Comparative analysis of RC and MTGNN models on Multidimensional time-series prediction

Lina Chen
WuhanUniversity
Wuhan,China
2019302080163@whu.edu.cn

Jie Cao
WuhanUniversity
Wuhan,China
2019300001077@whu.edu.cn

Qiuyuan Han
Beijing Normal University
Beijing,China
2658164762@qq.com

Jin Shang
WuhanUniversity
Wuhan,China
2021202080041@whu.edu.cn

Shuofan Zou
Beijing Normal University At Zhuhai
Zhuhai,China
202011059020@mail.bnu.edu.cn

Mengmeng Ye
Communication University of China
Beijing,China
1178834687@qq.com

Ling Ding
WuhanUniversity
Wuhan,China
2020302111333@whu.edu.cn

Teoh Teik Toe
NTU Business AI Lab
Nanyang Technological University
Singapore
ttteoh@ntu.edu.sg

Abstract—**Weather prediction is a very classic time series problem, the most commonly used neural network model is Reservoir Computing(RC) when solving such problems. MTGNN is a newly proposed Spatial-Temporal Neural Network using Graph structure learning layer to capture the Non-Euclidean Structure automatically and building relationship between data in different location. This paper is an attempt to apply Reservoir Computing and MTGNN to the weather prediction, so as to analyze and compare differences between them while addressing multidimensional time series problem.**

Keywords—Reservoir Computing, Graph Neural Network, weather prediction,MTGNN

I. INTRODUCTION

With the development of artificial intelligence, people begin to realize that artificial neural network has self-learning ability, rapid optimization computing ability, the strength to approximate any nonlinear function, great fault tolerance and other advantages. These characteristics make it serve as a perfect means in practical situations where prediction is needed[1].

Recent advances have shown that a machine learning method called Reservoir Computing(RC) has high efficiency and precision in processing complex data. RC is a novel neural network to replace recurrent neural network(RNN). This network proposed a reservoir layer composed of randomly generated node with multiple cycles, this allows both back and forward propagation happened in the same stage[2][3]. These nodes are generated randomly and kept remain, only the output layer needs to be trained, which makes training much faster than traditional methods[4]. Compared with recurrent neural network, the biggest advantage of RC is that it can simplify the network training process, solve the difficulties that the traditional recurrent neural network structure is hard to determine and training algorithm is too complex, and also overcome the problem of memory gradual elimination of the recurrent network.

Traditional weather forecasting, using Recurrent Neural Network (RNN), Long Short-Term Memory Network (LSTM)[6] and Temporal Convolutional Network (TCN)[12] focusing on time series at one place, can reach a state-of-art performance. These methods, however, cannot extract special features between different time series which is vital in geo-based task.

Spatial-Temporal Residual Networks is a complex network combined both spatial and temporal network designed to predict data with these patterns e.g., traffic-flow, weather data and geographic data.

MTGNN [7] is a newly proposed Spatial-Temporal Neural Network using Graph structure learning layer to capture the Non-Euclidean Structure automatically and building relationship between data in different location.

At the same time, Graph Neural Network(GNN) is also a new model with high capacity in reasoning and explanatory power. Furthermore, its end-to-end learning mechanism will greatly improve the efficiency when processing data[7].It was first proposed in 2009 by Franco Scarselli and other scholars[5]. In recent years, GNN has a battery of applications in the field such as social networks, knowledge graphs, recommendation systems[9] and even life sciences[8].In the model of GNN, a recurrent neural network is used to achieve a state transition that allows cycles, and the contractivity of the state dynamics assures the stability of the recursive encoding process[10].

Traditional weather prediction is mainly carried out through satellite cloud map, statistics and other methods. Such method still relies a lot on manual work with great uncertainty. At present, some scholars have proposed to apply BP neural network, recurrent neural network (RNN) in deep network, and long short-term memory network (LSTM)[6] to weather prediction. However, these algorithms are faced with the dilemma of excessive iterations and massive consumption of computational resources. Besides, due to the complexity of the weather forecasting itself and too many factors and indicators needed to be considered when predicting, the accuracy of the current weather forecasting model is not ideal enough. The rainfall, temperature and air pressure of the area should be considered simultaneously in the weather prediction analysis**.** We should not only research their various components as a univariate process, but also explore the relationship between the various components and the changing regulation to forecast and control the time series, which is the multivariate time series analysis[7].Considering multiple variables and dependencies between pairs of variables in weather prediction, GNN can obtain the interrelationship between the variables through the powerful acquisition of information and learning ability. On the other hand, RC is very suitable to address the time series analysis-prediction problem[11].

Therefore, this paper aims to predict the weather simultaneously with both Reservoir Computing and Graph Neural Network, compare the prediction results of the two models and explore the optimal model of the multivariate time series problem.

II. Methodology

A. Data cleaning in data mining

The data of this study are from Kaggle, which is an online platform for data mining and prediction competition. Therefore the data are more accurate, greater covered, and more suitable for model training compared to other data source. The data set covers various weather attributes of 30 US and Canadian Cities, as well as 6 Israeli cities from December, 2012 to December, 2017. The indicators are composed of the following parts: humidity, pressure, temperature, wind direction, and wind speed. For the sake of model effect, the study utilize the temperature data from 2012 to 2017.

After the data selection, we cleaned the data be used :

- Handling of invalid and missing values: Due to the error in recording and understanding, there are some invalid and missing values in the data, for this type of value, we have taken the following approach:
    - Estimation, based on the filling of other data of the object, is estimated by logical inference.
    - Variable deletion, if a variable has a lot of missing values and the variable is not particularly important for the problem being studied, you can consider deleting the variable.
- Detection and elimination of duplicate records: Records with exactly the same attribute values in the data, detect whether the records are duplicate by judging whether the attribute values between the records are equal, and merge or clear the duplicate records.
- Standardizing data: the data in the study equals (the initial data -mean)/std. In the formula, mean means the average of the initial data, and std means the standard deviation of initial data.

B. Model Construction

RC was developed from the cornerstone laid by echo state network (ESN) and liquid state machine (LSM) proposed by Jaeger[13] and Maass[14] respectively. It solves many problems of RNN, such as the large amount of training, complex training algorithm, memory fading, gradient disappearance and so on. As shown in Figure 1, RC is composed of input layer, reservoir and readout layer. Suppose that the input layer has M nodes, the reservoir has N neurons and the readout layer has L output units. The reservoir is a large-scale recursive sparse network generated randomly to replace the hidden layer of the traditional neural network to deal with the input sequence of multiple time scales. The network will not change after the generation. The output $\boldsymbol{x}(n)$ of the neurons in the reservoir is linearly combined into the output $\boldsymbol{y}(n)$ in the readout layer. The preset neuron connection weight W and input weight W_{in} will not be changed, so only the output weight W_{out} needs to be trained. Therefore, RC greatly simplifies the training process of the network, which is better than the traditional gradient descent training method.

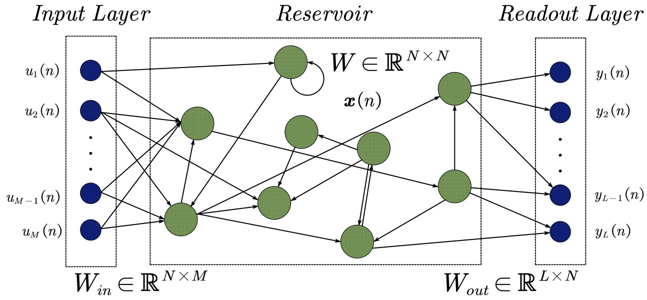

Fig.1. The general structure of Reservoir Computing.

The state equations and output equations of RC are:

$$\boldsymbol{x}(t+1) = \tanh(W_{in}\boldsymbol{u}(t) + W\boldsymbol{x}(t)) \quad (1)$$

$$\hat{\boldsymbol{y}}(t+1) = W_{out}[\boldsymbol{x}(t+1), \boldsymbol{y}(t), \boldsymbol{u}(\boldsymbol{t}+1)] + W_{bias}^{out} \quad (2)$$

where W_{bias}^{out} represents the bias of the output.

The basic idea of Reservoir Computing is to map the low-dimensional input into the high-dimensional dynamic space of the reservoir and transform the problem into a linear one. In the high-dimensional space, its complex internal state is used to deal with the low-dimensional linear combination. Therefore, RC has greater advantages in the description of nonlinear chaotic time series.Because of the special advantages brought by its own structure, RC is widely used in dynamic pattern classification, autonomous sine generation, chaotic time series prediction[13] and many other domains.

The performance of RC is mainly determined by the hyper-parameters of the reservoir, which is the core structure of the model. These hyper-parameters include the internal connection weight spectrum radius SR of the reservoir, which is the eigenvalue with the largest absolute value of the connection weight matrix W ; Reservoir scale N and reservoir input unit scale IS , which is used to scale the input nonlinear signal to a certain extent; The sparsity SD of the reservoir is used to represent the proportion of non-zero elements in the internal connection weight matrix W .Among all these parameters, N has a more significant impact on the performance of the model than other parameters.

MTGNN[7], proposed by Wu (2020), is the first general graphical neural network specially designed for multivariate time series prediction. It is one of the most state-of-art ST-GNN methods at present. This model improves Graph WaveNet by adding an adaptive learning layer. This Graph Learning Layer solves the problem that the previous GNN methods have to rely on the external input graph structure. In general, MTGNN has built a processing spindle with the intersection of temporal and spatial convolution layers, and added an extra learning layer, which can better capture the spatio-temporal dependence between multi-dimensional time series data. It's main structure is shown in Figure 2.

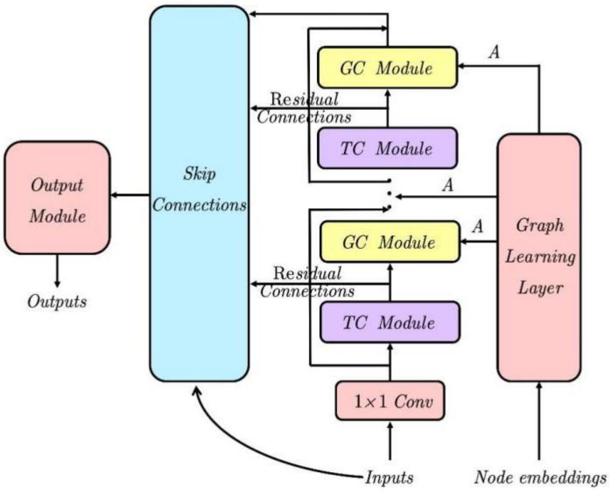

Fig.2. MTGNN model diagram.

Firstly, the adjacency matrix of its graph structure is extracted from the input time series information to obtain the uni-directional relationships of variables. This method can be realized through the following formulas.

$$M_1 = \tanh(\alpha E_1 \Theta_1) \tag{3}$$

$$M_2 = \tanh(\alpha E_2 \Theta_2) \tag{4}$$

$$A = \mathrm{Re}LU(\tanh(\alpha(M_1 M_2{}^T - M_2 M_1{}^T))) \tag{5}$$

where E_1 and E_2 are matrices randomly initialized that include the information of nodes embedding. Θ_1 and Θ_2 are model parameters, and α controls the saturation rate of the $\mathrm{Re}LU$ function. Formula 5 calculates the asymmetric information of the adjacency matrix, in which the effect of the adjacency matrix can be regularized by using $\mathrm{Re}LU$ activation.

And then the nodes' neighborhood information is aggregated in the GC model to obtain the spatial dependence between them. Compared with the graph structure that will not be changed in the previous model, the parameterized Graph Learning Layer and Graph Convolution Modul enable the hyper-parameters and graph structure to be jointly optimized through gradient descent. Then, in the Temporal Convolution Module with Dialated Inception Layer, which are used to process long time series data, the learning algorithm of finding local optimization based on course learning and segmenting multivariable time series data into multiple sub segments, is adopted. This algorithm is mainly used to reduce the space complexity. The structures of GC Module and TC Module are displaced below in Figure 3.

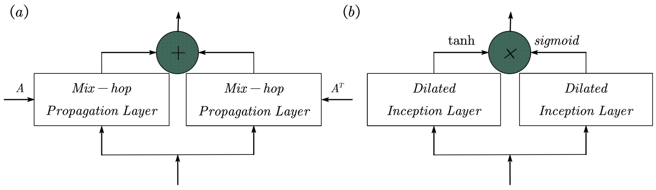

Fig.3. The structure of : (a) TC Module, (b) GC Module.

In addition, the author sets up the skip connection layer to splice module and normalize the information.

III. MODEL EVALUATION

After the prediction of time series data, we analyze the prediction results. In order to evaluate the performance of RC model and MTGNN, RMSE(Root Mean Square Error) is used as indicators to measure the prediction effect of the models. RMSE have the following forms:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2} \tag{6}$$

Where $\hat{y}_i$ is the predicted value and y_i is the actual value. These two indicators range from 0 to infinity. The greater the error, the greater the value. When the predicted value is completely consistent with the real value, it is equal to 0, that is, the perfect model.

TABLE I. RSME COMPARISON

	RC	MTGNN
temperature	1.0044	0.4
humidity	1.2025	0.565
pressure	1.1716	0.553
wind-direction	1.1456	0.751
wind-speed	1.0718	0.696

The following figures compares the predicted results with the actual situation, in which red lines represent the real data, and the blue lines represent the predicted results.

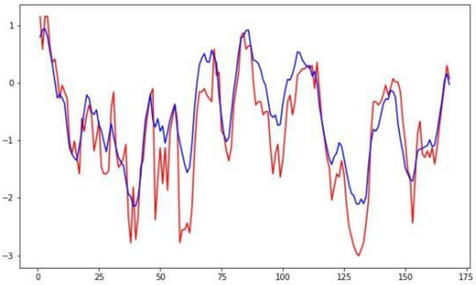

Fig.4. Humidity predicted by RC.

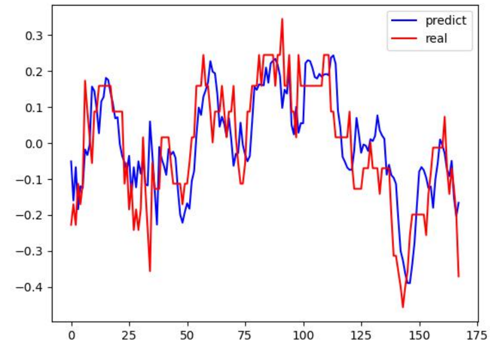

Fig.5. Humidity predicted by MTGNN.

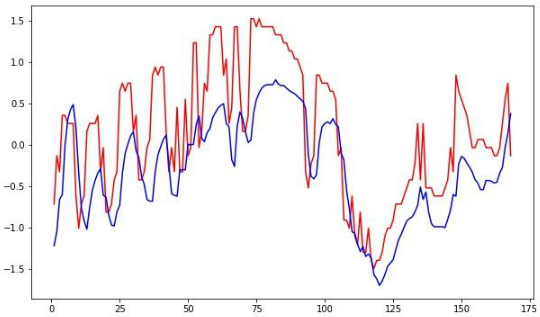

Fig.6. Pressure predicted by RC.

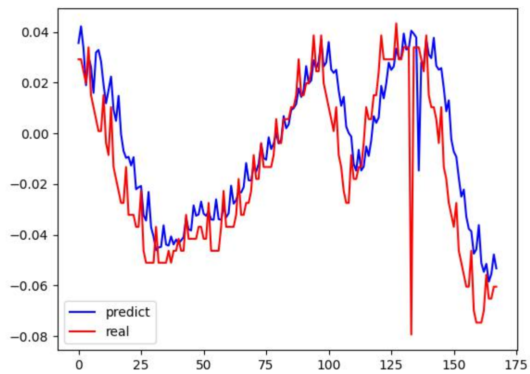

Fig.7. Pressure predicted by MTGNN.

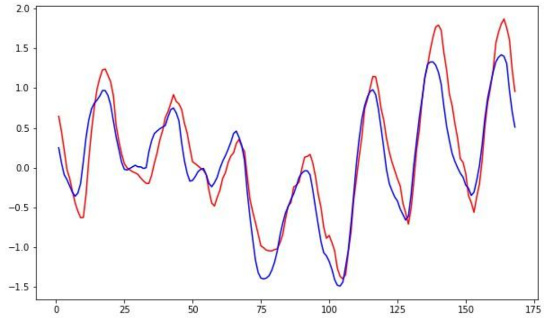

Fig.8. Temperature predicted by RC.

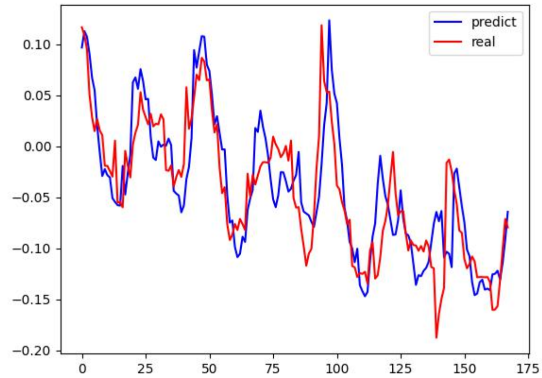

Fig.9. Temperature predicted by MTGNN.

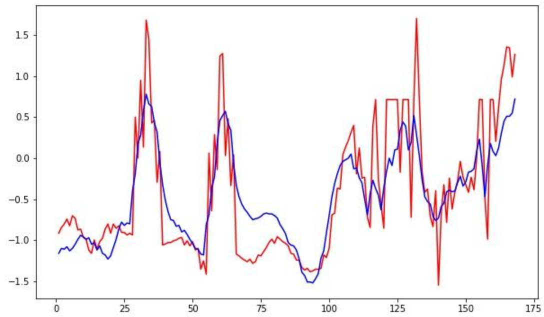

Fig.10. Wind direction predicted by RC.

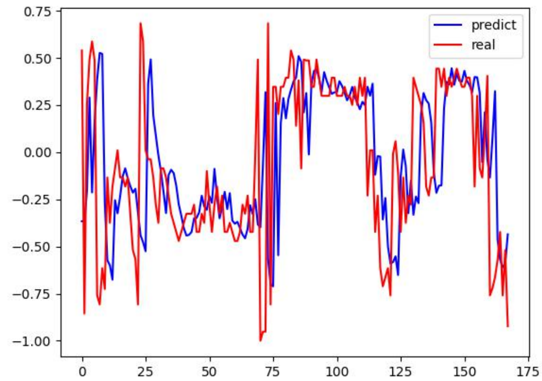

Fig.11. Wind direction predicted by MTGNN.

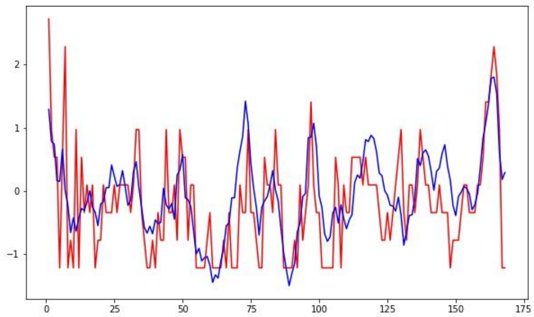

Fig.12. Wind speed predicted by RC.

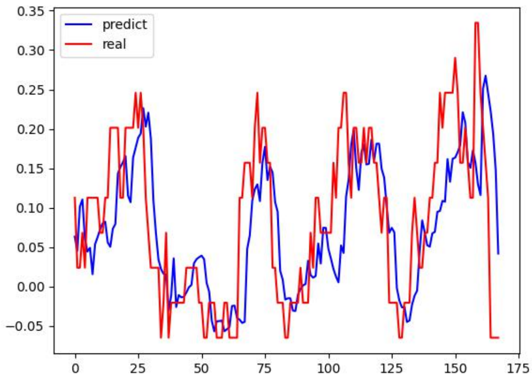

Fig.13. Wind speed predicted by MTGNN.

IV.CONCLUSION

This paper compares the capability of RC and MTGNN in time series prediction. On the public dataset of kaggle, we trained the model based on the weather data of 36 cities, calculated the error value of the model. From the results, we can find that MTGNN is significantly better than RC in time series prediction. It is speculated that the possible reason is that MTGNN uses multiple indicators (FEATURES) as nodes of the graph for prediction, which enhances the stability of the model in the face of complex weather conditions. In contrast, RC can only transform multidimensional time series into matrices when predicting them into the model, ignoring the correlation between nodes.

By comparing these two models, we can tentatively conclude that MTGNN is more suitable for predicting multidimensional time series. Based on this extension, future large-scale spatio-temporal data sets can also be predicted using MTGNN, such as urban traffic, air quality monitoring, etc. At this time, the model not only takes into account the change of data over time, but also the influence of other relevant factors, making the model more accurate and with wide application prospects.

REFERENCES

[1] MOON S H, KIM Y H, LEE Y H, et al. Application of machine learning to an early warning system for very short term heavy rainfall ［J］. Journal of Hydrology, 2019, 568:1042 － 1054.

[2] WENG T F, YANG H J, GU C G, et al. Synchronization of chaotic systems and their machine-learning models ［J］. Physical Ｒeview. E: Statistical Physics Plasmas Fluids ＆ Related Interdisciplinary Topics, 2019, 99(4) : 042203.

[3] CARROLLTL. Using reservoir computers to distinguish chaotic signals ［J］. Physical Review. E: Statistical Physics Plasmas Fluids ＆ Ｒelated Interdisciplinary Topics, 2018, 98(5) : 52209.

[4] Wen-Yan Liang, Shi-Rong Xu, Li Jiang, Xin-Hong Jia, Jia-Bing Lin, Yu-Lian Yang, Li-Ming Liu, Xuan Zhang,Design of parallel reservoir computing by mutually-coupled semiconductor lasers with optoelectronic feedback,Optics Communications,Volume 495,2021,127120,ISSN 0030-4018

[5] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner and G. Monfardini.The Graph Neural Network Model ［J］. IEEE Transactions on Neural Networks, vol. 20, no. 1, pp. 61-80, Jan. 2009, doi: 10.1109/TNN.2008.2005605.

[6] ZIA T, ZAHID U. Long short-term memory recurrent neural network architectures for Urdu acoustic modeling[J]. International Journal of Speech Technology, 2019, 22(1): 21-30.

[7] Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, Xiaojun Chang, and Chengqi Zhang. 2020. Connecting the Dots: Multivariate Time Series Forecasting with Graph Neural Networks. In 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '20), August 23–27, 2020, Virtual Event, USA. ACM, New York, NY, USA, 11 pages.

[8] Zhang Z, Chen L, Zhong F, Wang D, Jiang J, Zhang S, Jiang H, Zheng M, Li X. Graph neural network approaches for drug-target interactions. Curr Opin Struct Biol. 2022 Jan 21;73:102327. doi: 10.1016/j.sbi.2021.102327. Epub ahead of print. PMID: 35074533.

[9] Shiwen Wu, Fei Sun,et al.Graph Neural Networks in Recommender Systems: A Survey arXiv:2011.02260

[10] F.M. Bianchi, Claudio Gallicchio, Alessio Micheli,Pyramidal Reservoir Graph Neural Network,Neurocomputing,Volume 470,2022,Pages 389-404,ISSN 0925-2312

[11] Xinyu Han, Yi Zhao,Reservoir computing dissection and visualization based on directed network embedding,Neurocomputing,Volume 445,2021,Pages 134-148,ISSN 0925-2312,

[12] Lea, Colin, Flynn, Michael D., Vidal, Ren ́e, Reiter, Austin, and Hag er, Gregory D. Temporal convolutional networks for action segmentat ion and detection. In CVPR, 2017

[13] Jaeger, H. "The echo state approach to analyzing and training recurrent neural networks;(Tech. Rep. No. 148). Bremen." German National Research Center for Information Technology (2001).

[14] Maass, Wolfgang, Thomas Natschläger, and Henry Markram. "Real-time computing without stable states: A new framework for neural computation based on perturbations." Neural computation 14.11 (2002): 2531-2560.