Time Series Forecasting Using GRU Neural Network with Multi-lag After Decomposition

Xu Zhang[1], Furao Shen[1(✉)], Jinxi Zhao[1], and GuoHai Yang[2]

[1] National Key Laboratory for Novel Software Technology, Department of Computer Science and Technology, Collaborative Innovation Center of Novel Software Technology and Industrialization, Nanjing University, Nanjing, China
zhangxu0307@163.com, {frshen,jxzhao}@nju.edu.cn
[2] Nanjing Melangy Energy Science and Technology Co. Ltd., Nanjing, China
usedplaneandship@188.com

Abstract. Time series forecasting has a wide range of applications in society, industry, market, etc. In this paper, a new time series forecasting method (FCD-MLGRU) is proposed for solving short-term forecasting problem. First we decompose the original time series using Filtering Cycle Decomposition (FCD) proposed in this paper, secondly we train the Gated Recurrent Unit (GRU) Neural Network to forecasting the subseries respectively. In the process of training and forecasting, the multi-time-lag sampling and ensemble forecasting method is adopted, which reduces the dependence on the selection of time lag and enhance the generalization and stability of the model. The comparative experiments on the real data sets and theoretical analysis show that our proposed method performs better than other related methods.

Keywords: Time series forecasting · Gated Recurrent Unit Neural Network · Time series decomposition

1 Introduction

Large-scale time series data have widely emerged in the fields of economics, industry, education, society, etc. The forecasting of time series provides important guidance for decision-makers to take corresponding strategy. The forecasting problem of time series can be summed up as following: Build a forecasting model, which can capture the regularity of the history time series, so that it can predict value of the future which approximate the ground-truth. It can be formally written as:

$$x_{t+n}, x_{t+2}, ..., x_{t+1} = f\left(x_t, x_{t-1}, ..., x_{t-w}\right) \tag{1}$$

where f is forecasting model, x_t is time series data point at time step t, n is forecasting horizon, which means the number forecasting ahead, w is window

D. Liu et al. (Eds.): ICONIP 2017, Part V, LNCS 10638, pp. 523–532, 2017.
https://doi.org/10.1007/978-3-319-70139-4_53

size which means the number of historical data used to forecast the future data, we also call it as time lag.

To solve forecasting problem, some classical forecasting methods including Exponential Smoothing, Trend Extrapolation, Moving Average Model (MA), Autoregressive Integrated Moving Average Model (ARIMA) [1], etc. was proposed. However, these methods are linear models, not suitable for the non-linear large-scale time series forecasting.

In recent years, the Artificial Neural Network (ANN), especially deep learning method, has been widely used in the time series forecasting problem. Due to the ability of ANN which can approximate the nonlinear function with arbitrary precision [2], it has strong power and robustness to fit the nonlinear data. Shen *et al.* leveraged the CDBN to forecasting the exchange rate [3]. Shi *et al.* combined the Convolutional Neural Network and LSTM network to forecasting the precipitation [4], Marino *et al.* studied energy forecasting by comparing the standard LSTM and sequence to sequence LSTM [5].

However, there are also two main problems with the above time series forecasting methods. First of all, in these forecasting methods, the time lag is generally fixed, it has a great influence on the final forecasting result but is difficult to determined [6,7]. More importantly, with time series data generating continually, time lag, as a important parameter, may drift with the time. Secondly, if a single forecasting model is setup on time series straightforward, it will be more susceptible to noise information, especially in large-scale time series.

In order to cope with the above-mentioned problems encountered in time series forecasting, we propose a new forecasting method FCD-MLGRU. The contribution of this paper is mainly shown as following. Firstly, we propose Filtering Cycle Decomposition (FCD) method to decompose the time series into trend, cycle and residual subseries. Secondly, because we train the Gated Recurrent Unit (GRU) Neural Network to forecasting the subseries respectively, in the process of training and forecasting, the multi-time-lag sampling and ensemble forecasting method can be adopted. It can reduce the dependence on the selection of time lag and enhance the generalization and stability of the model. The comparative experiments on the real data sets show that our proposed method performs better.

2 Proposed Method

The forecasting method of this paper mainly involves into three parts: Time series decomposition, GRU Neural Network training and forecasting, Multi-lag sampling and ensemble forecasting in process of GRU. The whole process of proposed method is shown in Fig. 1.

First, given a time series, we decompose the original time series into trend, cycle and residual subseries. Based on the traditional time series seasonal decomposition model X-11 [8], we propose a simplified version of this decomposition model, we refer it as Filter Cycle Decomposition (FCD).

Next we train forecasting model on trend and residual subseries after FCD decomposition by using GRU Neural Network. Cycle subseries is invariant in period, so it need't forecasting.

In the process of training and forecasting, different from the previous forecasting methods with fixed time lag, in this paper, we use the variable-length time lag sampling and multi-lag forecasting method. Because the GRU is a variant of the Recurrent Neural Network (RNN), it can handle variable-length sequences.

In the last step, the subseries forecasting results sum up to obtain the final forecasting result.

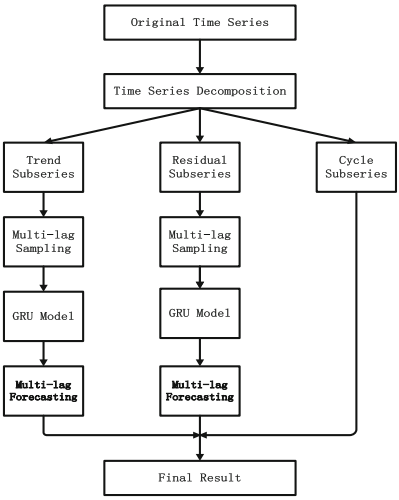

Fig. 1. Flowchart of proposed method

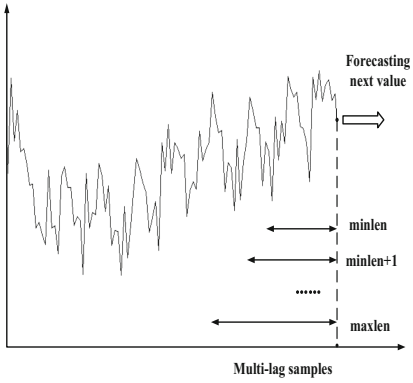

Fig. 2. Multi-lag sampling schematic

2.1 Time Series Decomposition

In the theory of time series analysis, time series can be considered as a combination of several subcomponents. A time series X can be decomposed as the following formula:

$$X\ (t) = T\ (t) + C\ (t) + R\ (t) \tag{2}$$

where $T(t)$ is trend time series, $C(t)$ is cyclic time series, $R(t)$ is residual time series. It is called addition model.

Another form of decomposition is the multiply model. In this paper, we use the additive model of time series decomposition, which means that we believe the subcomponents are completely independent of each other. This assumption will be an important basis for our later decomposition necessity analysis.

The Filtering Cyclic Decomposition (FCD) method is as following:

1. Select a fixed cyclic period m. Use $2 * m$ (m is even) or m (m is odd) window size to do Moving Average (MA) to estimate the trend subseries T.

2. Remove the trend information $y - T$.
3. Calculate the average of each moment in the cyclic period as cyclic m information C.
4. Obtain residual information $R = y - T - C$.

This is simple version of seasonal decomposition. Different from the traditional seasonal decomposition model like X-11 [8], the cycle period here is no longer the periodic frequency of the natural time like 24-h (Day), 7-day (Week) or 30-day (Month), but a fixed small given value (like 4 or 8, has not especially physics meaning). FCD is equivalent to a smooth filter which can extract periodic and residual characteristics. The following experiments show that, in the case of large data volumes at fine-grained time interval like hour or minute, if the cycle frequency of natural time is used as the period of decomposition, sometimes the magnitude of the residual subseries is much greater than trend time series. Residual time series is difficult to find the regularity, so that trend information is overwhelmed by the residual information. The decomposition is not conducive to improve the accuracy of forecasting, even will cause the decline of accuracy. We will show these detail results in the experiments of Sect. 4.

2.2 Gated Recurrent Unit (GRU)

We choose Gated Recurrent Unit (GRU) Neural Network as base regressor. It is a variation of Recurrent Neural Network (RNN). GRU was proposed in [9] to make each recurrent unit to adaptively capture dependencies of different time scales [10]. Compared with the vanilla RNN, GRU can hold a long-distance dependency because it reduces the problem about gradient vanish by introducing the gate. Unlike the Long-Short Term Memory Neural Network (LSTM), another variation of RNN which also can hold long-distance dependency, the GRU's network neural units architecture is much simpler, but its effectiveness is not reduced, sometimes even slightly better than LSTM [10].

As a variation of RNN, GRU Neural Network has a characteristic that more recent time step input will has greater influence on neural network [11], which is in consist with the characteristics of the time series forecasting, i.e. the information which is closer to the forecasting point is more important.

There are two gate structures in GRU: Update Gate $\boldsymbol{z_t}$ and Reset Gate $\boldsymbol{r_t}$. Update Gate decides how much the neural unit updates, Reset Gate decide how much previously state the neural unit forgets. It can be calculate as following formula:

$$\boldsymbol{z_t} = tanh\ (\boldsymbol{W_z} \cdot [\boldsymbol{h_{t-1}}, \boldsymbol{x_t}]) \tag{3}$$

$$\boldsymbol{r_t} = tanh\ (\boldsymbol{W_r} \cdot [\boldsymbol{h_{t-1}}, \boldsymbol{x_t}]) \tag{4}$$

where $\boldsymbol{h_{t-1}}$ is last output and $\boldsymbol{x_t}$ is input, $\boldsymbol{W}$ represent the weight, $\cdot$ is element-wise multiply.

After calculating the gate, we must choose what information will be added in the neural units memory as $\tilde{\boldsymbol{h}}_{\boldsymbol{t}}$ and expose state as output $\boldsymbol{h_t}$, it can be calculated as following formula:

$$\tilde{\boldsymbol{h}}_{\boldsymbol{t}} = tanh\ (\boldsymbol{W} \cdot [\boldsymbol{r}_{\boldsymbol{t}} * \boldsymbol{h}_{\boldsymbol{t-1}}, \boldsymbol{x}_{\boldsymbol{t}}]) \tag{5}$$

$$\boldsymbol{h}_{\boldsymbol{t}} = (\boldsymbol{1} - \boldsymbol{z}_{\boldsymbol{t}}) * \boldsymbol{h}_{\boldsymbol{t-1}} + \boldsymbol{z}_{\boldsymbol{t}} * \tilde{\boldsymbol{h}}_{\boldsymbol{t}} \tag{6}$$

More details about GRU Neural Networks can be referred to the [9].

2.3 Variable-Length Sampling and Multi-lag Ensemble Forecasting

In this paper we use the variable-length time lag sampling method in GRU training and forecasting to take full advantage of GRU which can handle variable length sequence. At the time of training, we forecast value of next time point x_{t+1} as ground-truth, we specify minimum length called L_{min} and a maximum length called L_{max}, then cut sequences in all possible length within $L_{min} \sim L_{max}$ prior to this point as regressor input, like $[x_t, x_{t-1}, ..., x_{t-L_{min}}]$, $[x_t, x_{t-1}, ..., x_{t-L_{min}+1}]$, ..., $[x_t, x_{t-1}, ..., x_{t-L_{max}}]$, so we get a group of samples-ground-truth pairs, then we can continuously slide to get next group. Note that the sequences whose length are less than L_{max} are padded with zero at the left end, so it can be operated by the GRU. The schematic diagram of multi-lag sampling is shown in Fig. 2.

At the time of forecasting, we use GRU to get different forecasting results in different time-lag, and finally take average of them as final result. This method can effectively suppress the instability of single-time-lag forecasting results and enhance the generalization ability of the model. It also reduce the dependence on selection of the time lag because we only need to specify a approximate range rather than a fixed accurate time lag. It can even maintain forecasting ability under the case when time lag drifts in data-stream.

3 Analysis

3.1 Why Decomposition

We make a semi-quantity explanation about the advantage of time series decomposition from the viewpoint of machine learning model error theory.

The square expectation error of any machine learning model consists of three parts, bias, variance and random noise [12]. As shown in the formula:

$$E\ (f\ (\boldsymbol{x}) - y')^2 = \sigma^2 + Var\ (\boldsymbol{x}) + Bias\ (\boldsymbol{x})^2 \tag{7}$$

where $\boldsymbol{x}$ is input data, y' is ground-truth value, σ is variance of the random noise, Var and $Bias$ represent variance and bias of the model.

After time series decomposition, the cyclic information can be completely retained in the final result. Assuming that under normal circumstances, the cyclic attributes will not change (if there is change, they all included in the residual information), so the cyclic subseries forecasting result bias and variance can be considered to be zero. In addition, without interference of cyclic information and residual information, the trend curve is very smooth, thus it can be easily fitted,

so its bias and variance will be smaller. The residual subseries fluctuates greatly and its regularity is not obvious. It is the main source of the forecasting error. Therefore, it is necessary to reduce the magnitude of the residual time series in order to drop its effect to the final forecasting result.

According to the previous discussion, the subcomponents after the time series decomposition are independent to each other. Thus it means its covariance is 0. According to the attribution of the bias and variance, (7) can be overwritten as following:

$$\begin{aligned} & E\left(f\left(\boldsymbol{x}\right)-\hat{y}\right)^2=\sigma^2+Var\left(\boldsymbol{x}\right)+Bias\left(\boldsymbol{x}\right)^2 \\ & =Bias\left(T+C+R\right)^2+Var\left(T+C+R\right)+\sigma^2 \\ & =\left(Bias\left(T\right)+Bias\left(C\right)+Bias\left(R\right)\right)^2 \\ & +Var\left(T\right)+Var\left(C\right)+Var\left(R\right)+\sigma^2 \end{aligned} \tag{8}$$

where $\boldsymbol{x}$ is original series data, T,C,R represent trend, cyclic, residual data respectively after decomposition. According to the discussion above, we can see that $Bias\left(C\right)=0$ and $Var\left(C\right)=0$, other items should be reduced after decomposition. This is reason why we use FCD. That is to say, we need trend time series after smoothing filter is easier to fit, and the magnitude of residual time series is as small as possible to reduce the effect of residual forecasting on the final result.

Of course, if the sub-components are not independent, we also can analyze the final forecast results through the covariance changing. We will discuss this problem in the future.

3.2 Advantages of Multi-lag Sampling and Ensemble Forecasting

Many researchers have devoted to discuss how to choose appropriate time lag. Frank *et al.* uses the False Nearest Neighbour method and Heuristics for window size estimation to select the appropriate time lag [13]. Rahman *et al.* train several ANN as the base regressor under different time lag for ensemble learning [14]. Most of these methods use traditional neural networks or machine learning algorithms, and their input dimensions are generally fixed, so it is difficult to deal with situations of different time lag. Generally these methods are training completely independent models under different time lags or require a lot of cross validation experiments to determine the final time lags.

However, GRU neural network can cope with different time lag. Assume we specify max time lag as L_{max} and min time lag as L_{min}. L is the total length of a time series, according to the sampling method mentioned in Sect. 2.3, we can get max number of all possible training samples groups G is:

$$G=L-L_{max} \tag{9}$$

every group has different time lag samples, so we can get max number of all possible training samples N:

$$N=\left(L_{max}-L_{min}+1\right)*G \tag{10}$$

So it can be seen that we get more training samples through this method. This is equivalent to finding the dependencies of all possible sequences within the minimum length to maximum length, which is used to train the GRU. Note that we use all these training samples with different time lag to train only one GRU neural network, not independent models under different time lags.

In forecasting process, we calculate the forecasting results under different time lag and finally take the average as the final result. Although there is only one GRU model on each subseries, but multi-time-lag forecasting are equivalent to ensemble learning method forecasting. According to the theory of ensemble learning, the variance of the forecasting model is reduced.

4 Experiment

4.1 Data

In order to verify the effectiveness of our proposed method, we conducted comparative experiments on several real time series data sets. The experimental data includes NSW 2013 and TAS 2016 annual electricity demand, the length is 17521, both of them data are collected from the Australian Energy Market Operator (AEMO) (https://www.aemo.com.au/). There is also a shared bicycle registration data record, the sequence length is 17380, collected from the UCI time series data set (https://archive.ics.uci.edu/ml/datasets.html).

We use the three most common metrics to evaluate the merits of these models: mean absolute error (MAE), mean root square error (MRSE) and symmetric mean absolute percentage error (SMAPE).

4.2 Results

GRU as Base Regressor. The single forecasting model comparative experiment results are shown in Table 1. It can be seen from the experimental results that the forecasting accuracy of LSTM and GRU is better than the other methods. Vanilla RNN can not memorize long sequences due to gradient vanish, thus the performance on the three data sets are not good enough. ANN and SVR are

Table 1. Base regressor used in forecasting straightforward

Method	NSW2013			Shared-Bike			TAS2016		
	MAE	RMSE	SMAPE	MAE	RMSE	SMAPE	MAE	RMSE	SMAPE
ANN	68.15	88.61	0.90	31.46	45.96	31.01	16.27	22.37	1.54
GRU	61.06	80.09	0.81	26.62	41.75	32.76	14.77	20.77	1.41
LSTM	65.12	83.82	0.88	29.22	47.69	30.17	14.75	20.74	1.40
RNN	70.54	91.71	0.93	33.91	48.05	34.60	24.23	30.37	2.31
SVR	66.85	87.12	0.89	35.97	55.58	36.30	14.74	21.03	1.41

slightly worse than RNN family, but they must require the fixed input dimension. Although the effect of LSTM and GRU is roughly close, the structure of GRU is more simple, thus it iterates and converges more faster than LSTM, thus we choose GRU as base regressor.

The Effectiveness of Time Series Decomposition. We compare the traditional seasonal decomposition (X-11) and the Filter Cycle Decomposition (FCD) method on NSW2013 and Shared-Bike, which have fine-grained record frequency (like hours or minutes) and larger data volume. The results of the two decomposition in Shared-Bike data set are shown as Figs. 3 and 4, it can be seen that the residual time series magnitude of FCD is lower than X-11, and cycle information is much more regular.

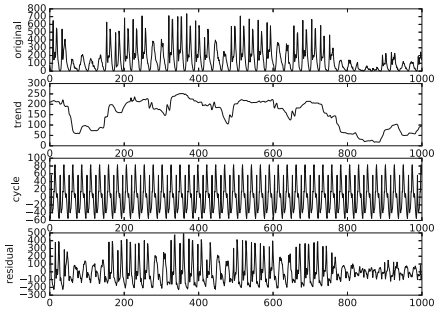

Fig. 3. Shared-Bike X-11 weekly seasonal decomposition (regional, peroid = 7 days)

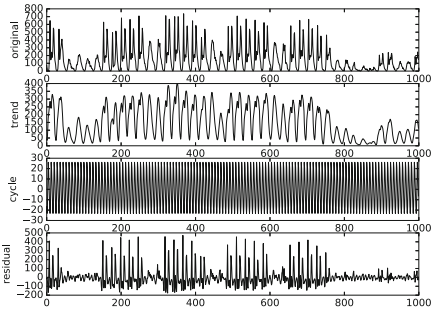

Fig. 4. Shared-Bike filter cycle decomposition (regional, peroid = 8)

In the experiment, we combined the two kinds of decomposition methods into the single forecasting model, the experimental results are shown in Table 2, obviously in most time, the second method is better. It is consistent with our previous theoretical analysis in Sect. 2.2.

Table 2. Forecasting using traditional time series decomposition and filter cycle decomposition

Method	NSW2013			Bike-Shared			Method	NSW2013			Bike-Shared		
	MAE	RMSE	SMAPE	MAE	RMSE	SMAPE		MAE	RMSE	SMAPE	MAE	RMSE	SMAPE
X-11ANN	70.51	91.78	0.92	33.14	48.69	38.12	FCD-ANN	80.93	99.44	1.05	15.69	21.53	24.06
X-11GRU	62.22	81.38	0.82	28.93	41.40	34.23	FCD-GRU	38.47	47.11	0.51	15.83	21.71	22.43
X-11LSTM	66.03	85.08	0.87	34.26	49.64	34.49	FCD-LSTM	41.23	51.62	0.53	13.68	19.61	17.98
X-11RNN	70.32	89.27	0.93	33.40	45.54	34.99	FCD-RNN	33.65	43.02	0.44	20.82	29.35	31.25
X-11SVR	59.52	76.66	0.78	35.56	53.72	38.97	FCD-SVR	45.83	59.02	0.63	17.04	28.58	21.38

Comparative Experiments. The above comparative experiments are carried out in the case of the same fixed time lag at 20. As discussed above, time lag itself is a difficult choice. For this reason, we use variable-length sampling to increase the number of training samples as discussed in Sect. 3.2, while using multi-lag ensemble forecasting method at forecasting phase. In the experiments, we conducted comprehensive comparative experiments on NSW2013, TAS2016 and Shared-Bike three data sets. In the experiment, we used the vanilla RNN, GRU, LSTM, ANN, SVR algorithm and these after decomposition version (FCD-RNN, FCD-GRU, FCD-LSTM, FCD-ANN, FCD-SVR) and the method which we propose in this paper (FCD-MLGRU). The experimental results are shown in Table 3.

Table 3. Comprehensive comparative experiments results

Method	NSW2013			Bike-Shared			TAS2016		
	MAE	RMSE	SMAPE	MAE	RMSE	SMAPE	MAE	RMSE	SMAPE
ANN	68.15	88.61	0.90	31.46	45.96	31.01	16.27	22.37	1.54
GRU	61.06	80.09	0.81	26.62	41.75	32.76	14.77	20.77	1.41
LSTM	65.12	83.82	0.88	29.22	47.69	30.17	14.75	20.74	1.40
RNN	70.54	91.71	0.93	33.91	48.05	34.60	24.23	30.37	2.31
SVR	66.85	87.12	0.89	35.97	55.58	36.30	14.74	21.03	1.41
FCD-ANN	80.93	99.44	1.05	15.69	21.53	24.06	14.59	20.75	1.39
FCD-GRU	38.47	47.11	0.51	15.83	21.71	22.43	10.38	14.65	0.98
FCD-LSTM	41.23	51.62	0.53	13.91	19.97	17.98	7.48	10.45	0.71
FCD-RNN	33.65	43.02	0.44	20.82	29.35	31.25	9.84	12.35	0.95
FCD-SVR	45.83	59.02	0.63	17.04	28.58	21.38	9.62	13.87	0.92
FCD-MLGRU	**25.36**	**33.38**	**0.34**	**13.82**	**19.81**	**17.33**	**6.87**	**9.81**	**0.65**

Through the results of the final comprehensive experiment, we can see that the forecasting error of proposed method is smaller than the remaining methods. On the one hand, almost all forecasting model after using decomposition performs better than single forecasting model, so it can be seen that the time series decomposition is important and necessary. On the other hand, after using variable-length sampling and multi-lag ensemble forecasting, the stability and generalization ability of the model are further strengthened, so the forecasting error is reduced further. In addition, we do not need to specify a time lag, but rather specify a range of time lag, which greatly reduces the dependence on the fixed time lag, reducing the burden of adjusting parameters. Even in the time series stream generation process, the model can also be adjusted adaptively to cope with time lag drift.

5 Conclusion

In this paper, a new time series forecasting method is proposed, which combines Filtering Cycle Decomposition (FCD), GRU Neural Network, variable length time lag sampling and multi-lag ensemble forecasting. Through the theoretical analysis, the necessity of time series decomposition is studied, and the advantages of variable length sampling and multi-lag ensemble forecasting are illustrated. The experimental results show that the method proposed in this paper performs better than other related methods on the real data sets.

Acknowledgments. This work is supported in part by the National Science Foundation of China under Grant Nos. (61373130, 61375064, 61373001), and Jiangsu NSF grant (BK20141319).

References

1. Box, G.E.P., Jenkins, G.: Time Series Analysis, Forecasting and Control. Holden-Day, Amsterdam (1976)
2. Hornik, K., Stinchcombe, M., White, H.: Multilayer feedforward networks are universal approximators. Neural Netw. **2**(5), 359–366 (1989)
3. Shen, F., Chao, J., Zhao, J.: Forecasting exchange rate using deep belief networks and conjugate gradient method. Neurocomputing **167**(C), 243–253 (2015)
4. Shi, X., Chen, Z., Wang, H., Yeung, D.Y., Wong, W., Woo, W.: Convolutional LSTM network: a machine learning approach for precipitation nowcasting. Computer Science (2015)
5. Marino, D.L., Amarasinghe, K., Manic, M.: Building energy load forecasting using deep neural networks (2016)
6. Chen, H., Yao, X.: Ensemble regression trees for time series predicitions (2008)
7. Zhang, G.P., Berardi, V.L.: Time series forecasting with neural network ensembles: an application for exchange rate prediction. J. Oper. Res. Soc. **52**(6), 652–664 (2001)
8. Shiskin, J., Young, A.H., Musgrave, J.C.: The X-11 Variant of the Census Method II Seasonal Adjustment Program. U.S. Department of Commerce, Bureau of the Census, Suitland (1967)
9. Cho, K., Van Merrienboer, B., Bahdanau, D., Bengio, Y.: On the properties of neural machine translation: encoder-decoder approaches. Computer Science (2014)
10. Chung, J., Gulcehre, C., Cho, K.H., Bengio, Y.: Empirical evaluation of gated recurrent neural networks on sequence modeling. Eprint Arxiv arXiv:1412.3555 (2014)
11. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473 (2014)
12. Bishop, C.M.: Pattern Recognition and Machine Learning (Information Science and Statistics). Springer, New York (2006). p. 049901
13. Frank, R.J., Davey, N., Hunt, S.P.: Time series prediction and neural networks. J. Intell. Robot. Syst. **31**(1), 91–103 (2001)
14. Rahman, M.M., Islam, M.M., Murase, K., Yao, X.: Layered ensemble architecture for time series forecasting. IEEE Trans. Cybern. **46**(1), 270 (2016)