Gated Res2Net for Multivariate Time Series Analysis

1st Chao Yang
School of Information Science and Engineering
Ocean University of China
Qingdao, China
yangchao249@hotmail.com

2nd Mingxing Jiang
School of Information Science and Engineering
Ocean University of China
Qingdao, China
jiangmx@ouc.edu.cn

3rd Zhongwen Guo
School of Information Science and Engineering
Ocean University of China
Qingdao, China
guozhw@ouc.edu.cn

4th Yuan Liu
School of Information Science and Engineering
Ocean University of China
Qingdao, China
liuyuan6787@stu.ouc.edu.cn

Abstract—**Multivariate time series analysis is an important problem in data mining because of its widespread applications. With the increase of time series data available for training, implementing deep neural networks in the field of time series analysis is becoming common. Res2Net, a recently proposed backbone, can further improve the state-of-the-art networks as it improves the multi-scale representation ability through connecting different groups of filters. However, Res2Net ignores the correlations of the feature maps and lacks the control on the information interaction process. To address that problem, in this paper, we propose a backbone convolutional neural network based on the thought of gated mechanism and Res2Net, namely Gated Res2Net (GRes2Net), for multivariate time series analysis. The hierarchical residual-like connections are influenced by gates whose values are calculated based on the original feature maps, the previous output feature maps and the next input feature maps thus considering the correlations between the feature maps more effectively. Through the utilization of gated mechanism, the network can control the process of information sending hence can better capture and utilize the both the temporal information and the correlations between the feature maps. We evaluate the GRes2Net on four multivariate time series datasets including two classification datasets and two forecasting datasets. The results demonstrate that GRes2Net have better performances over the state-of-the-art methods thus indicating the superiority.**

Index Terms—**multivariate time series analysis, gated mechanism, Res2Net, deep learning**

I. INTRODUCTION

Multivariate time series data are ubiquitous in many real-world applications. Thus, the analysis of multivariate time series data is meaningful and important [1]. Multivariate time series data analysis includes classification and forecasting. The applications of classification can be seen in human activity recognition [2], equipment maintenance planning [3], heart disease prediction [4], etc., while the applications of forecasting can be seen in air quality forecasting [5], renewable energy management [6], power consumption prediction [7], etc. The

characteristics of multivariate time series data are the natural temporal ordering and the correlations between the variables. Therefore, how to effectively extract and utilize the temporal features and the correlations between variables are essential for multivariate time series analysis.

Traditional machine learning algorithms based on statistics theory have been widely used in multivariate time series analysis such as K- Nearest Neighbor (KNN), Dynamic Time Warping (DTW), Support Vector Machine (SVM), etc. For these algorithms, feature engineering is vital for achieving satisfied performance. Because the quality of feature engineering highly depends on the expertise hence resulting in challenges and time expenses.

With the advent of big data era, more and more multivariate time series data are accessible for training which leads to the popularity of deep learning models. Deep learning models allow automatic feature extraction and end-to-end learning hence are less dependent on feature engineering which is an advantage compared with traditional machine learning algorithms. Recurrent Neural Network (RNN) is a typical kind of deep learning model and has been widely used in multivariate time series analysis tasks [8]–[10]. Compared with RNN, Convolutional Neural Network (CNN) is often used in computer vision tasks but less common for time series analysis. CNN can be fully parallelized during training to better exploit the GPU hardware. Thus, recent works have tried to apply CNN in sequence modeling [11] and time series classification [12].

The component of a model for feature extraction is called the backbone network whose output is the feature map of the input data. Famous backbone networks includes VGG [13], Inception [14], ResNet [15], etc. Recent studies have shown that typical CNN backbone networks which are firstly designed for image processing such as ResNet and FCN [16] are capable for time series analysis [17]. The neurons in CNN respond to stimuli in a restricted region, often called receptive filed, of

the visual field hence CNN is capable for feature extraction in a particular region. Since CNN learns the representations for fixed size contexts, for a better feature extraction ability, it is necessary to design a network architecture to achieves multi-scale feature representations.

Res2Net [18] is a recently developed backbone CNN. It improves the multi-scale representation ability at granular level by utilizing multiple available receptive fields. In a Res2Net module, groups of filters are connected in a hierarchical residual-like style for multi-scale receptive fields which is conducive for global and local feature extraction and the information interaction between the filters. Therefore, it achieves the state-of-the-art performance in computer vision tasks such as object detection [18] and person re-identification [19]. Thanks to the performance, it inspires implementing Res2Net for multivariate time series analysis. However, in Res2Net, the whole previous output feature maps are sent to the next group of filters directly which ignores considering how much information from the previous output should be sent. Or in other words, the model lacks the control on the information interaction between different groups of filters. For multivariate time series analysis, as the feature maps contain information from different variables, ignoring the correlations of the feature maps may lead to lacking the considerations on the inter-relationship of the variables. Thus, effective control on the information interaction process is important and meaningful.

In this paper, we propose a backbone CNN based on the gated mechanism and Res2Net called Gated Res2Net (GRes2Net). Compared with Res2Net, we utilize gates in the connection between the previous output feature maps and the next input feature maps in order to consider the correlations between different feature maps. Instead of simply executing addition in the residual-like connections, the gates whose values are calculated based on the original feature maps, the previous output feature maps and the input feature maps decide how much information will be sent through the connection. In this way, GRes2Net can control the process of information interaction and utilize the latent information more effectively for a better performance.

The contributions of our paper are in two aspects:

First, we propose a novel backbone CNN called GRes2Net for multivariate time series analysis. I.e., it can be used for both classification and forecasting. The proposed backbone achieves effectively considering correlations between the variables and extracting multi-scale temporal features of multivariate time series data. Moreover, the proposed backbone can be easily implemented in the existing CNN models or RNN-CNN cascade models.

Second, we conduct experiments on four multivariate time series datasets including two multivariate time series classification datasets and two multivariate time series forecasting datasets to assess the proposed backbone. We compare our model with other alternatives. The results demonstrate that our model outperforms the state-of-the-art methods hence indicating the advantage of our model.

The rest of this paper is organized as follows. Section II presents related studies and techniques. Section III introduces the structure of GRes2Net. Section IV gives the details of the four datasets and the corresponding state-of-the-art method, the evaluation standards, the experiment settings and the results of the experiments. Section V gives the conclusion and future work.

II. RELATED WORKS

A. Traditional Machine Learning for Multivariate Time Series Analysis

Traditional machine learning methods are popular for multivariate time series analysis. Prevailing approaches such as DTW, KNN and SVM are applicable for both classification and forecasting [20]–[23]. SVM is widely used in data mining tasks [24] and has been improved in many aspects [25], [26]. Models based on the combination of KNN and DTW suggests strong performances [27], [28]. Autoregressive Moving Average (ARMA) is a typical method for exploring the rational spectrum of stationary random processes and has been widely used for analyzing stationary time series [31]. Ensemble learning methods such as Random Forest (RF) and Gradient Boosting Machine (GBM) aggregate multiple classifiers into one model and generally result in better performance than single classifier. Related methods are commonly used in time series analysis [29] and data science competition [30].

B. Gated Mechanism

Gated mechanism is commonly used in RNN. Because the original RNN suffers for gradient vanishing or explosion. Besides, it is likely to ignore long-term dependencies [32], [33]. Thus, in order to address these problems, gated mechanism is applied. Long Short-Term Memory (LSTM) [34] and Gated Recurrent Unit (GRU) [35] are two well-known variants of RNN based on the thought of gated mechanism. Taking LSTM as an example, specifically, the computation of LSTM is defined by the equations (1)-(6):

$$g^{(k)} = \tanh\left(W^{gx}x_k + W^{gh}h_{(k-1)} + b^g\right) \quad (1)$$

$$i^{(k)} = \sigma\left(W^{ix}x_k + W^{ih}h_{(k-1)} + b^i\right) \quad (2)$$

$$f^{(k)} = \sigma\left(W^{fx}x_k + W^{fh}h_{(k-1)} + b^f\right) \quad (3)$$

$$o^{(k)} = \sigma\left(W^{ox}x_k + W^{oh}h_{(k-1)} + b^o\right) \quad (4)$$

$$s^{(k)} = g^{(k)} \odot i^{(k)} + s^{(k-1)} \odot f^{(k)} \quad (5)$$

$$h_{(k)} = \tanh\left(s^{(k)}\right) \odot o^{(k)} \quad (6)$$

where W^{gx}, W^{gh}, W^{ix}, W^{ih}, W^{fx}, W^{fh}, W^{ox}, W^{oh}, b^g, b^i, b^f and b^o are parameters to be learnt. $i^{(k)}$, $o^{(k)}$ and $f^{(k)}$ are three gates which determine how much the previous information will influence the current output and the current hidden state. Concretely, $f^{(k)}$ determines how much information from the previous hidden states is going to be abandoned. $i^{(k)}$ determines how much information will be sent into the current hidden states and $o^{(k)}$ determines the output of a single LSTM

cell. Gated mechanism is helpful in two aspects: First, it is able to avoid gradient vanishing. Second, it allows the model choose whether the previous information is supposed to be memorized or abandoned hence it allows the recurrent layer to capture long-term dependencies more easily. The benefits of the gated mechanism inspires implementing it in other networks.

C. Res2Net

Res2Net is a recently proposed backbone CNN. To achieve multi-scale available receptive fields, the filters with n channels are replaced with s groups of filters and each group of filters are with w channels. (To avoiding information loss, generally $n=s\times w$.) Groups of filters are connected in a hierarchical residual-like style. The channels of input feature maps are expanded by convolutional layers and then divided into several groups. A group of filters firstly extracts features from a group of input feature maps. The output feature maps are then sent to the next group of filters along with another group of input feature maps. This process repeats several times until all input feature maps are processed.

Specifically, in a single Res2Net module, channel expansion is firstly executed by implementing convolutional layer. Then, the original feature maps are obtained, denoted by $\mathbf{X}$. After that, $\mathbf{X}$ is evenly divided into several groups which are denoted by $\mathbf{x}_i$, where $i \in \{1,2,3\ldots.s\}$. Each group is a feature map subset which has the same spatial or temporal size and $1/s$ of channels compared with $\mathbf{X}$. The convolution is denoted by $\mathbf{K}_i()$, $\mathbf{y}_i$ is the output of $\mathbf{K}_i()$. Then, the $\mathbf{y}_i$ can be written in equation (7):
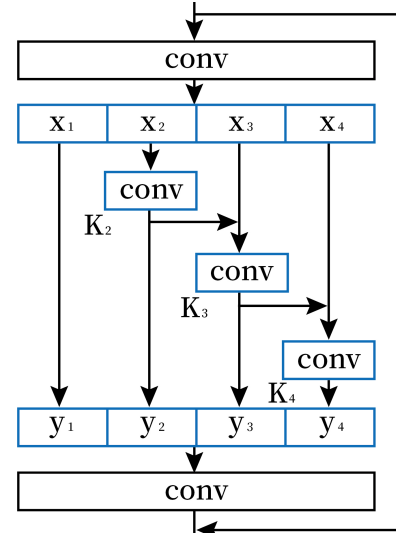
$$\mathbf{y}_i = \begin{cases} \mathbf{x}_i & i = 1 \\ \mathbf{K}_i\left(\mathbf{x}_i\right) & i = 2 \\ \mathbf{K}_i\left(\mathbf{x}_i + \mathbf{y}_{i-1}\right) & 2 < i \leqslant s. \end{cases} \tag{7}$$

All the outputs are concatenated and then fed into convolutional layers for channel compression and information fusion. In this way, Res2Net achieves multi-scale receptive fields thus allowing multi-scale feature representations. The architectures based on Res2Net backbone have achieved the state-of-the-art performance in several computer vision tasks [18]. However, the original structure of Res2Net suffers for controlling the information flow between groups. Hence, we considering implementing gated mechanism in Res2Net to fix that problem.

III. GATED RES2NET

The structure shown in Fig. 1(a) is the original Res2Net module while the structure shown in Fig. 1(b) is the proposed GRes2Net module. Fig. 1 shows the difference between Res2Net and GRes2Net. Code is available at: https://github.com/ChaoYang93/GraduatePaper/blob/master/GRes2Net.py. In our GRes2Net, instead of sending the whole previous output feature maps to the next group of filters along with the another group of input feature maps, gates are utilized to determine how much information should be sent. We use the same denotations in the previous section. Moreover, $\mathbf{g}_i$ denotes the

(a) Res2Net
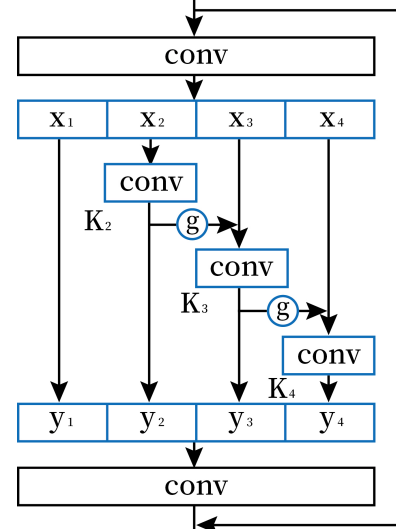
(b) GRes2Net

Fig. 1: Comparison between the Res2Net module and the proposed GRes2Net module (assuming s=4)

gate, then the $\mathbf{y}_i$ of GRes2Net can be presented as equation (8):

$$\mathbf{y}_i = \begin{cases} \mathbf{x}_i & i = 1 \\ \mathbf{K}_i\left(\mathbf{x}_i\right) & i = 2 \\ \mathbf{K}_i\left(\mathbf{x}_i + \mathbf{g}_i \cdot \mathbf{y}_{i-1}\right) & 2 < i \leqslant s \end{cases} \tag{8}$$

where for the $\mathbf{x}_i$, $\mathbf{g}_i$ is calculated as equation (9):

$$\mathbf{g}_i = \tanh\left(a\left(\text{concat}\left(a(\mathbf{X}), a(\mathbf{y}_{i-1}), a\left(\mathbf{x}_i\right)\right)\right)\right). \tag{9}$$

Generally, a can be fully connected layers or convolutional layers. The value of the $\mathbf{g}_i$ is calculated considering the original feature maps $\mathbf{X}$, the next input feature maps $\mathbf{x}_i$ and the previous output feature maps $\mathbf{y}_{i-1}$. An illustration for calculation can be seen in Fig. 2.
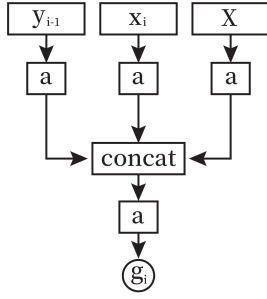
The following computations are the same as Res2Net. All

Fig. 2: The calculation process of $\mathbf{g}_i$

the outputs are concatenated and go through a convolutional layer for channel compression and information fusion. Convolutional layer whose kernel size equals to 1 is used for channel expansion and compression which is beneficial for information fusion without too many extra parameters.

For multivariate time series analysis, GRes2Net module uses one-dimension convolutional layers. Each group of the feature maps contains temporal information from different variables. Through the connection between the groups of filters, multi-scale receptive fields are achieved which is beneficial for both global and local temporal features extraction. Besides, based on the gated mechanism, the correlations of different feature maps are also considered in a more effective way compared with Res2Net. The model is able to control the information interaction process which is helpful for better utilizing the correlations between different feature maps. Thus, GRes2Net is capable for multivariate time series analysis.

IV. EXPERIMENTS

A. Datasets Description and the Corresponding State-of-the-art Method

1) EGG Dataset: EGG dataset is a multivariate time series classification dataset which consists of 64 attributes. The maximum length of a sequence in the dataset is 256. The dataset can be downloaded from [36]. It has predefined training and validation sets. The training set contains 600 sequences while the validation set contains the same number of sequences. LSTM-FCN, MLSTM-FCN, ALSTM-FCN and MALSTM-FCN [37] have been implemented on the dataset for evaluation. Among them, MALSTM-FCN achieves the state-of-the-art performance based on the accuracy.

In order to make sure the comparison is fair and reasonable, we use the same data preprocessing method as [37] and use accuracy to evaluate the performance of our model. We use the same training set to train our model and use the same validation set for evaluation as [37].

2) Occupancy Detection Dataset: Occupancy detection dataset is a multivariate time series classification dataset. This dataset is developed by [38] and can be obtained from [36]. The dataset contains 5 attributes and the maximum length of the sequence in the dataset is 3758. The aim is to detect whether the office is occupied according to the temperature, humidity, light, CO2 and humidity ratio. The

dataset has predefined training and validation sets. LSTM-FCN, MLSTM-FCN, ALSTM-FCN and MALSTM-FCN [37] have been implemented on the dataset and the state-of-the-art performance based on accuracy is achieved by MLSTM-FCN. We use the same data preprocessing method as [37] and use accuracy as the criterion. Moreover, we use the same training set to train our model and use the same validation set for evaluation as [37].

3) Appliances Energy Prediction Dataset: Appliances energy prediction dataset contains experimental data used to create forecasting models of appliances energy use in a low energy building. The dataset contains 19735 measurements. It is proposed by [39] and can be obtained from [36]. The dataset contains 29 attributes including temperature, humidity, weather, etc. In [39], GBM makes the most accurate prediction. The recent state-of-the-art performance is achieved by Multi-Layer Perceptron (MLP) proposed in [40]. Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE) and R Squared(R^2) are used for evaluation. The details of the criteria are shown in equations (10)-(13):

$$RMSE = \sqrt[2]{\frac{1}{M} \sum_1^M (y_i - \hat{y}_i)^2} \tag{10}$$

$$MAE = \frac{1}{M} \sum_1^M |y_i - \hat{y}_i| \tag{11}$$

$$MAPE = \frac{100}{M} \sum_1^M \left| \frac{y_i - \hat{y}_i}{y_i} \right| \tag{12}$$

$$R^2 = 1 - \frac{\sum_1^M (y_i - \hat{y}_i)^2}{\sum_1^M (y_i - \bar{y}_i)^2} \tag{13}$$

where M denotes the total number of the samples, y_i denotes the true value, $\bar{y}_i$ denotes the average of the true values and $\hat{y}_i$ denotes the output of the model.
We use the same data preprocessing method and evaluation criteria as [40] for reasonable comparison.

4) Individual Household Electric Power Consumption Dataset: Individual household electric power consumption dataset contains measurements of electric power consumption in one household over a period of almost 4 years. This dataset can be obtained from [36]. It contains 2075259 measurements gathered in a house located in Sceaux (7km of Paris, France) between December 2006 and November 2010 (47 months). The state-of-the-art performance is achieved by ARMA proposed in [41] using RMSE for evaluation. We build our model to predict daily series contains 42 measurements. The result based on ARMA with the same data preprocessing method is given in [41]. Moreover, we use RMSE, MAE, MAPE and R^2 for evaluation. It is worth mentioning that because the specific MAE, MAPE and R^2 are not given in [41], we use N/A instead.

B. Experiment Setting

In our experiment, we use GRes2Net as the backbone CNN for feature extraction. a is convolutional layer in the experiment. The output feature maps go through a global average pooling layer and fully connected layers to learn the mapping between the feature maps and the output. For classification tasks, we use cross-entropy as the lost function while for forecasting tasks we use Mean Squared Error (MSE). We train our model by an Adam optimizer with the initial learning rate 0.001 for totally 500 training epochs. Adam is a commonly used optimizer and generally has the best performance, so we did not try others. After each 100 epochs, the learning rate is adjusted to one tenth of the previous value. The batch size is 32 in the two classification tasks and the batch size is 64 in the two forecasting tasks. Dropout is used to avoid overfitting. The random invalid possibility of the neurons is 0.5. The early stopping is performed based on the validation error. I.e., we store the model that has the best performance on the validation set. Then, we repeat validating on the validation set for 5 times and record the results. To avoid drastic fluctuation of the performance, we calculate the average performance as the final result.

Besides, we implement deep LSTM and Res2Net on the four datasets for comparison. We use the same settings to train LSTM and Res2Net. The model based on Res2Net contains the same number of modules as the model based on GRes2Net for fair comparison. We construct a bidirectional LSTM contains 8 layers. Dropout is also used and the random invalid possibility is the same. Early stopping and 5 times validating are also executed.

We compare our model with the aforementioned state-of-the-art models, deep LSTM, Res2Net and other related models that have been implemented in the previous studies. We implement LSTM, Res2Net and GRes2Net with PyTorch, a famous and popular deep learning framework based on Python. All models are trained with GPU GTX 1060.

C. Experiment Results

Table I shows the classification results. As we can see, GRes2Net outperforms the alternatives on the two multivariate time series classification datasets. It is worth noticing that the previous state-of-the-art models are based on the variant of RNN-CNN cascade model. But original Res2Net and GRes2Net both have a better performance than the previous state-of-the-art models which indicates the advantages of the hierarchical residual-like connections for multivariate time series classification. Moreover, it is likely that replacing the CNN that used in the cascade model with GRes2Net may improve the performance.

Table II and Table III demonstrate that GRes2Net has the best performance on the two multivariate time series forecasting datasets according to the evaluation criteria. Several traditional machine learning algorithms are implemented for forecasting. It is obvious that GRes2Net has a significant improvement compared with the previous state-of-the-art methods. Surprisingly, LSTM is supposed to have a better performance than

TABLE I: Performance Comparison for Classification Tasks

Method	EGG (Accuracy)	Occupancy Detection (Accuracy)
LSTM	71.60	57.99
Res2Net	91.50	81.25
LSTM-FCN	90.67	71.05
MLSTM-FCN	91.00	76.31
ALSTM-FCN	90.67	71.05
MALSTM-FCN	91.33	72.37
GRes2Net	**92.76**	**83.33**

TABLE II: Performance Comparison Based on Appliances Energy Prediction Dataset

Method	RMSE	MAE	MAPE	R^2
GBM	66.65	35.22	38.29	0.57
RF	68.48	31.85	31.39	0.54
SVM	70.74	31.36	29.76	0.52
LSTM	74.41	39.21	41.08	0.24
MLP	59.84	27.28	27.09	0.64
Res2Net	13.98	7.64	10.47	0.97
GRes2Net	**12.84**	**6.99**	**9.74**	**0.98**

MLP on appliances energy prediction dataset while the result is opposite. It is possible that some tricks have been used for constructing MLP in the previous work.

According to the results of the experiments, Res2Net outperforms the state-of-the-art models on all four datasets. As mentioned in the previous section, in a Res2Net module, the previous output features maps are sent to the next group of filters without any restriction. However, for multivariate time series analysis, the drawback of the Res2Net is that, if simply sending all the previous output feature maps to the next group of filters, it is incapable for the model to control how much information is supposed to be sent. Hence, it is likely that the some connections have no or even negative influence for multivariate time series analysis.

To address that problem, gated mechanism is used. The gates play an important role in GRes2Net as it allows the model to control the process of information interactions between the different groups of filters. Concretely, the connections between the groups of filters are influenced by the correlations of the feature maps. Thanks to that, the model is able to determine how much information will be sent for achieving a better performance. In this way, for multivariate time series analysis, not only a multi-scale temporal features are extracted, but the correlations between the feature maps are more effectively considered. It is conducive for accurate multivariate time series analysis. This can be easily seen according to the results of the experiments as the performance of GRes2Net is the best on all four datasets. Besides, as the sizes of the four datasets are quite different and all the experimental results are satisfying, it is convincing to say that GRes2Net is able to perform well among different datasets. To conclude, GRes2Net demonstrates the superiority over Res2Net and other state-of-the-art models for multivariate time series analysis.

TABLE III: Performance Comparison Based on Individual Household Electric Power Consumption Dataset

Method	RMSE	MAE	MAPE	R^2
LSTM	0.61	0.39	41.49	0.72
Res2Net	0.20	0.13	13.13	0.97
ARMA	0.36	N/A	N/A	N/A
GRes2Net	**0.19**	**0.12**	**12.88**	**0.98**

V. CONCLUSION AND FUTURE WORK

In this paper, we propose GRes2Net, a backbone CNN for multivariate time series analysis based on gated mechanism and Res2Net. In our model, instead of sending the whole previous output feature maps to the next input, gates are used to control this process. The value of a single gate is calculated based on the undivided feature maps, the previous output feature maps and the next input feature maps. Then the gate determines how much information to be sent. In this way, our model is able to consider the correlations between the feature maps more effectively compared with Res2Net. Moreover, it is worth mentioning that GRes2Net can be implemented in the existing CNN models or RNN-CNN cascade models for multivariate time series analysis without extra effort. Experiments results indicate that our model achieves more accurate results for both classification and forecasting than the state-of-the-art models hence demonstrating the superiority.

In the future, we will further explore the design of the gates. Concretely, we are going to consider calculating the value of the gates based on different variables. Besides, the original feature maps are divided evenly. This division method is likely to be improved. Thus, we are going to explore a more effective way such as splitting by learning.

ACKNOWLEDGMENT

REFERENCES

[1] E. Keogh and S. Kasetty, "On the need for time series data mining benchmarks: a survey and empirical demonstration," *Data Mining and knowledge discovery*, vol. 7, no. 4, pp. 349–371, 2003.

[2] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, "Human activity recognition on smartphones using a multiclass hardware-friendly support vector machine," in *International workshop on ambient assisted living*. Springer, 2012, pp. 216–223.

[3] C.-L. Liu, W.-H. Hsaio, and Y.-C. Tu, "Time series classification with multivariate convolutional neural network," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 6, pp. 4788–4797, 2018.

[4] H. D. Masethe and M. A. Masethe, "Prediction of heart disease using classification algorithms," in *Proceedings of the world Congress on Engineering and computer Science*, vol. 2, 2014, pp. 22–24.

[5] M. Kolehmainen, H. Martikainen, and J. Ruuskanen, "Neural networks and periodic components used in air quality forecasting," *Atmospheric Environment*, vol. 35, no. 5, pp. 815–825, 2001.

[6] B. Doucoure, K. Agbossou, and A. Cardenas, "Time series prediction using artificial wavelet neural network and multi-resolution analysis: Application to wind speed data," *Renewable Energy*, vol. 92, pp. 202–211, 2016.

[7] T. Le, M. T. Vo, B. Vo, E. Hwang, S. Rho, and S. W. Baik, "Improving electric energy consumption prediction using cnn and bi-lstm," *Applied Sciences*, vol. 9, no. 20, p. 4237, 2019.

[8] B. S. Freeman, G. Taylor, B. Gharabaghi, and J. Thé, "Forecasting air quality time series using deep learning," *Journal of the Air & Waste Management Association*, vol. 68, no. 8, pp. 866–886, 2018.

[9] Z. Che, S. Purushotham, K. Cho, D. Sontag, and Y. Liu, "Recurrent neural networks for multivariate time series with missing values," *Scientific reports*, vol. 8, no. 1, p. 6085, 2018.

[10] P. Malhotra, V. TV, L. Vig, P. Agarwal, and G. Shroff, "Timenet: Pre-trained deep recurrent neural network for time series classification," *arXiv preprint arXiv:1706.08838*, 2017.

[11] J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin, "Convolutional sequence to sequence learning," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2017, pp. 1243–1252.

[12] B. Zhao, H. Lu, S. Chen, J. Liu, and D. Wu, "Convolutional neural networks for time series classification," *Journal of Systems Engineering and Electronics*, vol. 28, no. 1, pp. 162–169, 2017.

[13] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[14] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[15] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[16] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

[17] H. I. Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Muller, "Deep learning for time series classification: a review," *Data Mining and Knowledge Discovery*, vol. 33, no. 4, pp. 917–963, 2019.

[18] S.-H. Gao, M.-M. Cheng, K. Zhao, X.-Y. Zhang, M.-H. Yang, and P. Torr, "Res2net: A new multi-scale backbone architecture," *arXiv preprint arXiv:1904.01169*, 2019.

[19] Z. Cao and H. J. Lee, "Person re-identification based on res2net network," *arXiv preprint arXiv:1910.04061*, 2019.

[20] K.-j. Kim, "Financial time series forecasting using support vector machines," *Neurocomputing*, vol. 55, no. 1-2, pp. 307–319, 2003.

[21] W. Wu, J. Zhou, L. Mo, and C. Zhu, "Forecasting electricity market price spikes based on bayesian expert with support vector machines," in *International Conference on Advanced Data Mining and Applications*. Springer, 2006, pp. 205–212.

[22] T. Górecki and M. Łuczak, "Non-isometric transforms in time series classification using dtw," *Knowledge-based systems*, vol. 61, pp. 98–108, 2014.

[23] R. M. Marcacini, J. C. Carnevali, and J. Domingos, "On combining websensors and dtw distance for knn time series forecasting," in *2016 23rd International Conference on Pattern Recognition (ICPR)*. IEEE, 2016, pp. 2521–2525.

[24] Y. Liu and H. Huang, "Fuzzy support vector machines for pattern recognition and data mining," *International journal of fuzzy systems*, vol. 4, no. 3, pp. 826–835, 2002.

[25] Z. Sun, K. Hu, T. Hu, J. Liu, and K. Zhu, "Fast multi-label low-rank linearized svm classification algorithm based on approximate extreme points," *IEEE Access*, vol. 6, pp. 42 319–42 326, 2018.

[26] G. Cauwenberghs and T. Poggio, "Incremental and decremental support vector machine learning," in *Advances in neural information processing systems*, 2001, pp. 409–415.

[27] S. Oehmcke, O. Zielinski, and O. Kramer, "knn ensembles with penalized dtw for multivariate time series imputation," in *2016 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2016, pp. 2774–2781.

[28] H.-H. Hsu, A. C. Yang, and M.-D. Lu, "Knn-dtw based missing value imputation for microarray time series data," *Journal of computers*, vol. 6, no. 3, pp. 418–425, 2011.

[29] M. J. Kane, N. Price, M. Scotch, and P. Rabinowitz, "Comparison of arima and random forest time series models for prediction of avian influenza h5n1 outbreaks," *BMC bioinformatics*, vol. 15, no. 1, p. 276, 2014.

[30] S. B. Taieb and R. J. Hyndman, "A gradient boosting approach to the kaggle load forecasting competition," *International journal of forecasting*, vol. 30, no. 2, pp. 382–394, 2014.

[31] P. J. Brockwell and R. A. Davis, *Introduction to time series and forecasting*. springer, 2016.

[32] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.

[33] R. Jozefowicz, W. Zaremba, and I. Sutskever, "An empirical exploration of recurrent network architectures," in *International Conference on Machine Learning*, 2015, pp. 2342–2350.

[34] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[35] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.

[36] M. Lichman, "Uci machine learning repository," http://archive.ics.uci.edu/ml, accessed 2013.

[37] F. Karim, S. Majumdar, H. Darabi, and S. Harford, "Multivariate lstm-fcns for time series classification," *Neural Networks*, vol. 116, pp. 237–245, 2019.

[38] L. M. Candanedo and V. Feldheim, "Accurate occupancy detection of an office room from light, temperature, humidity and co2 measurements using statistical learning models," *Energy and Buildings*, vol. 112, pp. 28–39, 2016.

[39] L. M. Candanedo, V. Feldheim, and D. Deramaix, "Data driven prediction models of energy use of appliances in a low-energy house," *Energy and buildings*, vol. 140, pp. 81–97, 2017.

[40] M. Chammas, A. Makhoul, and J. Demerjian, "An efficient data model for energy prediction using wireless sensors," *Computers & Electrical Engineering*, vol. 76, pp. 249–257, 2019.

[41] P. Chujai, N. Kerdprasop, and K. Kerdprasop, "Time series analysis of household electric consumption with arima and arma models," in *Proceedings of the International MultiConference of Engineers and Computer Scientists*, vol. 1, 2013, pp. 295–300.