rectified linear units improve restricted boltzmann machines[C]//Proceedings of the 27th international conference on machine learning (ICML-10). 2010: 807- 814.
32. Zhao Q, Xiao F, Yang M, et al. AttentionDTA: prediction of drug–target binding affinity using attention model[C]//2019 IEEE international conference on bioinformatics and biomedicine (BIBM). IEEE, 2019: 64-69.

Advancing Identification of DNA-Protein Binding Residues Using Deep Learning Techniques

Haipeng Zhao [1] and Hongjie Wu [1]

[1] Suzhou University of Science and Technology
hongjie.wu@qq.com

Abstract. Accurate identification of DNA-protein binding sites is vital for understanding biological processes and facilitating drug discovery. This study introduces a novel method that integrates a Transformer encoder with Bi-directional Long Short-Term Memory (BiLSTM) to predict DNA-protein binding residues effectively. The method enriches protein representation by combining evolutionary information from the position-specific scoring matrix (PSSM) with spatial information from predicted secondary structures. Experimental results demonstrate the method's competitiveness, achieving an MCC of 0.349, SP of 96.50%, SN of 44.03%, and ACC of 94.59% on the PDNA-41 dataset.

Keywords: DNA-Protein Binding.

1 Introduction

DNA-protein interactions are critical for biological processes like transcription and DNA repair. Identifying binding sites is essential for understanding gene regulation and disease mechanisms and for drug design. Traditional experimental methods are costly and time-consuming. Computational methods offer a more efficient alternative.

Given the importance of protein-DNA binding, many wet-lab methods have been developed to identify protein-DNA binding residues. These methods include X-ray crystallography [6], Fast ChIP [7], and electrophoretic mobility shift assays (EMSAs) [8,9]. Although wet-lab methods can yield precise identification outcomes, they are expensive and labor intensive. Moreover, they cannot keep up with the growth rate of protein sequences in the post-genomic era [10]. Therefore, there is a need to develop an efficient and convenient computation-based method for identifying protein-DNA binding residues. With advancements in computer theory, a number of computational methods have emerged for this purpose. These methods can be broadly categorized into three types: sequence-based, structure-based, and hybrid methods [11].

Bioinformatics research primarily focuses on sequence-based methods, which pose a significant challenge. Predicting protein-DNA binding residues using only sequence-based features may have poor performance due to the limited information contained in protein sequences. However, the number of protein sequences is increasing day by day, research in this area is still focused on utilizing sequence features. In the past decade, several sequence-based methods have been proposed. These include BindN [12],

ProteDNA [13], DP-Bind [14], BindN+ [15], MetaDBSite [16], TargetDNA [17], DNABind [18], DNAPred [19] and PredDBR [20], among others. In BindN, they utilized three types of protein sequence features: hydrophobicity, side chain pKa value, and molecular mass of amino acids. These features were inputted into a support vector machine (SVM) to accurately predict protein-DNA binding residues. In DP-Bind, they utilized evolutionary information obtained from protein sequences, specifically the position-specific scoring matrix (PSSM) [21]. To enhance the recognition accuracy of protein-DNA binding residues, three conventional machine learning techniques were combined: penalized logistic regression, SVM, and kernel logistic regression. In TargetDNA, they used two protein sequence features, solvent accessibility and evolutionary information, and made use of an undersampling technique to divide the raw data into multiple sub-datasets and applied multiple SVMs for ensemble learning to predict protein-DNA binding residues.

Structure-based methods utilize either natural or predicted 3D structure information of proteins. This is because the 3D structure of a protein contains a large amount of information and the structure of a protein determines the function of the protein to some extent. Consequently, utilizing protein structure information for predicting protein-DNA binding residues often yields better performance than sequencebased methods. Common structure-based methods include: DBD-Hunter [22], DNABINDPROT [23], DR_bind [24], PreDs [25], etc. All these methods mentioned above use only the structure information of the protein and ignore the information that may be contained in the protein sequence that may be helpful in predicting the protein-DNA binding residues. To enhance prediction accuracy, hybrid methods integrate both sequence and structure information. Some common hybrid methods include: TargetATP [26], COACH [27], TargetS [28], SVMPred [29] and NsitePred [30], etc. In DR_bind, the model predicts protein-DNA binding residues by utilizing evolutionary, geometric and electrostatic properties to describe the protein structure. In COACH, they designed an algorithm named TM-SITE to infer binding sites from homologous structural templates and also an algorithm named S-SITE for sequence.

2 Method

The study uses the PDNA-543 and PDNA-41 datasets, enriching protein features by combining PSSM evolutionary information with secondary structure predictions. The model architecture includes a Transformer encoder, BiLSTM, and a convolutional feature extraction module, followed by a multilayer perceptron (MLP) decoder for residue classification.

PSSM features were generated using PSI-BLAST, and secondary structure predictions were made using PSIPRED. These features were combined to form a comprehensive protein representation.

The model integrates a Transformer encoder and BiLSTM to capture long-range dependencies and local residue features. A convolutional layer processes the encoded protein feature matrix, and an MLP decoder generates the binding pattern.

2.1 Framework

The PDNA-543 and PDNA-41 datasets were utilized, with the former used for training and the latter for testing the model's generalization performance.

2.2 Train

The model was evaluated using the DUD-E dataset and the Human dataset, which are standard benchmarks for DTI prediction.

The model was trained using binary cross-entropy loss and the Adam optimizer. Evaluation metrics included MCC, SP, SN, and ACC.

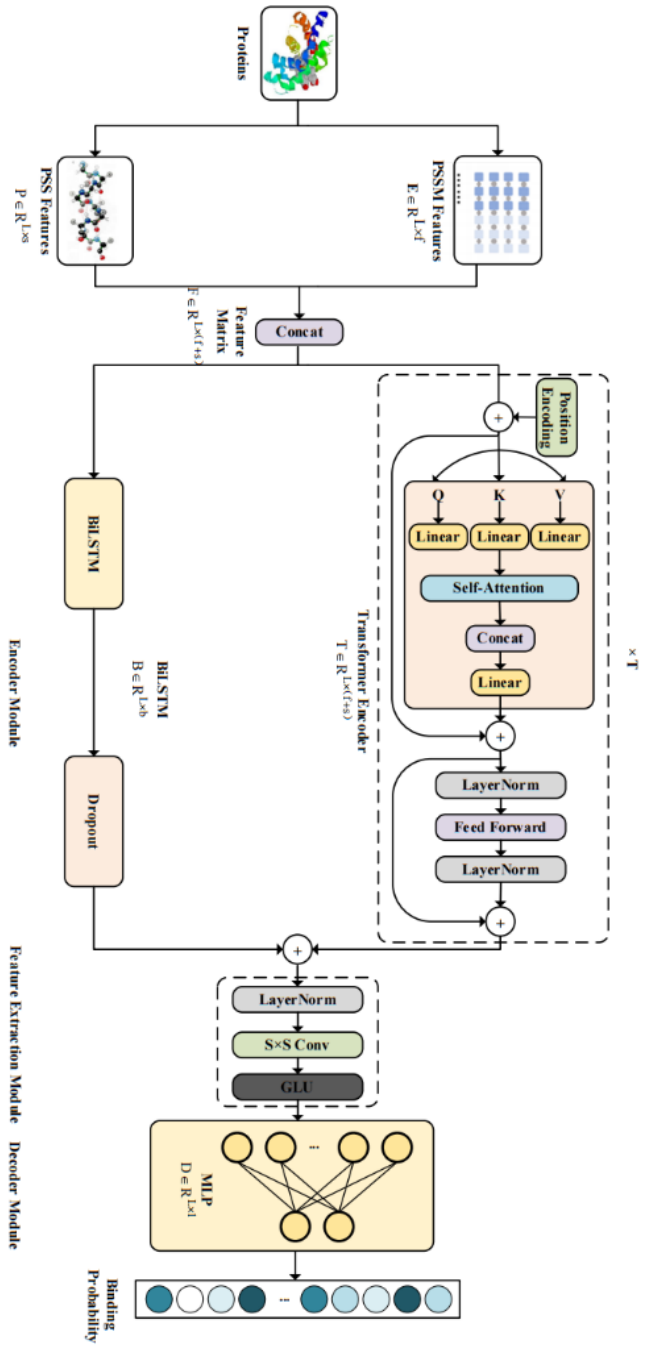


Fig. 1. Architecture.

3 Results

The proposed method demonstrated improved performance over existing classifiers, with significant improvements in MCC, SP, SN, and ACC on the PDNA-41 dataset. The combination of Transformer encoder and BiLSTM effectively captured both global and local residue features.

The study presents a robust method for identifying DNA-protein binding residues using deep learning. The method's effectiveness lies in its ability to capture long-range dependencies and local features, offering a user-friendly approach that requires only protein sequences as input. Future work will explore incorporating three-dimensional structural information and graph neural networks for further enhancements.

In this study, we propose an encoder-decoder model to predict protein-DNA binding sites. To represent a protein sequence, we use two sequence-based features, the evolutionary feature PSSM and the predicted secondary structure, respectively. Unlike current state-of-the-art methods, our model enables end to end prediction of an entire protein sequence without the need for feature pre-extraction for each residue using a sliding window technique, which demonstrates the ease of use of our model. Comparing with previous methods, our model achieves respectable performance on the PDNA-41 test set (MCC:0.343, SP:96.37%, SN:46.34%, ACC:94.79%), which proves the effectiveness of our model.

While our method has made some progress and can handle variable length protein sequences, it also limits our model to one protein input at a time. Therefore, we will further try more models for the problem of inconsistent protein sequence lengths. Given the success of graph neural networks in bioinformatics, we will try to employ graph structures to represent protein sequences to identify DNA binding residues. In addition, the features used in this work could be improved. With the great achievements in the field of protein structure prediction in recent years, we can use the predicted structural information to aid in this task.

Acknowledgments. This research was supported by the National Natural Science Foundation of China and other institutions.

Disclosure of Interests. The authors declare no conflict of interest.

References

Dobson C M. Chemical space and biology[J]. Nature, 2004, 432(7019): 824-828.11
Gao M, Skolnick J. The distribution of ligand-binding pockets around protein-protein interfaces suggests a general mechanism for pocket formation[J]. Proceedings of the National Academy of Sciences, 2012, 109(10): 3784-3789.
Zhao J, Cao Y, Zhang L. Exploring the computational methods for protein-ligand binding site prediction[J]. Computational and structural biotechnology journal, 2020, 18: 417-426.
Ofran Y, Mysore V, Rost B. Prediction of DNA-binding residues from sequence[J]. Bioin

formatics, 2007, 23(13): i347-i353.
5. Jones S, Van Heyningen P, Berman H M, et al. Protein-DNA interactions: a structural analysis[J]. Journal of molecular biology, 1999, 287(5): 877-896.
6. Smyth M S, Martin J H J. x Ray crystallography[J]. Molecular Pathology, 2000, 53(1): 8.
7. Nelson J D, Denisenko O, Bomsztyk K. Protocol for the fast chromatin immunoprecipitation (ChIP) method[J]. Nature protocols, 2006, 1(1): 179-185.
8. Heffler MA, Walters RD, Kugel J F. Using electrophoretic mobility shift assays to measure equilibrium dissociation constants: GAL4(p53 binding DNA as a model system[J]. Biochemistry and Molecular Biology Education, 2012, 40(6): 383-387 .
9. Hellman L M, Fried M G. Electrophoretic mobility shift assay (EMSA) for detecting protein±nucleic acid interactions[J]. Nature protocols, 2007, 2(8): 1849-1861.
10. Vajda S, Guarnieri F. Characterization of protein-ligand interaction sites using experimental and computational methods[J]. Current Opinion in Drug Discovery and Development, 2006, 9(3): 354.
11. Ding Y, Yang C, Tang J, et al. Identification of protein-nucleotide binding residues via graph regularized k-local hyperplane distance nearest neighbor model[J]. Applied Intelligence, 2022: 1-15.
12. Wang L, Brown S J. BindN: a web-based tool for efficient prediction of DNA and RNA binding sites in amino acid sequences[J]. Nucleic acids research, 2006, 34(suppl_2): W243-W248.
13. Chu W Y, Huang Y F, Huang C C, et al. ProteDNA: a sequence-based predictor of sequence-specific DNA-binding residues in transcription factors[J]. Nucleic acids research, 2009, 37(suppl_2): W396-W401.
14. Hwang S, Gou Z, Kuznetsov I B. DP-Bind: a web server for sequence-based prediction of DNA-binding residues in DNA-binding proteins[J]. Bioinformatics, 2007, 23(5): 634-636.
15. Wang L, Huang C, Yang M Q, et al. BindN+ for accurate prediction of DNA and RNA binding residues from protein sequence features[J]. BMC Systems Biology, 2010, 4: 1-9.
16. Si J, Zhang Z, Lin B, et al. MetaDBSite: a meta approach to improve protein DNA-binding
sites prediction[J]. BMC systems biology, 2011, 5(1): 1-7.
17. Hu J, Li Y, Zhang M, et al. Predicting protein-DNA binding residues by weightedly combining sequence-based features and boosting multiple SVMs[J]. IEEE/ACM transactions on computational biology and bioinformatics, 2016, 14(6): 1389-1398.
18. Liu R, Hu J. DNABind: A hybrid algorithm for structure(based prediction of DNA(binding residues by combining machine learning(and template(based approaches[J]. PROTEINS: structure, Function, and Bioinformatics, 2013, 81(11): 1885-1899.
19. Zhu Y H, Hu J, Song X N, et al. DNAPred: accurate identification of DNA-binding sites
from protein sequence by ensembled hyperplane-distance-based support vector ma chines[J]. Journal of chemical information and modeling, 2019, 59(6): 3057-3071.

20. Hu J, Bai Y S, Zheng L L, et al. Protein-dna binding residue prediction via bagging strategy and sequence-based cube-format feature[J]. IEEE/ACM transactions on computational biology and bioinformatics, 2021, 19(6): 3635-3645.

Reference

[1] Chou KC, Elrod DW. Prediction of membrane protein types and subcellular locations. Proteins: Structure, Function, and Bioinformatics 1999; 34(1), 137-153. https://doi.org/10.1002/(SICI)1097-0134(19990101)34:1<137::AID-PROT11>3.0.CO;2-O.
[2] Cai YD, Zhou GP, Chou KC. Support vector machines for predicting membrane protein types by using functional domain composition. Biophysical journal 2003; 84(5): 3257-3263. https://doi.org/10.1016/S0006-3495(03)70050-2.
[3] Cai YD, Chou KC. Predicting membrane protein type by functional domain composition and pseudo-amino acid composition. Journal of Theoretical Biology 2006; 238(2): 395-400. https://doi.org/10.1016/j.jtbi.2005.05.035.
[4] Chou KC, Shen HB. MemType-2L: a web server for predicting membrane proteins and their types by incorporating evolution information through Pse-PSSM. Biochemical and biophysicalresearch communications 2007; 360(2): 339-345. https://doi.org/10.1016/j.bbrc.2007.06.027.
[5] Liu H, Yang J, Wang M, Xue L, Chou KC. Using Fourier spectrum analysis and pseudo amino acid composition for prediction of membrane protein types. The Protein Journal 2005; 24(6):385-389. https://doi.org/10.1007/s10930-005-7592-4.
[6] Shen H, Chou KC. Using optimized evidence-theoretic K-nearest neighbor classifier and pseudo-amino acid composition to predict membrane protein types. Biochemical and biophysical research communications 2005; 334(1): 288-292. https://doi.org/10.1016/j.bbrc.2005.06.087.
[7] Shen HB, Yang J, Chou KC. Fuzzy KNN for predicting membrane protein types from pseudo-amino acid composition. Journal of theoretical biology 2006; 240(1): 9-13. https://doi.org/10.1016/j.jtbi.2005.08.016.
[8] Wang M, Yang J, Liu GP, Xu ZJ, Chou KC. Weighted-support vector machines for predicting membrane protein types based on pseudo-amino acid composition. Protein Engineering Design and Selection 2004; 17(6): 509-516.
[9] Wang M, Yang J, Liu GP, Xu ZJ, Chou KC. Weighted-support vector machines for predicting membrane protein types based on pseudo-amino acid composition. Protein Engineering Design and Selection 2004; 17(6): 509-516.
[10] Liu H, Wang M, Chou KC. Low-frequency Fourier spectrum for predicting membrane protein types. Biochemical and biophysical research communications 2005; 336(3): 737-739. https://doi.org/10.1016/j.bbrc.2005.08.160.
[11] Wang SQ, Yang J, Chou KC. Using stacked generalization to predict membrane protein types based on pseudo-amino acid composition. Journal of theoretical biology 2006; 242(4): 941-946. https://doi.org/10.1016/j.jtbi.2006.05.006.
[12] Chen YK, Li KB. Predicting membrane protein types by incorporating protein topology, domains, signal peptides, and physicochemical properties into the general form of Chou's pseudo amino acid composition. Journal of Theoretical Biology 2013; 318: 1-12. https://doi.org/10.1016/j.jtbi.2012.10.033.
[13] Han GS, Yu ZG, Anh V. A two-stage SVM method to predict membrane protein types by incorporating amino acid classifications and physicochemical properties into

a general form of Chou's PseAAC. Journal of Theoretical Biology 2014; 344: 31-39. https://doi.org/10.1016/j.jtbi.2013.11.017.
[14] Hayat M, Khan A. Predicting membrane protein types by fusing composite protein sequence features into pseudo amino acid composition. Journal of theoretical biology 2011; 271(1): 10-17. https://doi.org/10.1016/j.jtbi.2010.11.017.
[15] Hayat M, Khan A, Yeasin M. Prediction of membrane proteins using split amino acid and ensemble classification. Amino acids 2012; 42(6): 2447-2460. https://doi.org/10.1007/s00726-011-1053-5.
[16] Rezaei MA, Abdolmaleki P, Karami Z, Asadabadi EB, Sherafat MA, Abrishami-Moghaddam, H, Forouzanfar M. Prediction of membrane protein types by means of wavelet analysis and cascaded neural networks. Journal of theoretical biology 2008; 254(4): 817-820. https://doi.org/10.1016/j.jtbi.2008.07.012.
[17] Shen Y, Tang J, Guo F. Identification of protein subcellular localization via integrating evolutionary and physicochemical information into Chou's general PseAAC. Journal of Theoretical Biology 2019; 462: 230-239. https://doi.org/10.1016/j.jtbi.2018.11.012.
[18] Wang Y, Ding Y, Guo F, Wei L, Tang J. Improved detection of DNA-binding proteins via compression technology on PSSM information. PloS one 2017; 12(9): e0185587. https://doi.org/10.1371/journal.pone.0185587.
[19] Shen C, Ding Y, Tang J, Xu X, Guo F. An ameliorated prediction of drug–target interactions based on multi-scale discrete wavelet transform and network features. International journal of molecular sciences 2017; 18(8): 1781. https://doi.org/10.3390/ijms18081781.
[20] Ahmed N, Natarajan T, Rao KR. Discrete cosine transform. IEEE transactions on Computers 1974; 100(1): 90-93. https://doi.org/10.1109/T-C.1974.223784.
[21] Ding Y, Tang J, Guo F. Identification of protein–protein interactions via a novel matrix-based sequence representation model with amino acid contact information. International journal of molecular sciences 2016; 17(10): 1623. https://doi.org/10.3390/ijms17101623.
[22] Boeckmann B, Bairoch A, Apweiler R, Blatter MC, Estreicher A, Gasteiger E, Schneider M. The SWISS-PROT protein knowledgebase and its supplement TrEMBL in 2003. Nucleic acids research 2003; 31(1): 365-370.
[23] Li W, Godzik A, Cd-hit: a fast program for clustering and comparing large sets of protein or
nucleotide sequences, Bioinformatics 2006; 22(13): 1658-1659.
[24] Limin Fu, Beifang Niu, Zhengwei Zhu, Sitao Wu and Weizhong Li, CD-HIT: accelerated for clustering the next generation sequencing data. Bioinformatics 2012; 28(23): 3150-3152. https://doi.org/10.1093/bioinformatics/bts565.
[25] cheol Jeong J, Lin X, Chen XW. On position-specific scoring matrix for protein function prediction. IEEE/ACM transactions on computational biology and bioinformatics 2010; 8(2): 308-315. https://doi.org/10.1109/TCBB.2010.93.
[26] Nanni L, Brahnam S, Lumini A. Wavelet images and Chou's pseudo amino acid composition for protein classification. Amino Acids 2012; 43(2): 657-665. https://doi.org/10.1007/s00726-011-1114-9.
[27] Zhou D, Huang J, Schölkopf B. Learning with hypergraphs: Clustering, classification, and embedding. Advances in neural information processing systems 2006; 19.

[28] Huang Y, Liu Q, Metaxas D. Video object segmentation by hypergraph cut. In 2009 IEEE conference on computer vision and pattern recognition 2009; 1738-1745. https://doi.org/10.1109/CVPR.2009.5206795.
[29] Huang Y, Liu Q, Zhang S, Metaxas DN. Image retrieval via probabilistic hypergraph ranking. In 2010 IEEE computer society conference on computer vision and pattern recognition 2010; 3376-3383. https://doi.org/10.1109/CVPR.2010.5540012.
[30] Gao Y, Wang M, Zha ZJ, Shen J, Li X, Wu X. Visual-textual joint relevance learning for tag-based social image search. IEEE Transactions on Image Processing 2012; 22(1): 363-376. https://doi.org/10.1109/TIP.2012.2202676.
[31] Hwang T, Tian Z, Kuangy R, Kocher JP. Learning on weighted hypergraphs to integrate protein interactions and gene expressions for cancer outcome prediction. In 2008 Eighth IEEE International Conference on Data Mining 2008; 293-302. https://doi.org/10.1109/ICDM.2008.37.
[32] Gao Y, Wang M, Tao D, Ji R, Dai Q. 3-D object retrieval and recognition with hypergraph analysis. IEEE Transactions on Image Processing 2012; 21(9): 4290-4303. https://doi.org/10.1109/TIP.2012.2199502.
[33] Feng Y, You H, Zhang Z, Ji R, Gao Y. Hypergraph neural networks. In Proceedings of the AAAI conference on artificial intelligence 2019; 33(1): 3558-3565. https://doi.org/10.1609/aaai.v33i01.33013558.
[34] Henaff M, Bruna J, LeCun Y. Deep convolutional networks on graph-structured data. arXiv preprint arXiv: 1506.05163 2015. https://doi.org/10.48550/arXiv.1506.05163.
[35] Defferrard M, Bresson X, Vandergheynst P. Convolutional neural networks on graphs with fast localized spectral filtering. Advances in neural information processing systems 2016; 29.
[36] Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R. Dropout: a simple way to prevent neural networks from overfitting. The journal of machine learning research 2014; 15(1): 1929-1958.
[37] Kingma DP, Ba J. Adam. A method for stochastic optimization. arXiv preprint arXiv:1412.6980 2014..
[38] Alhamdoosh M, Wang D. Fast decorrelated neural network ensembles with random weights. Information Sciences 2014; 264: 104-117. https://doi.org/10.1016/j.ins.2013.12.016.
[39] Chou KC. Prediction of protein cellular attributes using pseudo‐amino acid composition. Proteins: Structure, Function, and Bioinformatics 2001; 43(3): 246-255. https://doi.org/10.1002/prot.1035.
[40] Wang L, Yuan Z, Chen X, Zhou Z. The prediction of membrane protein types with NPE. IEICE Electronics Express 2010; 7(6): 397-402. https://doi.org/10.1587/elex.7.397.
[41] Shen HB, Chou KC. Using ensemble classifier to identify membrane protein types. Amino acids 2007; 32(4): 483-488. https://doi.org/10.1007/s00726-006-0439-2.

Improving Drug-Target Interaction Predictions Through an Explainable Graph Transformer Model

Baozhong Zhu [1] and Hongjie Wu [1]

[1] Suzhou University of Science and Technology
hongjie.wu@qq.com

Abstract. Drug discovery is a complex and time-consuming process. Identifying drug-target interactions (DTIs) is crucial for early-stage drug development. This study introduces a novel model for DTI prediction that leverages protein binding sites and self-attention mechanisms. The model achieves high performance in DTI prediction and provides interpretability by identifying protein regions interacting with ligands.

Keywords: Drug-target Interaction Prediction.

1 Introduction

Drug discovery involves identifying drug-target interactions, which is a complex and resource-intensive process. Computational methods have been proposed to facilitate DTI identification and expedite drug discovery. This study presents a novel architecture for DTI prediction using protein binding sites and self-attention mechanisms.

Drug discovery is a complex and time-consuming process, and despite significant investments, success rates remain suboptimal [1]. Proteins are the primary targets of drugs and the identification of drug-target interaction (DTI) has become a crucial task in early-stage drug development and drug repurposing [2]. Since experimental DTI studies are expensive and time-consuming, computational methodologies have been proposed to facilitate the identification of putative DTI, thereby expediting the process of drug discovery [3]. One of the main methods for virtual screening involves predicting potential drugs by screening out drug candidate ligands for receptor proteins of interest from large-scale compound ligand libraries using many calculations [4]. Virtual screening methods can be divided into two categories: receptor-based virtual screening and ligand-based virtual screening. Receptor-based virtual screening mainly studies the three-dimensional structure of proteins and seeks interactions with small molecule compounds from the three-dimensional structure, making it also known as structure-based virtual screening [5]. However, these methods have practical limitations due to their heavy reliance on the high-quality three-dimensional structure of proteins and their computational expenses and inefficiencies. Ligand-based virtual screening typically begins with ligands and analyzes molecular structure and activity information of known inhibitors to summarize structural features that significantly contribute to their binding

capacity. This learned knowledge is then used to screen new ligands to find compound molecules that meet the requirements. Virtual screening methods often rely on predicting drug-target interactions, which can be understood as a series of continuous values that express the intensity of different drug-target interactions.

With the rapid development of deep learning methods [6], researchers have used deep learning models to measure drug-target interactions as binary classification tasks[7]. These DTI prediction models have been hugely successful because they can automatically capture data depth features, resulting in better models with excellent capabilities in complex molecular data processing [8]. DTI's deep learning models can be divided into two main categories [9]. One type act on processing sequence-based representations of input data, such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs). In related works, Peng proposed a method based on convolutional neural networks to extract drug and protein features from heterogeneous networks, and used convolutional neural network models to predict the interaction between drugs and proteins [10]. Karimi proposes a semi-supervised deep learning model that unifies recurrent and convolutional neural networks to jointly encode molecular representation and predict affinity using unlabeled and labeled data [11].

However, these models usually express drugs in the form of strings, and one-dimensional sequences are not a natural way of expressing molecules. Therefore, to compensate for the lack of molecular structure information, a second type of deep learning model, the graph neural network (GNN), was introduced, and the use of graph convolutional networks has also proven to be more beneficial for computational drug discovery [12]. GNN uses a graphical description of molecules, where atoms and chemical bonds correspond to nodes and edges, respectively [13]. The most commonly used GNN-based models today are the graph convolutional neural network (GCNN) [14] and the graph attention network (GAT), which is one of the variants of GCNN. Related work includes Zhao using the constructed graph convolutional network to learn the drug-protein pairs built to improve the prediction accuracy [15]. Zhao proposes a new graph convolutional DTI prediction model. Specifically, the first-order neighbor information of a node can be aggregated through GCN; The high-order neighbor information of the node is learned by the graph embedding method, which improves the accuracy of prediction [16].

Despite the impressive performance of both CNN-based and graph-based neural network methods in DTI prediction, certain challenges remain unresolved [17]. One significant limitation of most deep learning methods is that they employ only a few CNN layers, resulting in the compression of all feature information into a small area, which may cause the loss of local features of the original data. Moreover, all graph-based models are currently represented using the amino acid sequence of the protein, which cannot capture the crucial 3D structural features that are essential in DTI prediction.

Obtaining a high-resolution 3D structure of a protein is a difficult task due to its complex nature and large number of atoms, necessitating a massive 3D (sparse) matrix to capture the entire structure. This paper proposes a novel approach for predicting DTI that leverages the structural features of small molecules and protein binding sites in the form of graphs. To preserve the influence of molecular structure on the prediction results, a transformer model is introduced to extract global features. Moreover, a

selfattention Bidirectional Long Short-Term Memory mechanism is employed to identify the parts of the protein that are most likely to bind to a given drug, thereby enhancing the model's interpretability

2 Method

The proposed framework consists of four main modules. The Data Preparation module extracts protein binding sites. The Graph Embedding Learning module generates a graph map of protein pockets and ligands using TAGCN to extract global and local features. The Feature Extraction module uses a transformer block and a Self-attentive BiLSTM block to learn the relationship between ligands and protein binding sites. The Prediction module uses a binary classifier for DTI prediction.

2.1 Framework

The framework includes a pretreatment module for identifying protein binding sites, a graph representation module, a feature extraction module with a transformer block and BiLSTM block, and a prediction module for DTI prediction.

TAGCN is used to generate embeddings from the graph representation, capturing both local and global features.The transformer block focuses on global information, while the Self-attentive BiLSTM block identifies key contributors to predicted interactions. A two-layer fully connected neural network with a logistic sigmoid function predicts DTI probabilities.

Our model was subjected to rigorous evaluation using two widely recognized DTI datasets, namely, the DUD-E dataset and the Human dataset. These benchmark datasets are commonly used in the field of drug target interaction prediction. The DUD-E dataset comprises 102 targets belonging to eight distinct protein families. Each target comprises roughly 224 active compounds and more than 10,000 bait molecules. On the other hand, the Human dataset was constructed by combining a highly credible and reliable set of negative drug-protein samples with known positive samples using systematic in silico screening methods. The dataset contains 5423 interactions between drug and target molecules. Table 1 presents a summary of the key statistics for these two datasets. All datasets are publicly available. DUD-E dataset is available at http://dude.docking.org, Human dataset is available at https://github.com/IBMInterpretableDTIP.

2.2 Train

The model was evaluated using the DUD-E dataset and the Human dataset, which are standard benchmarks for DTI prediction.

Proteins are represented as graphs with atoms as nodes and connections as edges. Ligands are represented in SMILE format and encoded similarly.

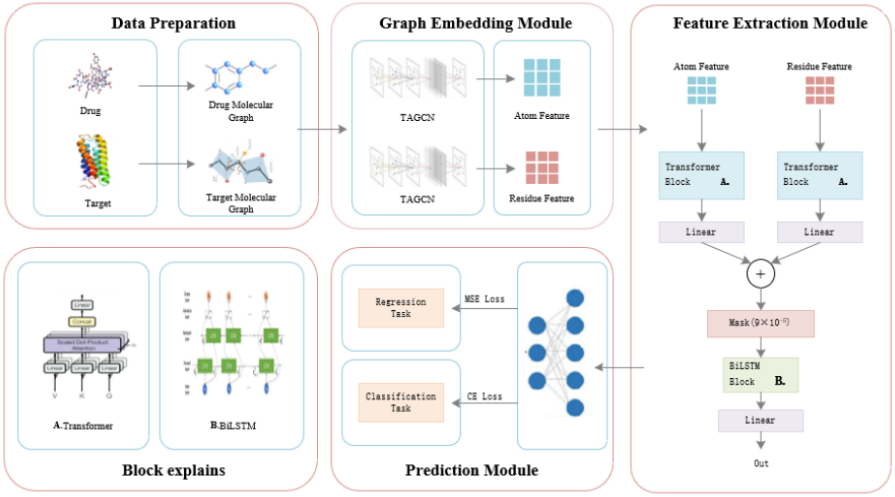


Fig. 1. Architecture.

3 Results

The model demonstrated superior performance over existing DTI predictive models. It achieved high AUC scores, precision, recall, and F1 scores on the Human dataset and showed significant enrichment in the DUD-E dataset comparison. The model's improved performance can be attributed to sophisticated input representations, robust feature extraction mechanisms, and the interpretability provided by the self-attention mechanism. The proposed model effectively captures drug-target interactions and provides interpretability by identifying specific protein binding sites. This approach holds promise for accelerating drug discovery processes.

We posit that the improved performance of our proposed model can be attributed to several factors:

(1) Input representation plays a crucial role in predicting the binding affinity of drug-target complexes. Utilizing more sophisticated input representations, such as structural diagrams, can aid in capturing crucial structural information regarding molecules.

(2) Feature extraction technique is an important consideration, and transformerbased architectures provide a robust automatic feature extraction mechanism that can capture high-order nonlinear relationships. Additionally, graph-based neural networks that employ graphical representations of drugs and proteins can effectively capture the topological relationships between drug molecules and target proteins, further enhancing the performance.

(3) To more effectively model and interpret the binding relationship of drug-target complexes, we introduce a self-attentive BiLSTM with masks. This model not only retains past and future information of the sequence input flowing in both directions but also explicates the degree of binding of drug-target complexes through the attention weight ratio.

Acknowledgments. This research was supported by the National Natural Science Foundation of China and other institutions.

Disclosure of Interests. The authors declare no conflict of interest.

References

1. Abbasi Mesrabadi, H., Faez, K. & Pirgazi, J. Drug–target interaction prediction based on protein features, using wrapper feature selection. Sci Rep 13, 3594 (2023). https://doi.org/10.1038/s41598-023-30026-y.
2. Soh, J., Park, S. & Lee, H. HIDTI: integration of heterogeneous information to predict drug target interactions. Sci Rep 12, 3793 (2022). https://doi.org/10.1038/s41598-022-07608-3.
3. Azuaje, F., Zhang, L., Devaux, Y. et al. Drug-target network in myocardial infarction reveals multiple side effects of unrelated drugs. Sci Rep 1, 52 (2011). https://doi.org/10.1038/srep00052.10
4. Beroza, P., Crawford, J.J., Ganichkin, O. et al. Chemical space docking enables large-scale structure-based virtual screening to discover ROCK1 kinase inhibitors. Nat Commun 13, 6447 (2022). https://doi.org/10.1038/s41467-022-33981-8.
5. Crunkhorn, S. Novel virtual screening approach. Nat Rev Drug Discov 16, 18 (2017). https://doi.org/10.1038/nrd.2016.272.
6. Ding, Yijie, Jijun Tang, and Fei Guo. "Identification of drug–target interactions via dual laplacian regularized least squares with multiple kernel fusion." Knowledge-Based Systems 204 (2020): 106254.
7. Ding, Yijie, Jijun Tang, and Fei Guo. "Identification of drug–target interactions via fuzzy bipartite local model." Neural Computing and Applications 32 (2020): 10303-10319.
8. Ding, Yijie, Jijun Tang, and Fei Guo. "Identification of drug-side effect association via semisupervised model and multiple kernel learning." IEEE journal of biomedical and health informatics 23.6 (2018): 2619-2632.
9. Ding, Yijie, Jijun Tang, and Fei Guo. "Identification of drug-target interactions via multiview graph regularized link propagation model." Neurocomputing 461 (2021): 618-631.
10. Peng J, Li J, Shang X. A learning-based method for drug-target interaction prediction based on feature representation learning and deep neural network. BMC Bioinformatics. 2020 Sep 17;21(Suppl 13):394. doi: 10.1186/s12859-020-03677-1. PMID: 32938374; PMCID: PMC7495825.
11. Karimi M, Wu D, Wang Z, Shen Y. DeepAffinity: interpretable deep learning of compound protein affinity through unified recurrent and convolutional neural networks. Bioinformatics. 2019 Sep 15;35(18):3329-3338. doi: 10.1093/bioinformatics/btz111. PMID: 30768156; PMCID: PMC6748780.
12. Veličković, Petar, et al. "Graph attention networks." arXiv preprint arXiv:1710.10903 (2017).
13. Torng, Wen, and Russ B. Altman. "Graph convolutional neural networks for predicting drug-target interactions." Journal of chemical information and modeling 59.10 (2019): 4131- 4149.

14. Lim, Jaechang, et al. "Predicting drug–target interaction using a novel graph neural network with 3D structure-embedded graph representation." Journal of chemical information and modeling 59.9 (2019): 3981-3988.
15. Tianyi Zhao, Yang Hu, Linda R Valsdottir, Tianyi Zang, Jiajie Peng, Identifying drug–target interactions based on graph convolutional network and deep neural network, Briefings in Bioinformatics, Volume 22, Issue 2, March 2021, Pages 2141–2150, https://doi.org/10.1093/bib/bbaa044
16. Zhao BW, You ZH, Hu L, Guo ZH, Wang L, Chen ZH, Wong L. A Novel Method to Predict Drug-Target Interactions Based on Large-Scale Graph Representation Learning. Cancers (Basel). 2021 Apr 27;13(9):2111. doi: 10.3390/cancers13092111. PMID: 33925568; PMCID: PMC8123765.
17. Ding, Yijie, Jijun Tang, and Fei Guo. "Protein crystallization identification via fuzzy model on linear neighborhood representation." IEEE/ACM transactions on computational biology and bioinformatics 18.5 (2019): 1986-1995.
18. Saberi Fathi, Seyed Majid, and Jack A. Tuszynski. "A simple method for finding a protein's ligand-binding pockets." BMC Structural Biology 14.1 (2014): 1-9.
19. Ding, Yijie, Jijun Tang, and Fei Guo. "Human protein subcellular localization identification via fuzzy model on kernelized neighborhood representation." Applied Soft Computing 96 (2020): 106596.11
20. Wu, Hongjie, et al. "Empirical potential energy function toward ab initio folding G protein coupled receptors." IEEE/ACM Transactions on Computational Biology and Bioinformatics 18.5 (2020): 1752-1762.
21. Wang, Hao, et al. "Exploring associations of non-coding RNAs in human diseases via three matrix factorization with hypergraph-regular terms on center kernel alignment." Briefings in Bioinformatics 22.5 (2021): bbaa409.
22. Yang, Hongpeng, et al. "Drug–disease associations prediction via multiple kernel-based dual graph regularized least squares." Applied Soft Computing 112 (2021): 107811.
23. Sun, Mengwei, et al. "MLapSVM-LBS: Predicting DNA-binding proteins via a multiple Laplacian regularized support vector machine with local behavior similarity." Knowledge Based Systems 250 (2022): 109174.
24. Du, Jian, et al. "Topology adaptive graph convolutional networks." arXiv preprint arXiv:1710.10370 (2017).
25. Vaswani, Ashish, et al. "Attention is all you need." Advances in neural information processing systems 30 (2017).
26. Zhou, Peng, et al. "Attention-based bidirectional long short-term memory networks for relation classification." Proceedings of the 54th annual meeting of the association for computational linguistics (volume 2: Short papers). 2016.
27. Yazdani-Jahromi, Mehdi, et al. "AttentionSiteDTI: an interpretable graph-based model for drug-target interaction prediction using NLP sentence-level relation classification." Briefings in Bioinformatics 23.4 (2022): bbac272.
28. Liu, Hui, et al. "Improving compound–protein interaction prediction by building up highly credible negative samples." Bioinformatics 31.12 (2015): i221-i229.
29. Kipf, Thomas N., and Max Welling. "Semi-supervised classification with graph convolutional networks." arXiv preprint arXiv:1609.02907 (2016).

30. Wang, Erniu, et al. "A graph convolutional network–based method for chemical-protein interaction extraction: algorithm development." JMIR Medical Informatics 8.5 (2020): e17643.
31. Wu, Yifan, et al. "BridgeDPI: a novel graph neural network for predicting drug–protein interactions." Bioinformatics 38.9 (2022): 2571-2578.
32. Durrant, Jacob D., and J. Andrew McCammon. "NNScore 2.0: a neural-network receptor– ligand scoring function." Journal of chemical information and modeling 51.11 (2011): 2897- 2903.
33. Ballester, Pedro J., and John BO Mitchell. "A machine learning approach to predicting protein–ligand binding affinity with applications to molecular docking." Bioinformatics 26.9 (2010): 1169-1175.
34. Ragoza, Matthew, et al. "Protein–ligand scoring with convolutional neural networks." Journal of chemical information and modeling 57.4 (2017): 942-957.
35. Torng, Wen, and Russ B. Altman. "Graph convolutional neural networks for predicting drug-target interactions." Journal of chemical information and modeling 59.10 (2019): 4131- 4149.
36. Tsubaki, Masashi, Kentaro Tomii, and Jun Sese. "Compound–protein interaction prediction with end-to-end learning of neural networks for graphs and sequences." Bioinformatics 35.2 (2019): 309-318.
37. Wang, Erniu, et al. "A graph convolutional network–based method for chemical-protein interaction extraction: algorithm development." JMIR Medical Informatics 8.5 (2020): e17643.

Author Index

Table of Email Addresses from the Corresponding Authors

0001	18631511729@163.com
0004	heli@xidian.edu.cn
0015	heli@xidian.edu.cn
0027	heli@xidian.edu.cn
0041	18015612591@163.com
0051	18015612591@163.com
0054	hongjie.wu@qq.com
0063	hongjie.wu@qq.com
0072	hongjie.wu@qq.com
0079	hongjie.wu@qq.com
0088	hongjie.wu@qq.com