from Australia.[1] Experimental results demonstrate the advantages of our approach compared with 4 baselines: History Average, Seasonal Autoregressive Integrated Moving Average Model, Multi-Layer Perceptron and Long Short-Term Memory.

The remainder of this paper is structured as follows. Section 2 reviews related works. In Sect. 3 we present our methodology. In Sect. 4 we describe the data set used in our study, and present our experimental settings, evaluation measures, and the empirical results with discussion. Finally, Sect. 5 concludes the paper and gives an outlook on future work.

2 Related Works

Many approaches had been reported in the literature on short-term load forecasting. However, most existing works deal with group behavior of power consumption instead of individual residents, since residential forecasts were conventionally considered trivial due to the volatile nature of individual loads. Therefore, the issues on short-term household load forecasting remain open [3–10]. In literature, methods for energy load forecasting include: time-series analysis (e.g. ARIMA [11]), regression analysis method [12], neural networks [13], support vector machine [14], fuzzy theory [15], etc.

Cao et al. [11] adopted ARIMA considering meteorological conditions for intraday load forecasting, so as to enhance the forecasting accuracy and robustness. Pang et al. [16] proposed a neural network which takes only the load values of current and previous time steps as the input to predict the load value at next time step. The result is further compensated by rough set to increase the forecasting accuracy. Chaouch et al. [17] proposed a functional time series for short term forecasting of household-level intra-day electricity load curve. Ghofrani et al. [18] used spectral analysis and Kalman filter on short-term load forecasting for residential customers.

As an improved method of Recurrent Neural Network (RNN), the Long Short-Term Memory (LSTM) [19] has made great progress in the field of sequence learning. Marino et al. [20] validated two variants of the LSTM: standard LSTM and LSTM-based Sequence to Sequence (S2S) architecture for single-meter residential load forecasting issue. Kong et al. [21] also proposed a LSTM based framework in the same task.

Deeper and deeper CNNs (Convolutional Neural Network) were proposed by researchers [22, 23], which reveals the importance of network depth, that is, deeper networks typically lead to superior results. However, with the increase of network depth, the gradient vanishing problem emerges in the process of neural network back propagation. By using shortcut connections, Residual Networks (ResNets) [24] perform residual mapping fitted by stacked nonlinear layers. It solves the gradient vanishing problem along with the increase of network depth. Combining these two approaches, Zhang et al. [25] proposed a deep-learning based approach, called ST-

[1] http://data.gov.au/dataset/electricity-consumption-benchmarks.

ResNet, to forecast the inflow and outflow of crowds in each and every region of a city. Inspired by above existing works, we use one-dimensional convolution to extract features from power consumption data sequence collected through smart meters and optimize the structure of fusion network.

3 Methodology

Smart meters will periodically collect residential consumption data, and time interval is typically half an hour. Therefore, our task is formulated as following: given meters reading data series of observations $\{S_t | t = 0, 1, \ldots, n-1\}$, predict future electricity consumption S_n. Unlike electric load at the system level, the complexity of household load forecasting lies in the significant volatility and uncertainty. As mentioned in [21], the diversity in the aggregated level smooths the daily load profiles, while the electricity consumption of a single resident is more dependent on the underlying human behavior. Daily routines and lifestyle of individual residents, together with various types of appliances, may have a more direct impact on the load profile. In order to improve the prediction performance, we propose to abstract hidden knowledge from previous observation data and establish a learning algorithm to model the correlation between abstracted knowledge and forecasting targets.

3.1 Residual Networks

The general idea of the Residual Networks [24] is that, by using shortcut connections, residual networks perform residual mapping fitted by stacked nonlinear layers, which is easier to be optimized than the original mapping. Identity Mapping ResNet [26] simplifies the residual networks and solves gradient vanishing problem along with the increase of the network depth, and thus greatly improves the training effect of the network. A residual unit [26] with an identity mapping is defined as:

$$X^{(l+1)} = X^{(l)} + F(X^{(l)}) \tag{1}$$

where $X^{(l)}$ and $X^{(l+1)}$ are the input and output serial of the lth residual unit respectively, and F is a residual function [24].

3.2 Residual Conventional Fusion Network (RCFNet)

Figure 1 shows the structure of the model used in this paper.

As described earlier, the time series data of power consumption is obtained by half-an-hour interval. Here we divide the temporal correlation of data into three types: (a) Adjacent data, we use the most recent 24 h (half an hour interval, 48 data points) for proximity structure; (b) Period data, such as electricity consumption data at the same time every day, and the last 28 days (28 data points) are selected; (c) Trend data, we choose the last ten weeks of data (10 data points) for tendency structure. In this way, time series data of three characteristics is constructed as input of the model. The same residual convolution network structure is employed to extract above three types of features, which is shown by Fig. 1 (details of ResConvNet in Fig. 1 will be discussed in Sect. 3.3).

As for external factors, we manually extract some features. Considering the difficulties in external information acquisition, here we simply differentiate weekdays and weekends. Specifically, consumption data during weekdays or weekends is encoded into a one-hot vector, as the input to a two-layer fully connected network. The first three modules (i.e. tendency, periodicity, proximity) output a feature vector with a length of fixed size (e.g. 10), and the external module also outputs a vector with the same length, and then the four outputs are spliced into one vector. Then, the vector is used as the input of the FusionNet (detail shown in Fig. 2c). Finally, output from FusionNet is fed to a ReLU (Rectified Linear Unit) activation function and the results are mapped to [0, 1].

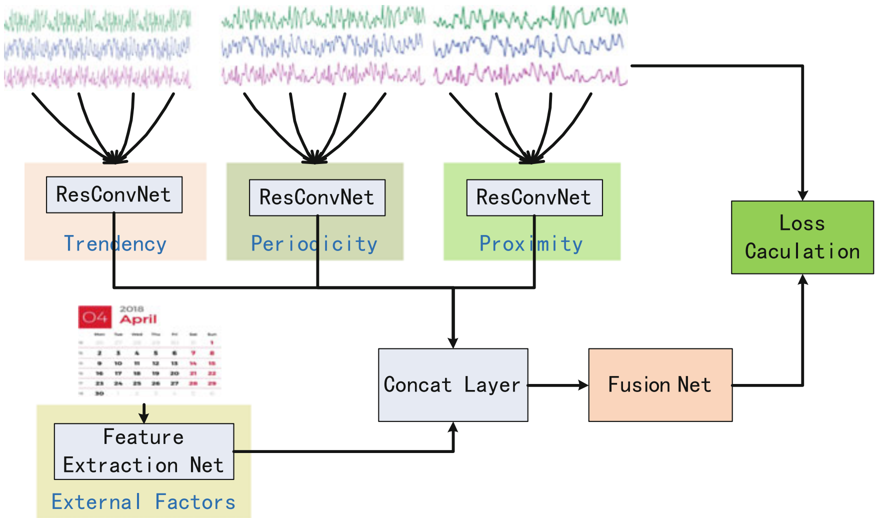

Fig. 1 RCFNet architecture

3.3 Time Serial Feature Extraction Network

The first three components (tendency, periodicity, proximity) share the same network structure, as shown in Fig. 2a. Details of each component are explained in the following.

(1) **Convolution**. A good model should be able to capture time series features. Convolution Neural Network has a strong ability to grasp the spatial structure information characteristics [27], in which two-dimensional convolution kernel is used to extract features of adjacent areas of image. Note that power consumption data is one-dimensional time series data. Thus we consider using one-dimensional convolution kernel to perform one-dimensional convolution, so that the adjacent features in time series can be extracted. By stacking multiple layers of convolution network layer, we can obtain larger receptive field. Considering the loss of resolution caused by subsampling while preserving distant dependencies [28], we do not use subsampling, but only convolutions.

The proximity module in the structure showed in Fig. 1 samples the power consumption data over a recent period to model the proximity characteristics of data. Here $X_t^a = \left(X_t^a\right)^0 = [S_{t-l_a}, S_{t-(l_a-1)}, \ldots, S_{t-1}]$ represents the proximity dependency sequence as an input to the entire network at time t, and proceeding to the next layer through convolution operations as:

$$\left(X_t^a\right)^1 = f((W_a)^1 * \left(X_t^a\right)^0 + (b_a)^1) \tag{2}$$

where $*$ denotes the convolution, f is an activation function, $(W_a)^1$, $(b_a)^1$ are the learnable parameters in the first layer.

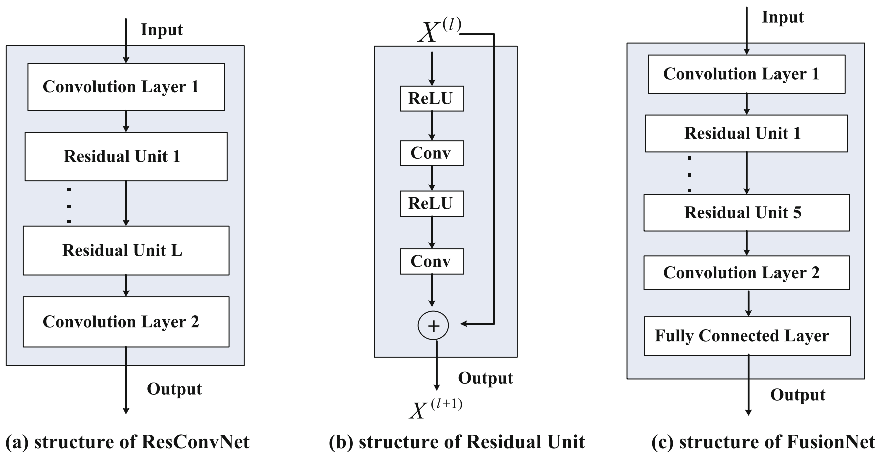

Fig. 2 Time serial feature extraction network

(2) **Residual Unit**. We use the residual learning method [26] to extract features of the sequence from deep network. In our residual convolution network as in Fig. 2a, b, we stack L residual units on the convolution output

$$\left(X_t^a\right)^{l+1} = \left(X_t^a\right)^l + F(\left(X_t^a\right)^l;\ (\theta_a)^l),\ l = 1, \ldots, L \tag{3}$$

Here, F represents a residual function (e.g. a combination of ReLU and convolution, see Fig. 2b), and $(\theta_a)^l$ contains all the learnable parameters of layer L. We use batch normalization [29] before ReLU layer. At the top of all residual units, we add a convolution layer to build a network structure composed of two convolution layers and one residual unit, which is used to extract the adjacent features.

Similarly, extraction of periodic and trend characteristics are achieved through the same structure (see Fig. 1). Here $X_t^p = \left(X_t^p\right)^0 = [S_{t-l_p \cdot p}, S_{t-(l_p-1)\cdot p}, \ldots, S_{t-p}]$ and $X_t^q = \left(X_t^q\right)^0 = [S_{t-l_q \cdot q}, S_{t-(l_q-1)\cdot q}, \ldots, S_{t-q}]$ respectively represent that extraction of periodic and trend sampling data, l_p and l_q respectively represent the length of the corresponding sequence. p is the sampling span representing the periodicity of the sequence, which is equal to one-day that describes daily periodicity. q is equal to one-week representing the trend of the sequence. The number of data sampling points is 48 per day, and the corresponding settings for periodicity interval p and trend interval q are respectively $p = 48,\ q = 7 \times 48$.

(3) **External Component**. Power consumption largely depends on residential behavior, and there are a lot of influential factors, such as weather conditions, holidays, etc. Power consumption behavior differs among weekdays and weekends. Limited by data availability, here we simply consider the influence of weekdays and weekends on residential power consumption. We encode it as a one-hot vector connected to the two layers of the full connection layer, where the external output is defined as E_t, and the output is mapped to a vector length of fixed size (e.g. 10) to ensure consistent outputs of previous three components.
(4) **Fusion**. We concatenate four modules' output (i.e. tendency, periodicity, proximity and external factor) above to one vector. We feed the vector into the FusionNet (see Fig. 2c),which consist of one convolution layer, five stacked residual units, one convolution layer and one fully connected layer. Finally, output from FusionNet is fed to a ReLU (Rectified Linear Unit) activation function and the results are mapped to [0, 1].

The network trains predictions $\widehat{S}_t$ by inputting three data sequences and extrinsic features, and then, minimizes the mean squared error between the predicted power consumption and the actual power consumption:

$$MSE(\theta) = \left\| S_t - \widehat{S}_t \right\|_2^2 \tag{4}$$

Here θ denotes learnable parameters, and S_t is observed value at time t.

3.4 RCFNet Training

The program design is demonstrated with pseudo code in Program 1. First, training instances are constructed from original sequence data (lines ①–②). Then, RCFNet is trained via back-propagation and Adam [30] optimizer (lines ③–⑧) (Here we use Adam optimizer because [31] indicates it outperforms other candidates, including the stochastic gradient descent (SGD), Adagrad [32], Adadelta [33] and RMSProp [34]). Finally, we validate the model on the test set (⑨–⑩).

Program 1:RCFNet Training Program
Input:

length of adjacent, periodicity, trend sequences: l_a, l_p, l_q;
period interval: p; trend interval: q;
historical observations: $S = \{\ S_0, \ldots, S_{n-1}\}$;
external features: $E = \{\ E_0, \ldots, E_{n-1}\}$;

Data preparing:

① for all available time interval $t(1 \leq t \leq n-1)$, add series data $X_t^a = \left[S_{t-l_a}, S_{t-(l_a-1)}, \ldots, S_{t-1}\right]$, $X_t^p = \left[S_{t-l_p \cdot p}, S_{t-(l_p-1)\cdot p}, \ldots, S_{t-p}\right]$, $X_t^q = \left[S_{t-l_q \cdot q}, S_{t-(l_q-1)\cdot q}, \ldots, S_{t-q}\right]$, E_t, $S_t$to set $D = (\{X_t^a, X_t^p, X_t^q, E_t\}, S_t)$ (S_t and E_t are the actual power consumption and external factors at time t, respectively)
② divide D into training set D_{tr} and test set D_{ts}

Model Training:

③ build network on Keras [35] with Tensorflow [36] backend
④ initialize all learnable parameters θ in RCFNet
⑤ repeat:
⑥ randomly select a batch training data set from that set D_{tr}
⑦ evaluate performance by mean squared error to find trainable parameter θ
⑧ until the convergence condition is satisfied

Output:

Learned RCFNet model

Model Evaluation:

⑨ fetch all the samples from test set D_{ts} as the input to the learned RCFNet model
⑩ evaluate performance by using Root Mean Square Error (RMSE)

4 Experiments

In this section, we perform experiments on power consumption dataset from Australia to validate the efficiency of the proposed RCFNet. The dataset contains power consumption from 25 households in Victoria, published by Australian Government, Department of Industry, Innovation and Science, with a two-year coverage from April 1, 2012 to March 31, 2014. The sampling interval is half an hour. To eliminate data missing issue, we selected 8 households with full data coverage in this work.

4.1 Baselines

We compare our RCFNet with the following 4 baselines:

(1) **HA** (History Average). It is the most straightforward way for prediction. The basic assumption is that power consumption during each period over days exhibits strong regularity. The amount of electricity consumed is predicted by average the historical consumption during the same period on previous days.
(2) **SARIMA** (Seasonal ARIMA). It takes seasonal period difference into consideration. The seasonal parameter is set to days (48 sampling intervals) as the periodic characteristics of residential electricity usage are repeated on a daily basis.
(3) **MLP** (Multi-Layer Perceptron). A three-layer artificial neural network is used to predict the amount of electricity used. The model uses one day's data before the point (48 time periods, half an hour apart) as input.
(4) **LSTM** (Long Short-Term Memory). The model can connect previous time information to the task of the current time to complete the memory of the power consumption habits. A three-layer LSTM network is used here to predict the amount of power used the day before the point in time (48 time periods, half an hour apart) as input.

4.2 Preprocessing

All models are built on a desktop PC with a 3.4 GHz Intel i7-6700 processor and 16 GB of memory, using Python 3.6.4 and Keras 2.1.5 with TensorFlow 1.3.0 backend as model training tools.

In our RCFNet model, we first use the min-max regularization method to normalized training data to values between [0, 1]. In the output part of RCFNet, we use the ReLU activation function, which has better training convergence speed and better effect. During the validation phase, we convert the predicted values into normal values by reversing the operation. The external factors here, such as weekdays or weekends, are encoded as a one hot vector.

Convolutions use filters of size 3×1. Each residual unit uses two convolution layers. Here we use fixed layer size of residual units. The training set is divided into two parts, where the first 90% is used for training and the last 10% for validation. To prevent excessive training iterations, we implement early stopping technique to find optimal number of training iterations. After obtaining the best model, we continue to do fixed epoch training on the best model to get better results. Here we continue to train 50 times as the final result of model training. In the LSTM and MLP models, the sequence data with the length of 48 is used as input, while the final fusion model adopts the following parameters: the length of adjacent sequence is 48, the length of periodic sequence is 28, and the length of trend sequence is 10.

In order to maintain the consistency of the results, we use the last 60 days for model validation and the rest of the data as training data sets.

We validate the prediction performance using Root Mean Square Error (RMSE), which is defined as:

$$RMSE = \sqrt{\frac{1}{n}\sum_{t=1}^{n}(observed_t - predicted_t)^2} \tag{5}$$

where n is the total number of predictions.

4.3 Results

We validate our model against the other four models in Table 1, where the performance of each model is listed by each household ID. Moreover, in order to testify the influence of different parameter factors on RCFNet, we design six different model variants on RCFNet, RCF (network considering adjacent, periodicity, tendency and external factors), RCF-NoE (RCF without considering external factors), RCF-NoP (RCF without considering periodicity), RCF-NoQ (RCF without considering tendency), RCF-NoPQ (RCF without considering periodicity and tendency) and RCF-NoEPQ (RCF without considering periodicity, tendency and external factors). The best result in each column is shown by the bold underlined numbers.

From Table 1, we can see that RCFNet outperforms the other four baselines, which indicates that RCFNet captures electricity consumption characteristics well. In addition, we can also find that the impacts of proximity, periodicity, tendency, and external factors differ among households. This reflects that different residents' electricity consumption characteristics are quite different. Some residents have stronger periodicity, while others show more obvious trend characteristics. It can be seen that if some feature branches are not taken into account, some results of RCFNet may be worse than certain baseline models, which shows that these features have great influence on the accuracy of RCFNet. Here, we only simply consider the settings $l_a = 48$, $l_p = 28$, $l_q = 10$ for the length of adjacent l_a, periodic l_p and trend sequence l_q, and the influence of weekdays or weekends for external factor. If we consider the influence of these factors more comprehensively, our model may achieve better results. On the whole, the network which comprehensively considers the proximity, periodicity, tendency and external factors has better prediction performance.

Table 1 Comparison among different methods on Australia Set

Model	ID-1098	ID-17625	ID-3117	ID-3494	ID-3762	ID-4192	ID-8927	ID-9918
HA	259.0383	143.8904	387.866	103.2604	2997.6932	326.3549	181.6399	100.9354
SARIMA	175.1828	107.4669	280.309	91.8854	2106.2326	221.5403	138.7098	78.6782
MLP	196.7551	132.1633	306.0178	119.6558	2892.7752	300.1101	170.6765	101.0996
LSTM	185.5381	112.8852	284.9665	91.2207	2169.6357	229.701	142.3328	81.8849
RCFNet (ours)								
RCF	**168.9863**	**107.2604**	**280.037**	**86.3546**	**2087.2318**	**215.0213**	**137.0944**	**77.0784**
RCF-NoE	175.0846	109.1304	280.7952	88.4304	2099.0213	219.0942	139.6173	77.2698
RCF-NoP	171.8832	108.7702	281.3781	86.974	2138.8793	219.5323	138.2551	78.2808
RCF-NoQ	172.3574	108.7478	280.7664	87.3681	2156.0119	216.7504	138.9258	77.4132
RCF-NoPQ	175.8988	109.157	282.172	87.6574	2284.9714	220.4727	140.4179	78.978
RCF-NoEPQ	178.489	109.4951	282.8467	89.7897	2309.8326	224.6844	140.9597	79.9351

5 Conclusion and Future Work

In this paper, we propose a deep network architecture, called RCFNet, based on the historical data of residential electricity consumption to achieve accurate short-term electricity consumption forecast. The network structure takes into account the proximity, periodicity and tendency property of electricity consumption, as well as the influence of external factors. Through an extensive case study on residential power consumption dataset in Australia, the forecasting performance of proposed RCFNet is validated compared with other four typical forecasting models (i.e. History Average, Seasonal ARIMA, Multi-Layer Perceptron and Long Short-Term Memory). Besides, RCFNet has good scalability and can be widely used in other types of time series prediction. In future, we will further explore the characteristics of power consumption, and consider to optimize the RCFNet structure using GAN (Generative Adversarial Networks).

Acknowledgements This research work was partly supported by National Key Research and Development Program of China (Grant No. 2016YFC0800100), Major Research Program of the National Natural Science Foundation of China (Grant No. 91546103), and Anhui Provincial Natural Science Foundation (Grant No. 1708085QG162).

References

1. W. Kong, Z.Y. Dong, Y. Jia et al., Short-term residential load forecasting based on LSTM recurrent neural network. IEEE Trans. Smart Grid **PP**(99), 1–1 (2017)
2. O.I. Asensio, M.A. Delmas, Nonprice incentives and energy conservation. Proc. Natl. Acad. Sci. **112**(6), 510–515 (2015)
3. A. Marinescu, C. Harris, I. Dusparic et al., Residential electrical demand forecasting in very small scale: An evaluation of forecasting methods, in *International Workshop on Software Engineering Challenges for the Smart Grid* (IEEE, 2013), pp. 25–32
4. E. Mocanu, P.H. Nguyen, M. Gibescu et al., Deep learning for estimating building energy consumption. Sustain. Energy Grids Netw. **6**, 91–99 (2016)
5. A. Veit, C. Goebel et al., Household electricity demand forecasting: benchmarking state-of-the-art methods (2014)
6. B. Stephen, X. Tang, P.R. Harvey et al., Incorporating practice theory in sub-profile models for short term aggregated residential load forecasting. IEEE Trans. Smart Grid **8**(4), 1591–1598 (2017)
7. S. Humeau, T.K. Wijaya, M. Vasirani et al., Electricity load forecasting for residential customers: exploiting aggregation and correlation between households, in *Sustainable Internet and ICT for Sustainability* (IEEE, 2013), pp. 1–6
8. M.D. Wagy, J.C. Bongard, J.P. Bagrow et al., Crowdsourcing predictors of residential electric energy usage. IEEE Syst. J. **PP**(99), 1–10 (2017)
9. Y.H. Hsiao, Household electricity demand forecast based on context information and user daily schedule analysis from meter data. IEEE Trans. Industr. Inf. **11**(1), 33–43 (2017)
10. Y. Wang, Q. Xia, C. Kang, Secondary forecasting based on deviation analysis for short-term load forecasting. IEEE Trans. Power Syst. **26**(2), 500–507 (2011)
11. X. Cao, S. Dong, Z. Wu et al., A data-driven hybrid optimization model for short-term residential load forecasting, *in IEEE International Conference on Computer and Information Technology;*

Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing (IEEE, 2015), pp. 283–287
12. Y. Li, D. Niu, Application of principal component regression analysis in power load forecasting for medium and long term, in *International Conference on Advanced Computer Theory and Engineering* (IEEE, 2010), pp. V3-201–V3-203
13. Z.H. Osman, M.L. Awad, T.K. Mahmoud, Neural network based approach for short-term load forecasting. Int. J. Sci. Environ. Technol. **1**(5), 1–8 (2012)
14. N. Ye, Y. Liu, Y. Wang, Short-term power load forecasting based on SVM, in *World Automation Congress* (IEEE, 2012), pp. 47–51
15. C.L. Zhang, Power system short-term load forecasting based on fuzzy clustering analysis and rough sets. J. North China Electric Power Univ. (2008)
16. Q. Pang, M. Zhang, Very short-term load forecasting based on neural network and rough set, in *International Conference on Intelligent Computation Technology and Automation* (IEEE, 2010), pp. 1132–1135
17. M. Chaouch, Clustering-based improvement of nonparametric functional time series forecasting: application to intra-day household-level load curves. IEEE Trans. Smart Grid **5**(1), 411–419 (2014)
18. M. Ghofrani, M. Hassanzadeh, M. Etezadi-Amoli et al., Smart meter based short-term load forecasting for residential customers, in *North American Power Symposium* (IEEE, 2011), pp. 1–5
19. S. Hochreiter, J. Schmidhuber, Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
20. D.L. Marino, K. Amarasinghe, M. Manic, Building Energy Load Forecasting Using Deep Neural Networks (2016)
21. W. Kong, Z.Y. Dong, Y. Jia et al., Short-term residential load forecasting based on LSTM recurrent neural network. IEEE Trans. Smart Grid **PP**(99), 1–1 (2017)
22. K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition. Comput. Sci. (2014)
23. C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2015), pp. 1–9
24. K. He, X. Zhang, S. Ren et al., Deep Residual Learning for Image Recognition (2015), pp. 770–778
25. J. Zhang, Y. Zheng, D. Qi, Deep Spatio-Temporal Residual Networks for Citywide Crowd Flows Prediction (2016)
26. K. He, X. Zhang, S. Ren et al., Identity mappings in deep residual networks, in *European Conference on Computer Vision* (Springer, Cham, 2016), pp. 630–645
27. Y. Lecun, L. Bottou, Y. Bengio et al., Gradient-based learning applied to document recognition. Proc. IEEE **86**(11), 2278–2324 (1998)
28. J. Long, E. Shelhamer, T. Darrell, Fully convolutional networks for semantic segmentation, in *Computer Vision and Pattern Recognition* (IEEE, 2015), pp. 3431–3440
29. S. Ioffe, C. Szegedy, Batch normalization: accelerating deep network training by reducing internal covariate shift, in *ICML*, pp. 448–456
30. D.P. Kingma, J. Ba, Adam: a method for stochastic optimization. Comput. Sci. (2014)
31. S. Ruder, An Overview of Gradient Descent Optimization Algorithms (2016)
32. J. Duchi, E. Hazan, Y. Singer, Adaptive subgradient methods for online learning and stochastic optimization. J. Mach. Learn. Res. **12**, 2121–2159 (2011)
33. M.D. Zeiler, ADADELTA: an adaptive learning rate method. Comput. Sci. (2012)
34. G. Hinton, N. Srivastava, K. Swersky, RMSProp: divide the gradient by a running average of its recent magnitude, in *Neural Networks for Machine Learning, Coursera lecture 6e* (2012)
35. F. Chollet, Keras (2015). https://github.com/fchollet/keras
36. M. Abadi, A. Agarwal, P. Barham et al., TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems (2016)

Prediction of Permeability Index of Blast Furnace Based on Online Sequential Extreme Learning Machine

Yan Di, Sen Zhang, Xiaoli Su, Yixin Yin and Baoyong Zhao

Abstract Permeability index of the blast furnace is one of the vital monitoring parameters to reflect the operation status of the blast furnace. At present, there are few prediction models for the permeability index at home and abroad. Therefore, this paper proposes to establish a prediction model of the permeability index by using online sequential extreme learning machine (OS-ELM) combined with wavelet analysis, and this paper compares it with the prediction models established by extreme learning machine (ELM), support vector machine (SVM) and BP neural network algorithm. The simulation results show that the prediction model based on OS-ELM has better accuracy than others.

Keywords Blast furnace · Permeability index · Prediction model
Online sequential extreme learning machine

1 Introduction

Blast furnace ironmaking is a highly complex and multivariable production process [1]. The permeability index is a characteristic quantity of the permeability change of the blast furnace column, when permeability index is poor, it will cause the static pressure of the furnace shell to rise. If the permeability is too high, the gas utilization rate will be reduced. The permeability index is one of the significant indicators

Y. Di · X. Su · Y. Yin · B. Zhao
University of Science and Technology Beijing, Beijing 100083, China

S. Zhang (✉)
School of Electrical and Electronic Engineering,
University of Science and Technology Beijing, Beijing, China
e-mail: zhangsen@ustb.edu.cn

Y. Di · X. Su · Y. Yin · B. Zhao
Key Laboratory of Knowledge Automation for Industrial Processes
of Ministry of Education, School of Automation and Electrical Engineering,
University of Science and Technology Beijing, Beijing 100083, China

Y. Jia et al. (eds.), *Proceedings of 2018 Chinese Intelligent Systems Conference*, Lecture Notes in Electrical Engineering 529,
https://doi.org/10.1007/978-981-13-2291-4_37

of measuring the anterograde state of blast furnace. According to the permeability index, the blast furnace operator can find and avoid the abnormal state of the blast furnace, such as hanging and slip, so as to judge blast furnace in time and to operate reasonably according to the change of the blast furnace. From this it can be seen that accurate prediction of permeability index of blast furnace is crucial for blast furnace operators.

The permeability index characterizes the change in the permeability of the blast furnace [2]. The index of the cold air flow and the pressure difference are used to predict the permeability index in the conventional model [3]. However, there are many parameters in the process of blast furnace ironmaking, and there are many non-linear relationships between parameters. It is difficult to accurately characterize the permeability index only by using cold air flow and pressure difference. And this will affect the accuracy of prediction and cannot meet actual production requirements.

Online Sequential Extreme Learning Machine (OS-ELM) is a fast single hidden layer neural network algorithm [4]. In the initial part, the weights from the hidden layer to the output layer are learned through a small number of samples by using ELM. The second part is online learning. The output weights learned in the initial part are updated through a single sample or data block sequentially in this phase. Compared with the traditional learning algorithm, it not only considers the timeliness of the data, but also has the features of fast computing speed, strong generalization ability, and does not fall into the local minimum. Therefore, it has been widely used in various fields [5, 6].

This paper uses the measured data to establish the model of permeability index based on the OS-ELM algorithm. Firstly, in order to improve the accuracy of the model, this paper analyses the factors affecting the permeability index according to the relevant mechanism of the blast furnace, and given to blast furnace production data contain noise, this paper uses wavelet transform to process the production data. Then the model is compared with the model established by ELM, SVM and BP [7, 8]. The experimental results show that the prediction model established by using OS- ELM algorithm can accurately predict the permeability index in real time. This model is more suitable for solving the problems in this paper and provides support for the subsequent operation of the blast furnace.

2 Selection of Input Parameters of Model and Data Processing

2.1 The Permeability Index Factors Analysis

The parameters of the blast furnace ironmaking process are numerous and interact with each other. Furnace foremen who have experience not only use air volume, wind pressure, and top pressure to predict the permeability index, but also commonly