Journal of Computer Technology and Software

ISSN:2998-2383

Vol. 3, No. 9, 2024

Multivariate Time Series Forecasting and Classification via GNN and Transformer Models

Jiawei Wang
University of California, Los Angeles, USA
jiaweiwang@ucla.edu

Abstract: This paper proposes a time series data mining method based on graph neural network and Transformer architecture, aiming to solve the challenges of modeling complex dependencies and dynamic features in multivariate time series. By introducing an adaptive adjacency matrix, the model can dynamically learn the relationship between variables and use a graph neural network to capture local dependency characteristics; combined with the multi-head attention mechanism of Transformer, the global time dependency of time series is further modeled. The experiment selected two tasks, regression and classification, for verification. The results on the power load forecasting and UCI human activity recognition datasets show that the proposed method is superior to traditional statistical models, machine learning models, and existing deep learning models in various indicators (such as MSE, MAE, accuracy, F1 value), which fully demonstrates its superior performance. In addition, the ablation experiment analysis further verifies the contribution of each key module to the model performance and demonstrates the model's ability to learn variable relationships and capture key information of time series through visual analysis. The study shows that the proposed method not only has a strong time-series mining capability but also has good generalization and robustness. In the future, we will further explore its application potential in label-scarce scenarios and real-time tasks, and improve its applicability and deployment efficiency in a wider range of fields by combining self-supervised learning and model lightweight technology.

Keywords: Time series data mining, graph neural network, Transformer, adaptive adjacency matrix

1. Introduction

Time series data is widely present in various fields such as medicine, transportation, and finance, and is one of the important objects of data mining research. By analyzing the potential patterns and regularities in time series data, it can help people predict future trends, identify abnormal events, and perform tasks such as classification and clustering, thereby providing support for scientific research, business decisions, and public services [1]. However, time series data is usually high-dimensional, dynamic, and complex, and may contain problems such as noise and irregular sampling, which brings many challenges to time series data mining [2].

Traditional time series data mining methods are mostly based on statistics and machine learning techniques. These methods rely on artificially constructed features and simple linear models, and it is difficult to effectively capture complex time dependencies and interactive characteristics between multiple variables. With the development of deep learning technology, models such as recurrent neural networks (RNN) and convolutional neural networks (CNN) have been widely used in time series data analysis, showing powerful feature extraction capabilities [3]. However, these models still have certain limitations in dealing with long-term dependencies and complex multivariate dependencies.

In recent years, graph neural networks (GNNs) and Transformer architectures have gradually become research hotspots in the field of time series data mining due to their excellent modeling capabilities and flexibility [4]. GNN can capture the complex relationship between multiple variables in time series data through graph structure, while the Transformer architecture can effectively handle long-term dependencies and global information in sequence data with its self-attention mechanism. The combination of the two provides a new idea for solving key problems in time series data mining.

The time series data mining method based on graph neural network and Transformer architecture can make full use of the spatiotemporal structural characteristics of data and effectively improve the accuracy and robustness of prediction and classification. This study proposes a time series data mining model that integrates graph neural network and Transformer architecture. Through the synergy of adaptive graph construction, time dependency modeling, and attention mechanism [5], it effectively copes with the complexity, diversity, and noise interference in time series data.

This paper verifies the effectiveness and superiority of the proposed method by conducting experiments on public data sets in multiple fields. The research results show that compared with traditional methods, the time series data mining method based on graph neural network and Transformer has significantly improved in prediction accuracy, classification performance, and model

generalization ability, showing its wide application prospects in time series analysis.

2. Method

In this study, in order to fully mine the multivariate relationships and temporal characteristics in time series data, a time series data mining method based on graph neural network and Transformer architecture is proposed. This method is mainly divided into two stages: first, the complex dependencies of multivariate time series are modeled using a graph neural network, and then the global temporal dependencies in the sequence are captured by the Transformer to complete the feature learning of time series and the realization of task objectives. The model architecture is shown in Figure 1.

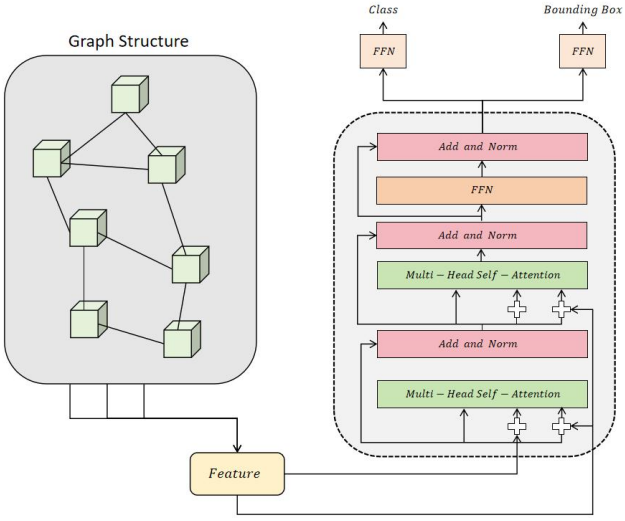

Figure 1. Network architecture diagram

First, for a given multivariate time series data, it can be represented as $X \in R^{N \times T \times F}$, where N represents the number of variables, T represents the length of the time step, and F represents the feature dimension of each variable. To model the complex relationship between variables, a dynamic graph is constructed using a graph neural network, where each variable is used as a node and the edge weights between nodes are learned through an adaptive mechanism. Specifically, by introducing a dynamic adjacency matrix $A \in R^{N \times N}$, the relationship between variables can be described, and its value is calculated by the characteristics of the time series:

$$A_{i,j} = soft\max(\phi(h_i, h_j))$$

Where h_i, h_j is the initial feature representation of nodes i and j, respectively, and $\phi()$ represents a similarity function (e.g., a function based on inner product or multi-layer perceptron). The generation of an adaptive adjacency matrix ensures that the model can capture the dynamic relationship between variables.

After constructing the dynamic graph, the graph convolutional network (GCN) is used to extract the interaction features between variables. The graph convolution operation can be expressed as:

$$H^{(l+1)} = \sigma(AH^{(l)}W^{(l)})$$

Where $H^{(l)}$ is the input feature representation of the l-th layer of graph convolution, $W^{(l)}$ is the learned weight matrix, and $\sigma()$ is the activation function (such as ReLU). After multiple layers of graph convolution, the fused representation $H^{(l)} \in R^{N \times F'}$ of each variable is obtained, where E is the final feature dimension.

After completing the modeling of the relationship between variables, the extracted features are sent to the Transformer module for learning time series features. In order to adapt to the time series input, the output of the graph convolution is first time-embedded and expanded to obtain $Z \in R^{T \times N \times F'}$. Then, positional encoding is introduced to represent the position of the time step. The positional encoding calculation formula is:

$$PE(t, 2i) = \sin(\frac{t}{10000^{2i/F'}})$$

$$PE(t, 2i+1) = \cos(\frac{t}{10000^{2i/F'}})$$

Where t represents the time step and i represents the index of the feature dimension. By adding position encoding, the model can better capture time information.

The Transformer module captures the global dependency of the time series through a multi-head self-attention mechanism and a feedforward network. The calculation formula of the self-attention mechanism is:

$$Attention(Q, K, V) = soft\max(\frac{QK^T}{\sqrt{d_k}})V$$

Q, K, V is the query matrix, key matrix and value matrix, which are obtained by linear transformation of input features, and d_k is the scaling factor of the attention mechanism, which is used to balance the numerical scale of the dot product. The calculation of multi-head attention is performed in parallel by multiple attention heads. The specific formula is:

$$MultiHead(X) = Concat(head_i, ..., head_h)W^O$$

The calculation of each attention head follows the above single-head attention formula.

In the encoder structure of Transformer, the output of multi-head attention and feedforward network is fused through residual connection and layer normalization, and finally forms the global feature representation of time series. Combining the local features of graph convolution and the global features of Transformer, the model can show strong prediction and classification capabilities in time series mining.

The training goal of the model is to minimize the task-related loss function. This paper conducts two tasks. First, the target loss function of the prediction task is given:

$$L_{pred} = \frac{1}{T}\sum_{i=1}^{T}(y_t - y'_t)^2$$

The classification task uses cross entropy loss :

$$L_{class} = -\sum_{i=1}^{C} y_i \log(y'_i)$$

y_t and y'_t are the true value and predicted value respectively, and C is the number of categories.

3. Experiment

3.1 Datasets

In this study, in order to verify the effectiveness and versatility of the proposed algorithm, two representative datasets were selected for experiments. The first dataset is the UCI Human Activity Recognition Dataset [6], which contains time series data of various human activities collected by the accelerometer and gyroscope of a smartphone. The dataset covers 6 activity categories, including walking, going up and down stairs, sitting, standing, and lying down. Each record contains time series features (such as acceleration and angular velocity) and corresponding activity labels. This dataset is widely used in classification tasks and can effectively verify the performance of the model in time series classification.

The second dataset is the Electricity Load Forecasting Dataset, which contains electricity consumption in multiple regions over a period of time, including date, time, electricity usage, and other relevant environmental variables (such as temperature, humidity, etc.). This dataset is widely used in time series forecasting tasks, aiming to predict future electricity demand based on historical electricity usage data. Electricity load data has significant time dependence and multivariate relationships and is an important benchmark dataset for verifying the model's ability to handle complex dependency characteristics in forecasting tasks. Through experimental analysis of these two datasets, the applicability and performance of the proposed algorithm in time series classification and forecasting tasks can be comprehensively evaluated.

3.2 Experimental Results

In order to evaluate the performance of the proposed algorithm in the time series forecasting task, this study selected the power load forecasting dataset and conducted comparative experiments with the current mainstream regression models. The comparative models include the classic autoregressive integrated moving average model (ARIMA) [7] based on statistics, random forest regression (RF) [8]based on machine learning, support vector regression (SVR) [9], and deep learning models such as long short-term memory network (LSTM) [10]and time series transformer [11] based on attention mechanism. The experimental results are shown in Table 1.

Table 1. Experimental results

Model	MAE	MSE	R^2
ARIMA	34.56	1456.34	0.752
RF	28.32	1298.45	0.812
SVR	29.45	1345.67	0.795
LSTM	24.12	1154.78	0.845
Time Transformer	22.67	1098.56	0.862
Ours	18.45	984.23	0.889

From the experimental results, it can be seen that the proposed method (Ours) outperforms the comparison model in all evaluation indicators. Specifically, in terms of MAE and MSE, the proposed method reached 18.45 and 984.23, respectively, which is significantly lower than the traditional statistical method ARIMA by 46.6% and 32.4%. Compared with the deep learning models LSTM and Time Transformer, the MSE of our method is reduced by 14.8% and 10.4%, and the MAE is reduced by 23.5% and 18.6%, respectively, showing the obvious advantage of the proposed method in reducing the prediction error. In addition, from the perspective of R^2 value, the proposed method reached 0.889, which is further improved compared with 0.862 of Time Transformer, indicating that the proposed method has higher accuracy and reliability when fitting complex time series data.

These results show that this study successfully captured the local variable relationship and global time dependency characteristics in the time series by combining the graph neural network and the Transformer architecture, effectively enhancing the model's predictive ability. Compared with ARIMA and machine learning models (such as RF and SVR), this method automatically learns data features through a deep learning structure, reducing the limitations of manual feature engineering; compared with LSTM and Time Transformer, this method shows stronger modeling capabilities and generalization performance when dealing with complex dependencies in multivariate time series.

Secondly, we not only conducted experiments on regression tasks, but also further verified the effectiveness of the proposed method on classification tasks. The UCI human activity recognition dataset was selected as the experimental

object. This dataset contains 6 types of human activity labels, aiming to evaluate the applicability of the model to time series classification tasks. The comparison models include support vector machine (SVM), random forest (RF), convolutional neural network (CNN), long short-term memory network (LSTM), and Transformer-based classification model. By comparing and analyzing the classification accuracy (Accuracy), precision (Precision), recall (Recall), and F1 value of these models under the same conditions, the performance advantages of the model proposed in this paper in time series classification tasks can be fully demonstrated. The experimental results are shown in Table 2.

Table 2. Classification experimental results

Model	ACC	Precision	Recall	F1
SVM	0.8145	0.7923	0.7867	0.7895
RF	0.8432	0.8245	0.8189	0.8217
CNN	0.8678	0.8512	0.8456	0.8484
LSTM	0.8845	0.8723	0.8689	0.8706
Transformer	0.8967	0.8845	0.8812	0.8828
Ours	0.9212	0.9134	0.9089	0.9111

From the experimental results, it can be seen that the proposed method (Ours) shows obvious advantages in classification tasks and outperforms other comparison models in all evaluation indicators. Specifically, the accuracy (ACC) of this method reaches 0.9212, which is 2.46% and 3.68% higher than the Transformer-based model (0.8967) and LSTM (0.8845), respectively. In addition, the precision (Precision) and recall (Recall) of this method are 0.9134 and 0.9089, respectively, which shows that the model can better balance the classification performance of minority and majority categories. In contrast, traditional machine learning models (such as SVM and RF) perform relatively poorly because they cannot fully capture the global dependencies and relationships between variables of time series. From the perspective of F1 value, the proposed method reached 0.9111, which is 2.83% higher than Transformer and 4.05% higher than LSTM. This further verifies the overall advantage of this method in time series classification tasks. Thanks to the ability of combining graph neural networks to capture complex relationships between variables and Transformer to model global time dependencies, this method shows excellent classification accuracy and robustness in multi-category classification scenarios. Experimental results show that this method can not only effectively handle time series classification tasks, but also surpass the performance of existing mainstream models, providing a better solution for time series data mining.

In addition, this paper also conducted ablation experiments using classification tasks to verify the impact of different model components on the overall performance. By gradually removing key modules, such as the neural network part, the adaptive adjacency matrix generation module, and the Transformer multi-head attention mechanism, the contribution of each part to the model classification ability is evaluated. The ablation experiment can further reveal the core effectiveness of this method and provide an important reference for model design and optimization. The experimental results are shown in Table 3.

Table 3. Classification experimental results

Model	ACC	Precision	Recall	F1
With out GNN	0.8912	0.8745	0.8698	0.8721
Without Adaptive Adjacency Matrix	0.9045	0.8923	0.8878	0.8900
Without Multi-head Attention in Transformer	0.9134	0.9021	0.8984	0.9002
Ours	0.9212	0.9134	0.9089	0.9111

From the ablation experiment results, it can be seen that gradually removing the key modules of the model has a significant impact on the overall performance. After removing the graph neural network (GNN), the accuracy (ACC) of the model dropped to 0.8912 and the F1 value dropped to 0.8721, indicating that GNN plays an important role in extracting complex dependencies between multiple variables, and the absence of this module will lead to a significant decrease in classification ability. In addition, after removing the adaptive adjacency matrix generation module, although the performance is slightly improved to an accuracy of 0.9045, there is still a certain gap compared to the complete model, indicating that dynamically constructing a variable relationship graph can significantly improve the modeling effect of the relationship between variables.

When the multi-head attention mechanism of Transformer is removed, the accuracy of the model further drops to 0.9134 and the F1 value drops to 0.9002, which shows that the multi-head attention mechanism plays a key role in capturing the global dependency characteristics of time series. In contrast, the complete model (Ours) performs best in all indicators, with an ACC of 0.9212 and an F1 value of 0.9111, which verifies that the method in this paper can make better use of the structural characteristics of time series data when combining graph neural networks with Transformer architectures, and show the best classification performance and robustness.

Furthermore, this paper gives the multi-head attention weights of the Transformer, and the experimental results are shown in Figure 2.

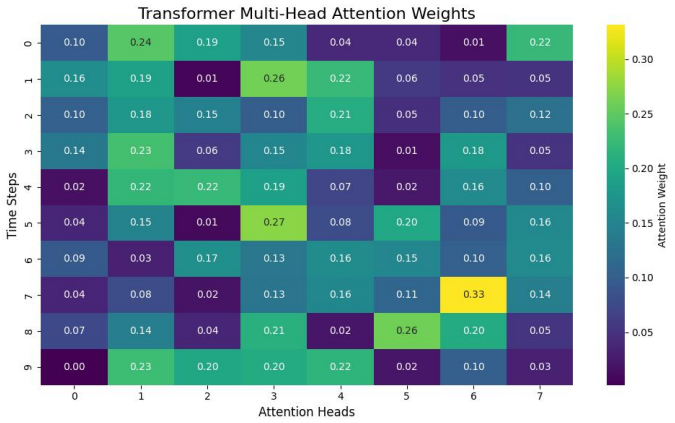

Figure 2. Transformer Multi-Head Attention Weight

As can be seen from the figure, the distribution of the Transformer model's multi-head attention weights between time steps and attention heads is significantly different, which reflects the model's focus when processing features at different time steps. Some time steps (such as step 8) have higher weights for some attention heads (such as head 6), which may indicate that these time steps contain important feature information related to the task, while the distribution of attention weights at other time steps is relatively scattered, indicating that the model may pay more attention to global features at these time steps.

In general, the introduction of the multi-head attention mechanism enables the model to capture the complex interactions between different time steps and features in the time series and assign more weights to key positions. This flexible weight distribution not only improves the representation ability of the model but also verifies the effectiveness of the multi-head attention mechanism in time series data modeling, providing strong support for subsequent classification or prediction tasks. The experimental results clearly demonstrate the model's comprehensive attention to global and local features, laying the foundation for performance improvement.

Similarly, this paper also gives the heat map of the adaptive adjacency matrix. The experimental results are shown in Figure 3.

As can be seen from the figure, the adaptive adjacency matrix effectively captures the dynamic relationship between variables. The values of the diagonal are all 1, indicating that the relationship strength between each variable and itself is the strongest, which is logical. In addition, the distribution of weight values between different variables is obviously different. For example, the relationship strength between variable 0 and variable 2 is high (weight is 0.78), indicating that they may have strong correlation or interaction characteristics, while the weight between variable 3 and variable 5 is low (only 0.25), indicating that their dependence is weak.

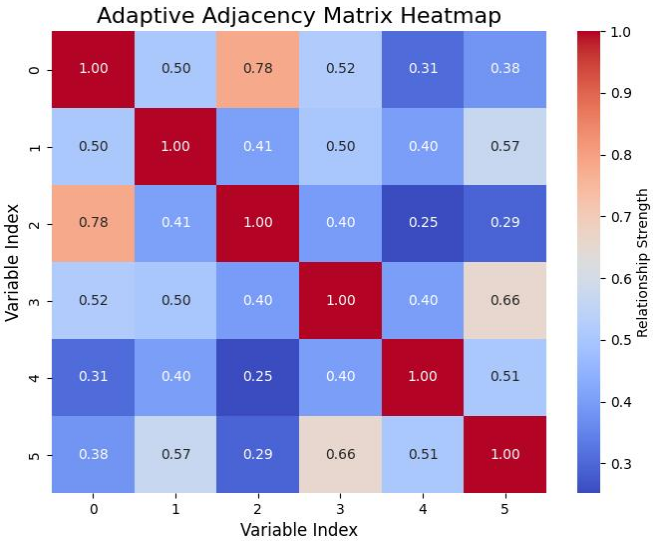

Figure 3. Adaptive Adjacency Matrix Heatmap

Overall, the heat map reflects the model's ability to capture complex dependencies among multiple variables. The asymmetric distribution of weight values indicates that the model can adaptively adjust the weights between variables according to the dynamic characteristics of the time series, which provides important support for subsequent feature extraction and task prediction. The experimental results further verify the importance and effectiveness of the adaptive adjacency matrix in multivariate time series modeling.

4. Conclusion

This paper proposes a time series data mining method based on graph neural network and Transformer architecture. By combining the local variable relationship modeling capability of graph neural network and the global time dependency capture capability of Transformer, it effectively solves the modeling problem of complex dependencies and dynamic features in multivariate time series. Experimental results on regression and classification tasks show that the proposed method outperforms traditional methods and existing deep learning models in multiple evaluation indicators, verifying its superiority and applicability in time series data mining.

Through ablation experiments, the role of each module in the model is analyzed, and the key contribution of the reasonable design of adaptive adjacency matrix, graph neural network, and multi-head attention mechanism to the model performance is further proved. The visual experimental results show the excellent ability of the model in capturing the relationship between variables and the key positions of time series and intuitively reflect the generalization and robustness of this method in different scenarios. These

research results not only expand the technical boundaries of time series data mining but also provide effective solutions to complex problems in practical applications.

Future research can further explore the adaptability of this method in larger data sets and more practical scenarios, such as weather forecasting, financial market analysis, and medical data diagnosis. At the same time, combining unsupervised learning, self-supervised learning, and other methods will help enhance the performance of the model in scenarios where labels are scarce. In addition, further optimizing the computational efficiency and memory usage of the model and improving its deployment capabilities in edge devices and real-time tasks are also important directions worthy of attention in the future.

References

[1] Choi, K., Yi, J., Park, C., & Yoon, S. (2021). Deep learning for anomaly detection in time-series data: Review, analysis, and guidelines. IEEE access, 9, 120043-120065.

[2] Tuli, S., Casale, G., & Jennings, N. R. (2022). Tranad: Deep transformer networks for anomaly detection in multivariate time series data. arXiv preprint arXiv:2201.07284.

[3] Shiri, F. M., Perumal, T., Mustapha, N., & Mohamed, R. (2023). A comprehensive overview and comparative analysis on deep learning models: CNN, RNN, LSTM, GRU. arXiv preprint arXiv:2305.17473.

[4] Platonov, O., Kuznedelev, D., Diskin, M., Babenko, A., & Prokhorenkova, L. (2023). A critical look at the evaluation of GNNs under heterophily: Are we really making progress?. arXiv preprint arXiv:2302.11640.

[5] Niu, Z., Zhong, G., & Yu, H. (2021). A review on the attention mechanism of deep learning. Neurocomputing, 452, 48-62.

[6] Arshad, M., Jaskani, F. H., Sabri, M. A., Ashraf, F., Farhan, M., Sadiq, M., & Raza, H. (2021). Hybrid machine learning techniques to detect real time human activity using UCI dataset. EAI Endorsed Transactions on Internet of Things, 7(26), e1-e1.

[7] Kontopoulou, V. I., Panagopoulos, A. D., Kakkos, I., & Matsopoulos, G. K. (2023). A review of ARIMA vs. machine learning approaches for time series forecasting in data driven networks. Future Internet, 15(8), 255.

[8] Xue, L., Liu, Y., Xiong, Y., Liu, Y., Cui, X., & Lei, G. (2021). A data-driven shale gas production forecasting method based on the multi-objective random forest regression. Journal of Petroleum Science and Engineering, 196, 107801.

[9] Sun, Y., Ding, S., Zhang, Z., & Jia, W. (2021). An improved grid search algorithm to optimize SVR for prediction. Soft Computing, 25, 5633-5644.

[10] Nosouhian, S., Nosouhian, F., & Khoshouei, A. K. (2021). A review of recurrent neural network architecture for sequence learning: Comparison between LSTM and GRU.

[11] Ahmed, S., Nielsen, I. E., Tripathi, A., Siddiqui, S., Ramachandran, R. P., & Rasool, G. (2023). Transformers in time-series analysis: A tutorial. Circuits, Systems, and Signal Processing, 42(12), 7433-7466.