



Neural Networks with Improved Extreme Learning Machine for Demand Prediction of Bike-sharing

Fan Wu¹ · Si Hong¹ · Wei Zhao^{2,3} · Xiaoyan Wang⁴ · Xun Shao⁵  · Xiujun Wang^{2,3} · Xiao Zheng^{2,3}

Accepted: 17 February 2021 / Published online: 25 March 2021
© The Author(s) 2021

Abstract

Accurate demand prediction of bike-sharing is an important prerequisite to reducing the cost of scheduling and improving the user satisfaction. However, it is a challenging issue due to stochasticity and non-linearity in bike-sharing systems. In this paper, a model called pseudo-double hidden layer feedforward neural networks is proposed to approximately predict actual demands of bike-sharing. Specifically, to overcome limitations in traditional back-propagation learning process, an algorithm, an extreme learning machine with improved particle swarm optimization, is designed to construct learning rules in neural networks. The performance is verified by comparing with other learning algorithms on the dataset of Streeter Dr bike-sharing station in Chicago.

Keywords Demand prediction · Bike-sharing · pseudo-double hidden layer feedforward neural networks · Extreme learning machine · Particle swarm optimization

1 Introduction

With the development of sharing economy, bike-sharing systems have rapidly emerged in major cities all over the world. Bike-sharing can be described as a short-term bike rental service for inner-city transportation providing bikes at unattended stations. It has become one of the most important low-carbon travel ways. Compared with traditional rental service, bike-sharing will not be limited by the boxes at bike-stations. It provides convenient services, but generates complicated problems. For instance, the layout of bike-sharing stations is flexible and the capacities of stations is not fixed, leading to big fluctuate demands for stations. Some new characteristics are exhibited, such as the uneven distribution of user demands in time and space.

Accurate demand prediction of bike-sharing can effectively improve user experience and enhance brand competence, which was elaborated in [2]. El-Assi et al. [3] and

Ermagun et al. [4] elaborated the main problems and difficulties in the demand prediction of bike-sharing. Solutions can be mainly divided into two types. The traditional one is based on statistical analysis. Yang et al. [5] proposed a semi-parametric geographically weighted regression method to estimate a bike-sharing trip using location-based social network data. [6] proposed a method combining bootstrapping and subset selection that utilized partially useful information in each bike-sharing station. It can solve problems in which data cleaning approaches failed due to the lack of original data.

The other method is based on artificial neural networks. Yang et al. [7] proposed convolutional neural networks to predict daily demands of bike-sharing at both city and station levels. Lin et al. [8] proposed graph convolutional neural networks with data-driven graph filter model. The heterogeneities of demands among different bike-sharing stations were also discussed. Xu et al. [9] developed a dynamic demand prediction model based on a deep learning approach with large-scale datasets. The comparison results suggested that prediction accuracy of long-short term memory neural networks was better than statistical models and advanced machine learning methods. Chang et al. [10] developed a prediction framework integrating artificial immune system and neural networks. The performance is verified by comparing with other models. Feng et al. [11] discussed the Markov chain population model to predict bike demands

Part of this work [1] has been submitted to 10th EAI International Conference on Mobile Networks and Management (EAI MONAMI 2020).

✉ Xun Shao
x-shao@ieee.org

Extended author information available on the last page of the article.

among different travel stations. Kim [12] studied the influence of weather conditions and time characteristics on demands of bike-sharing. Furthermore, deep learning methods and comprehensive methods with heuristic algorithms were applied in various engineering projects [13–16], but rarely applied for the demand prediction of bike-sharing.

In addition, those methods have some limitations. Increasing the number of hidden layers is a feasible approach to achieve the certain prediction accuracy, which may lead to overfitting phenomena and reduce the generalization performance of prediction models. In addition, accelerating gradient descent can improve the convergence rate, while it leads to an unstable generalization. Considering these limitations, a novel neural network model is proposed, which is called pseudo-double hidden layer feedforward neural networks. In this paper, an algorithm, extreme learning machine with improved particle swarm optimization, is proposed to tune weights and biases in neural network to improve prediction accuracy. Finally, experiments are performed on the dataset of Streeter Dr bike-sharing station in Chicago to verify the effectiveness of the model proposed.

2 Pseudo-double hidden layer feedforward neural networks

2.1 Network structure

Pseudo-double hidden layer feedforward neural networks (PDLFNs) are biologically inspired computational models, which consist of processing elements (or neurons) and connections between them with coefficients. The structure of PDLFNs is different from the single hidden layer feedforward neural networks (SLFNs) and the double hidden layer feedforward neural networks (DLFNs). As shown in Fig. 1, it contains one input layer, pseudo-double hidden layers, and one output layer. The hidden layers consist of layer V and layer B. In SLFNs and DLFNs, the hidden layer is the collection of neurons with activation functions as well as provides one or two intermediate layers between the input layer and the output layer. While in PDLFNs, layer V is a special hidden layer. There is only one neuron with a smooth function in this layer. Thus, PDLFNs can directly process original sample data to produce the final results.

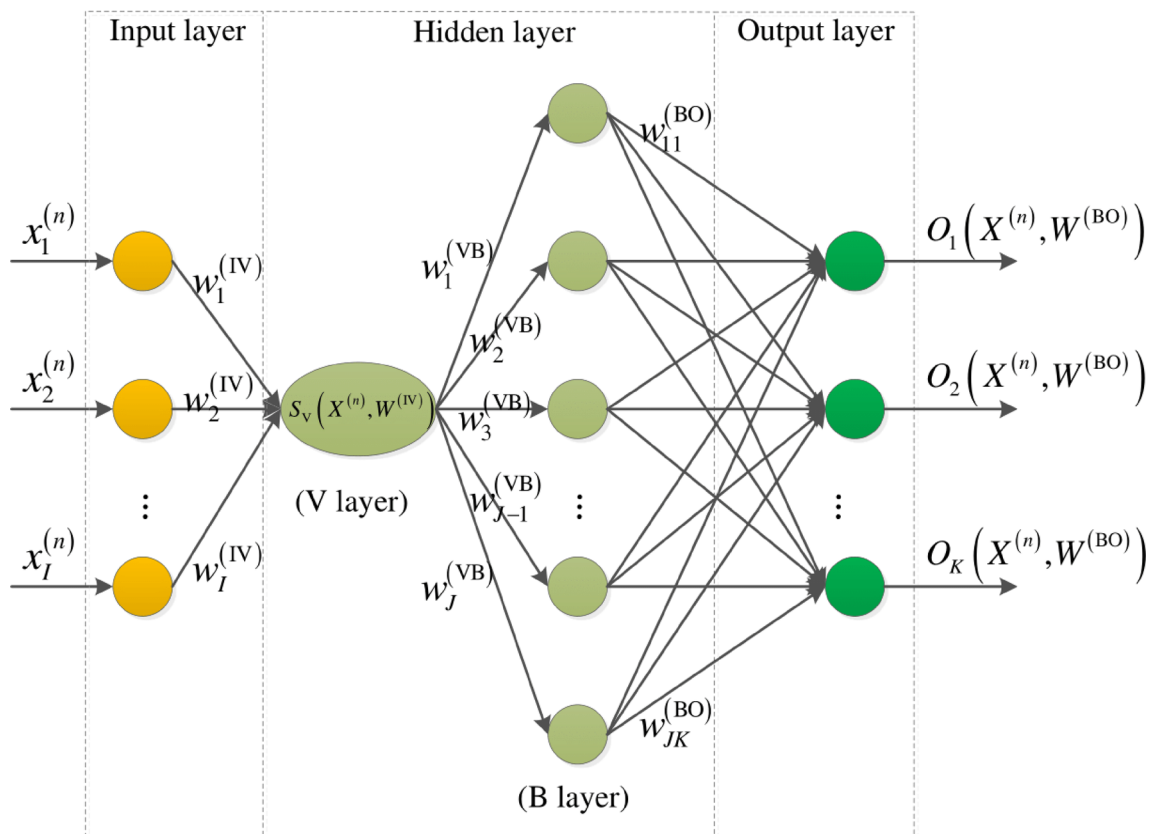


Fig. 1 PDLFNs structure

The design consideration of V layer mainly comes from the following two results. In the first place, multiple hidden layers can reach high prediction accuracy even by setting fewer neurons in each hidden layer [17]. In addition, there is some noise disturbance in sample data, which can be reduced by smooth processes [18].

Without loss of generality, assume that the numbers of neurons in the input layer, layer B and output layer are I , J and K , respectively. There are N samples inputted into PDLFNs, commonly in the shape of means of multivariate time series. Each sample contains I -dimensional data. Mathematically, the n -th ($1 \leq n \leq N$) sample is represented by the vector $X^{(n)} = (x_1^{(n)}, x_2^{(n)}, \dots, x_I^{(n)})^T$, where $x_i^{(n)}$ ($1 \leq i \leq I$) is the data presented to the i -th neuron in the input layer. The corresponding weight vector of the input layer to layer V is denoted as $W^{(IV)} = (w_1^{(IV)}, w_2^{(IV)}, \dots, w_I^{(IV)})^T$, where $w_i^{(IV)}$ specifies the influence between the i -th neuron in the input layer and the neuron in layer V.

Compared with DLFNs, the neuron of layer V in PDLFNs is no longer with a bias value and an activation function, but with a smooth function. The smooth function, denoted as $S_V(X^{(n)}, W^{(IV)})$, can take corresponding forms according to characteristics of sample datasets, such as moving average functions, exponential smoothing functions, autoregressive functions, and adaptive filtering functions.

$$\begin{aligned} & O_k(X^{(n)}, W^{(BO)}) \\ &= A_O \left(\sum_{j=1}^J (w_{jk}^{(BO)} B_j(X^{(n)}, W^{(VB)})) + b_k^{(O)} \right) \\ &= A_O \left(\sum_{j=1}^J (w_{jk}^{(BO)} A_B(w_j^{(VB)} S_V(X^{(n)}, W^{(IV)}) + b_j^{(B)})) + b_k^{(O)} \right), \quad (1) \end{aligned}$$

The rest part of PDLFNs is similar to traditional feedforward neural networks, where each neuron is assigned

a bias and the output of each neuron is an activation function. The summation in each neuron includes a bias for lowering or raising its input to the activation function such as linear function, sigmoid function and hard limit function. It is worth mentioning that the activation functions for the neurons in the same layer are always the same. The weight vector between layer V and layer B is denoted as $W^{(VB)} = (w_1^{(VB)}, w_2^{(VB)}, \dots, w_J^{(VB)})^T$, where $w_j^{(VB)}$ ($1 \leq j \leq J$) is the weight of the connection between the neuron in layer V and the j -th neuron in layer B. The bias for the j -th neuron in layer B is denoted as $b_j^{(B)}$. The output of the j -th neuron in layer B is denoted as $B_j(X^{(n)}, W^{(VB)})$ in Eq. 2, where $A_B(x)$ is the activation function for the neurons in layer B.

$$B_j(X^{(n)}, W^{(VB)}) = A_B(w_j^{(VB)} S_V(X^{(n)}, W^{(IV)}) + b_j^{(B)}). \quad (2)$$

$$O = (O_1(X^{(n)}, W^{(BO)}), O_2(X^{(n)}, W^{(BO)}), \dots, O_K(X^{(n)}, W^{(BO)}))^T. \quad (3)$$

Finally, the outputs of PDLFNs can be represented by the vector in Eq. 3. $O_k(X^{(n)}, W^{(BO)})$ ($1 \leq k \leq K$) is the output of the k -th neuron in the output layer from Eq. 1, where $W^{(BO)}$ is the weight matrix between layer B and the output layer in Eq. 4, $w_{jk}^{(BO)}$ the weight of the connection between the j -th neuron in layer B and the k -th neuron in the output layer, $b_k^{(O)}$ the bias for the k -th neuron in the output layer, $A_O(x)$ the activation function for the neurons in the output layer.

$$W^{(BO)} = \begin{bmatrix} w_{11}^{(BO)} & w_{12}^{(BO)} & \dots & w_{1K}^{(BO)} \\ w_{21}^{(BO)} & w_{22}^{(BO)} & \dots & w_{2K}^{(BO)} \\ \vdots & \vdots & \ddots & \vdots \\ w_{J1}^{(BO)} & w_{J2}^{(BO)} & \dots & w_{JK}^{(BO)} \end{bmatrix}. \quad (4)$$

$$v_{md}^{t+1} = \begin{cases} \left(w_{\max} - \frac{t(w_{\max} - w_{\min})}{T_{\max}} \right) v_{md}^t + c_1 r_1^t (pbest_{md} - z_{md}^t) + c_2 r_2^t (gbest_d - z_{md}^t), & 0 \leq t < \frac{2T_{\max}}{3}, \\ \lambda v_{md}^t + c_1 r_1^t (pbest_{md} - z_{md}^t) + c_2 r_2^t (gbest_d - z_{md}^t), & \frac{2T_{\max}}{3} \leq t \leq T_{\max}, \end{cases} \quad (5)$$

2.2 Learning algorithm

The traditional learning algorithms in the feedforward neural networks are mainly based on the gradient descent methods. The back-propagation (BP) learning algorithm is a representative one, where gradients can be computed efficiently by propagation from the output to the input. It is one of the most successful and widely popular learning algorithms for training neural networks in recent years. However, several limitations arise, such as not easy to determine network structures and learning rates, unstable convergence results, and time-consuming learning.

To resolve the issues above related with gradient-based algorithms, Guangbin Huang [19] proposed an efficient learning algorithm called extreme learning machine (ELM) for feedforward neural networks, especially for SLFNs. ELM is equipped with several salient features different from traditional popular gradient-based learning algorithms.

1. Ease of use. No parameters need to be manually tuned during the iterative procedure except for the predefined network architecture. The number of neurons in the hidden layer is equal to or approximately equal to the number of samples by default.

2. Fast learning speed. Most training can be completed usually in within minutes.
3. High generalization performance. It obtains better generalization performance than gradient-based learning algorithms in most cases.
4. Suitable for nonlinear activation functions. Almost all piecewise continuous functions can be used as activation functions.

ELM avoids some difficulties in gradient-based learning algorithms, such as to determine stopping criteria, learning rates and the number of learning epochs, and to find local minima. However, it is also found that ELM tends to require more hidden neurons than traditional gradient-based learning algorithms as well as result in ill-condition problems due to random determination of the input weights and hidden biases. Considering these limitations of ELM, an improved particle swarm optimization (IPSO) is proposed to optimize input weights and hidden biases.

2.2.1 Improved particle swarm optimization

Particle swarm optimization (PSO) is one of the most representative meta-heuristic optimization algorithms. It mimics the social behavior of organisms, such as birds in a flock or fish in a school, which grants them surviving advantages.

Considering a swarm with M particles in a D -dimensional search space, there is a position vector $Z_m^t = (z_{m1}^t, z_{m2}^t, \dots, z_{mD}^t)^T$ ($1 \leq m \leq M$) and a velocity vector $V_m^t = (v_{m1}^t, v_{m2}^t, \dots, v_{mD}^t)^T$ for the m -th particle after the t -th iteration. Z_m^0 and V_m^0 are the initial position and velocity of the m -th particle, respectively. The best position of the m -th particle is denoted as $PBEST_m = (pbest_{m1}, pbest_{m2}, \dots, pbest_{mD})^T$, and the best position of all particles as $GBEST = (gbest_1, gbest_2, \dots, gbest_D)^T$. In classical version of PSO, the position and velocity vectors are updated according to Eqs. 6 and 7.

$$v_{md}^{t+1} = wv_{md}^t + c_1 r_1^t (pbest_{md} - z_{md}^t) + c_2 r_2^t (gbest_d - z_{md}^t), \quad (6)$$

$$z_{md}^{t+1} = z_{md}^t + v_{md}^{t+1}, \quad (7)$$

where $1 \leq d \leq D$; w , c_1 , c_2 , r_1^t and r_2^t are respectively inertia weight constant, two acceleration constants with positive values, and two uniform random parameters within $[0, 1]$. The final solution of PSO is sensitive to these control parameters.

The focus of the improved approaches has revolved around adapting the inertia weight w . It is important for balancing the global search, also known as exploration, and local search, known as exploitation.

In order to make the algorithm converge to the global optimal values more quickly and effectively, a comprehensively improved method is proposed. It is an adaptive PSO algorithm combined with the compression

factor. The velocity vectors are updated according to Eq. 5, where $\lambda = 2/|2 - \beta - \sqrt{\beta(\beta - 4)}|$ is the compression factor given $\beta = c_1 + c_2$. T_{\max} , w_{\max} and w_{\min} are the maximum iteration number, the initial inertial weight and the final inertial weight, respectively. We can ensure that this algorithm is equipped with a strong global search ability in the early iterative stage by the adaptive inertia weight in the first part in Eq. 5. Correspondingly, we can also ensure that this algorithm is equipped with a local sophisticated search ability by the compression factor in the second part.

2.2.2 Improved extreme learning machine

In this paper, ELM combining with IPSO (IPSO-ELM) is proposed as the learning algorithm of PDLFNs in Fig. 2. It

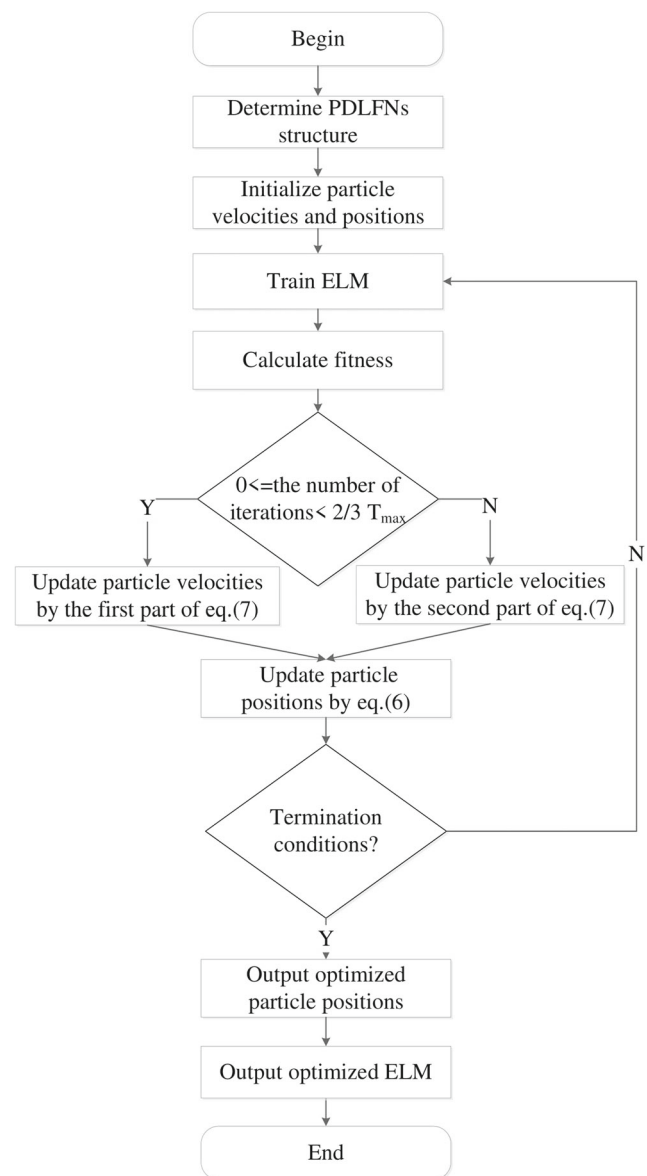


Fig. 2 IPSO-ELM learn process

applies IPSO to optimize input weights and hidden biases to improve generalization abilities.

3 Demand prediction model

3.1 Prediction periods

Self-regulating abilities in bike-sharing systems can generally meet users' rental demands during flat hump periods and low peak periods. However, it does not hold for peak periods. Meanwhile, the users' rental during peak periods is one of the main factors influencing scheduling schemes. Thus, we only discuss the demand prediction problem during peak periods in this paper.

3.2 Demand Prediction Model

The demand prediction model of bike-sharing is shown in Fig. 3. In this model, the numbers of neurons in the input layer, layer B and output layer are 3, 29 and 1, respectively. The bike rental demands from the $(n - 3)$ -th day to the $(n - 1)$ -th day are inputted into the input layer. Weighted

moving average function (WMA) in Eq. 8 is applied as the smooth function for neurons in layer V.

$$S_V(X^{(n)}, W^{(IV)}) = \frac{\sum_{i=1}^3 x_i^{(n)} w_i^{(IV)}}{\sum_{i=1}^3 w_i^{(IV)}}. \quad (8)$$

Hyperbolic tangent sigmoid function in Eq. 9 and linear function in Eq. 10 are applied as the activation functions for neurons in layer B and the output layer, respectively.

$$B_j(X^{(n)}, W^{(VB)}) = \frac{1 - e^{-2(w_j^{(VB)} S_V(X^{(n)}, W^{(IV)}) + b_j^{(B)})}}{1 + e^{-2(w_j^{(VB)} S_V(X^{(n)}, W^{(IV)}) + b_j^{(B)})}}, \quad (9)$$

$$O(X^{(n)}, W^{(BO)}) = \sum_{j=1}^{29} w_j^{(BO)} B_j(X^{(n)}, W^{(VB)}). \quad (10)$$

There is only one neuron in the output layer. Thus, $W^{(BO)}$, the weight matrix of layer B to the output layer, is degenerated to the weight vector in this model. $O(X^{(n)}, W^{(BO)})$, from the neuron in the output layer of PDLFNs, is the prediction demand in the n -th day.

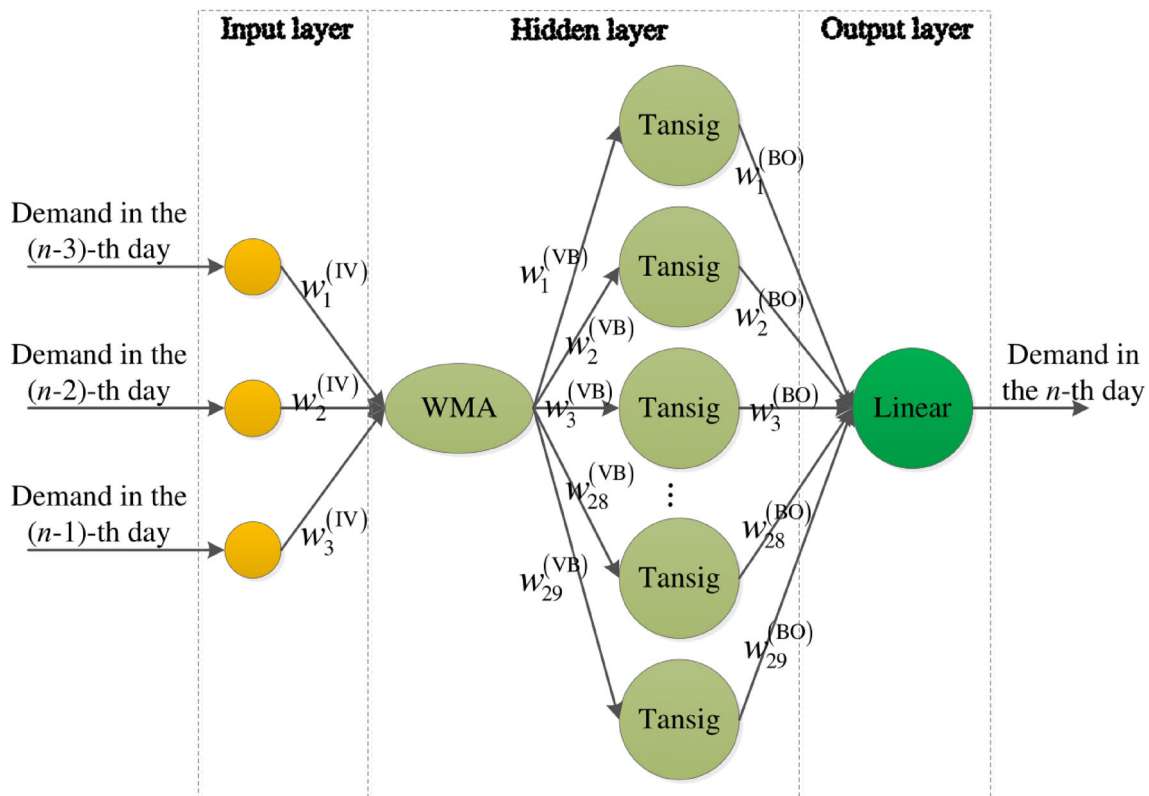


Fig. 3 Demand prediction model of bike-sharing based on PDLFNs

3.3 Evaluation criteria

In order to verify the effectiveness of the proposal, mean square error MSE and square correlation coefficient R^2 , are selected as the evaluation criteria.

$$MSE = \frac{1}{N} \sum_{n=1}^N (r_n - p_n)^2,$$

$$R^2 = \frac{\left(\sum_{n=1}^N (p_n - \bar{p})(r_n - \bar{r}) \right)^2}{\sum_{n=1}^N (p_n - \bar{p})^2 \sum_{n=1}^N (r_n - \bar{r})^2},$$

where p_n , r_n , \bar{p} and \bar{r} are predicted bike rental demands in the n -th day, recorded demands in the n -th day, average demands throughout the whole period, and recorded demands throughout the whole period, respectively.

4 Demand prediction of streeter dr bike-sharing station

4.1 Load and prepare data

The dataset in this paper is from the official website of Streeter Dr bike-sharing station in Chicago. We split the data into two sets, one for training and the other for testing. The recorded bike rental demands during 92 days from March 1st, 2017, to May 31st, 2017 are aggregated as the training set. And the recorded demands during the next 30 days are aggregated as the test set.

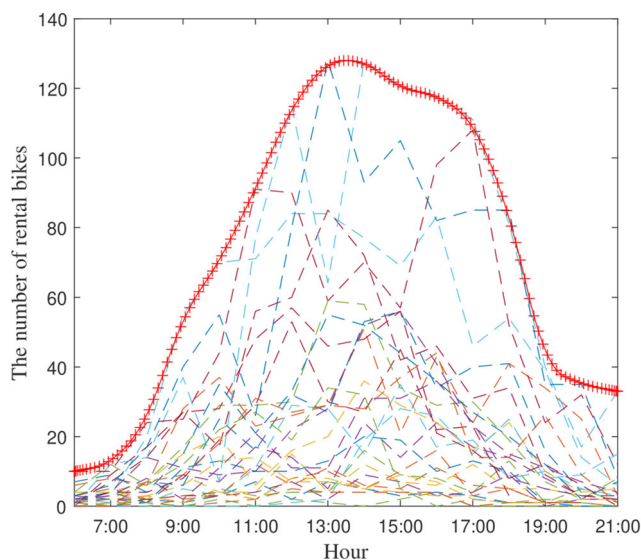
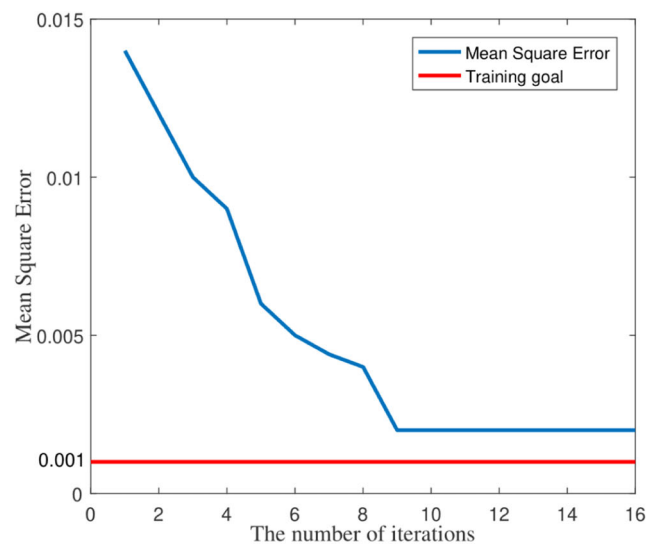


Fig. 4 The number of rental bikes in March

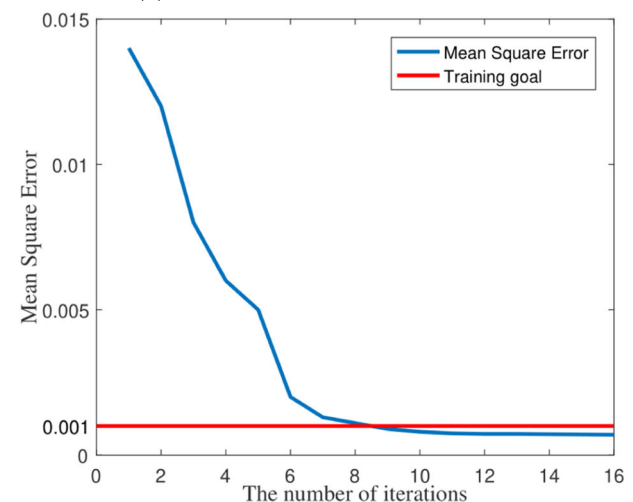
According to the regularities in the dataset, we can find that the peak periods of renting bikes are always in the same interval. Taking March for instance, the number of rental bikes is shown in Fig. 4. The peak periods of renting bikes are mainly distributed from 13 p.m. to 17 p.m.

4.2 Prediction results

To verify the performance of IPSO, $MSEs$ before and after the improvement are shown Fig. 5. The global search ability of particle swarm is effectively improved by optimizing the particles' velocities, by adjusting inertia weight in the early iterative stage and adding a compression factor in the later

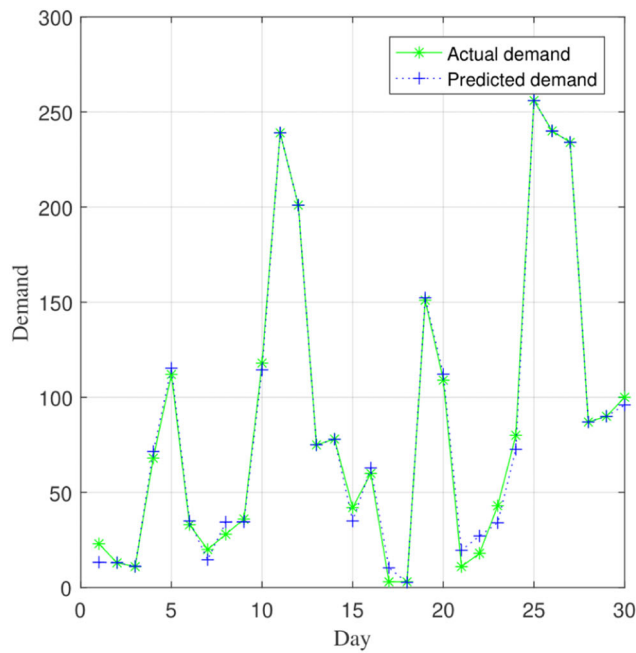


(a) Mean square error of PSO-ELM

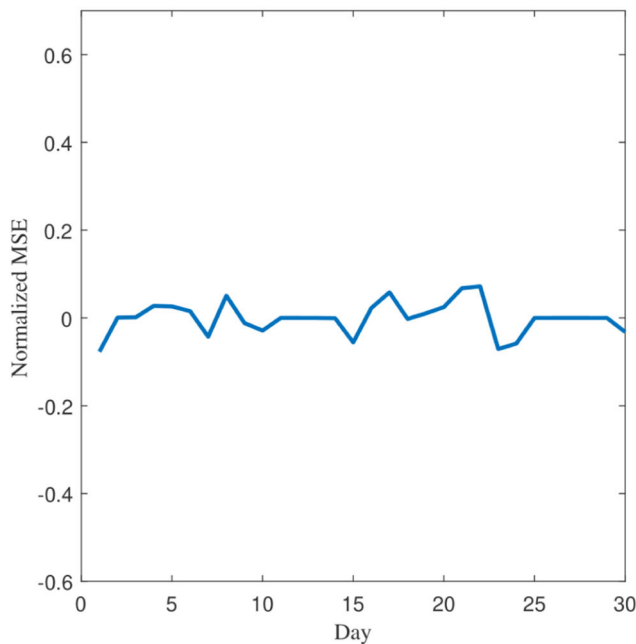


(b) Mean square error of IPSO-ELM

Fig. 5 Mean square errors of two learning algorithms



(a) Prediction accuracy

(b) Normalized MSE **Fig. 6** Prediction results of PDLFNs with IPSO-ELM

stage. IPSO-ELM is capable of jumping out of local optima and finding a solution in the final stage. The prediction results are shown in Fig. 6.

Table 1 MSE and R^2

		SLFNs	DLFNs	PDLFNs
ELM	MSE	535.049	327.241	48.057
	R^2	0.829	0.908	0.907
PSO-ELM	MSE	459.260	272.223	41.867
	R^2	0.883	0.913	0.984
IPSO-ELM	MSE	373.145	204.323	33.025
	R^2	0.902	0.935	0.992

4.3 Comparison and discussion

In order to verify the effectiveness of the prediction model in this paper, comparative experiments are conducted, considering 3 different network structures (SLFNs, DLFNs and PDLFNs) with 3 different learning algorithms (ELM, PSO-ELM and IPSO-ELM). The evaluation criteria are MSE and R^2 proposed in the preceding section. The experiment results of 9 prediction models are shown in Table 1.

The longitudinal comparison is among the same network structure with different learning algorithms. MSE and R^2 in IPSO-ELM are always the minimum and maximum in LSFNs, DLFNs and PDLFNs. It suggests that IPSO-ELM is the best learning algorithm among these 3 algorithms. The neural networks with IPSO-ELM can obtain the most accurate prediction results. The horizontal comparison among different network structures with the same learning algorithm shows that PDLFNs can obtain the most accurate prediction results. From the perspective of MSE and R^2 , PDLFNs with IPSO-ELM is the best prediction model among 9 models, and the prediction accuracy improved by changing the network structure is greater than changing the learning algorithm. The effectiveness in other experiment results is shown in Figs. 7 and 8.

5 Conclusion

Aiming at predicting the demands of bike-sharing, this paper constructs the PDLFNs model consisting of “input layer - V layer - B layer - output layer” and improves ELM combining with IPSO as learning algorithm. This model has two advantages verified by the comparative experiments on predicting the demands of Streeter Dr bike-sharing station. In the first place, a simple network structure is equipped with stable generalization and high accuracy. In addition, improved ELM is an effective learning algorithm of feedforward neural networks besides SLFNs, which

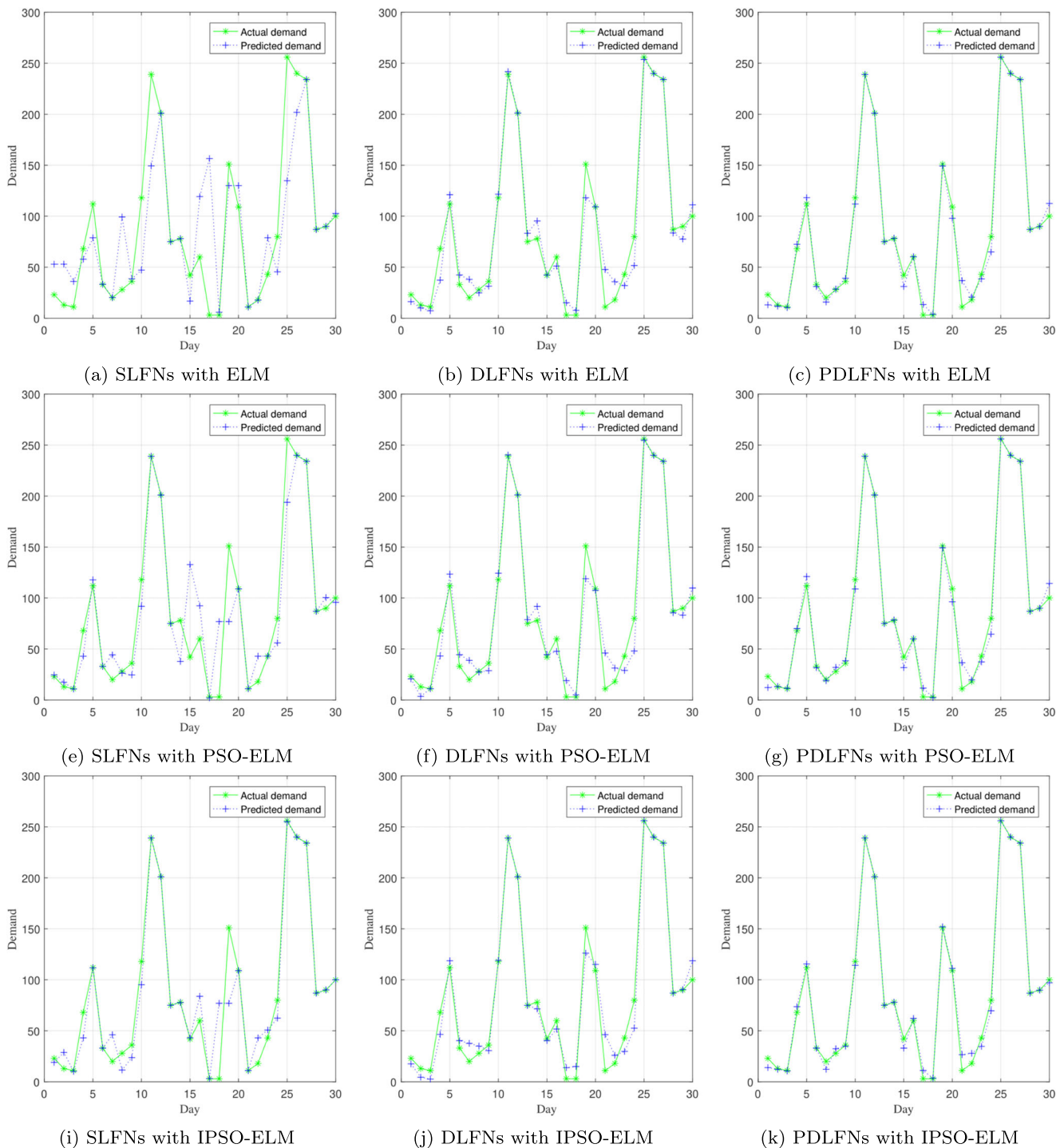


Fig. 7 Prediction results in different models

optimizes the selection process of the input weights and hidden biases. From the comparative experiments, we have verified effectiveness on the prediction accuracy.

In this paper, we have predicted the demands of bike-sharing during peak periods, which are main factors

influencing scheduling schemes. Nevertheless, optimal scheduling schemes are still influenced by the demands during flat hump periods and low peak periods. We will consider this issue in the future and predict 24-hour demands in the bike-sharing system.

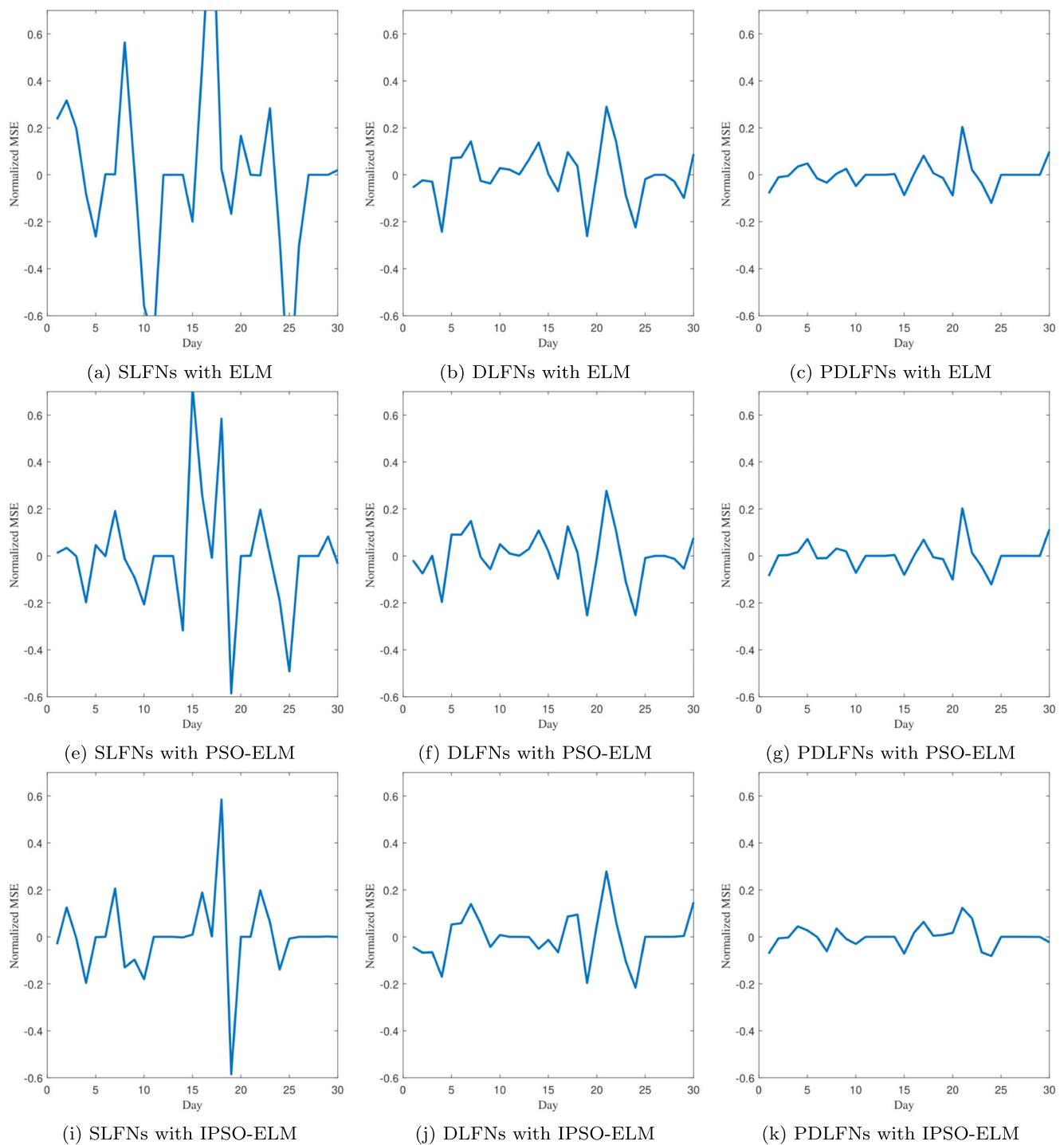


Fig. 8 Normalized MSE in different models

Acknowledgements This work was supported in part by the National Natural Science Foundation of China under Grant 61702006, and in part by the Program for Synergy Innovation in the Anhui Higher Education Institutions of China under Grant GXXT-2019-025. Part of this work was carried out under the Cooperative Research Project Program of the Research Institute of Electrical Communication, Tohoku University, and under the Telecommunications Advancement Foundation, Japan.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not


included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Fan W, Si H, Wei Z, Xiao Z, Wen Q, Xun S (2020) A novel neural network model for demand prediction of bike-sharing
2. Xu H, Duan F, Pu P (2019) Dynamic bicycle scheduling problem based on short-term demand prediction. *Appl. Intell.* 49:1968–1981
3. El-Assi W, Mahmoud MS, Habib KN (2017) Effects of built environment and weather on bike sharing demand: a station level analysis of commercial bike sharing in Toronto. *Transportation* 44:589–613
4. Ermagun A, Lindsey G, Loh TH (2018) Bicycle, pedestrian, and mixed-mode trail trac: A performance assessment of demand models. *Landsc. Urban Plan.* 177:92–102
5. Yang F, Ding F, Qu X, Ran B (2019) Estimating urban shared-bike trips with location-based social networking. *Data Sustainability*, vol 11
6. Cheng P, Hu J, Yang Z, Shu Y, Chen J (2019) Utilization-aware trip advisor in bike-sharing systems based on user behavior analysis. *IEEE Trans. Knowl. Data Eng.* 31:1822–1835
7. Yang H, Xie K, Ozbay K, Ma Y, Wang Z (2018) Use of deep learning to predict daily usage of bike sharing systems. *Transportation Research Record Journal of the Transportation Research Board*
8. Lin L, He Z, Peeta S (2018) Predicting station-level hourly demand in a large-scale bike sharing network: A graph convolutional neural network approach. *Transp. Res. Part C Emerg. Technol.* 97:258–276
9. Xu C, Ji J, Liu P (2018) The station-free sharing bike demand forecasting with a deep learning approach and large-scale datasets. *Transp. Res. Part C Emerg. Technol.* 95:47–60
10. Chang PC, Wu JL, Xu Y, Zhang M, Lu XY (2019) Bike sharing demand prediction using artificial immune system and artificial neural network. *Soft. Comput.* 23:613–626
11. Feng C, Hillston J, Reijnders D (2017) Moment-based availability prediction for bike-sharing systems. *Perform. Eval.* 117:58–74
12. Kim K (2018) Investigation on the effects of weather and calendar events on bike-sharing according to the trip patterns of bike rentals of stations. *J. Transp. Geogr.* 66(Jan):309–320
13. Benkedjouh T, Medjaher K, Zerhouni N, Rechak S (2015) Health assessment and life prediction of cutting tools based on support vector regression. *J. Intell. Manuf.* 26:213–223
14. Cao YJ, Jia LL, Chen YX, Lin N, Yang C, Zhang B, Liu Z, Li XX, Dai HH (2019) Recent advances of generative adversarial networks in computer vision. *IEEE Access* 7:14985–15006
15. Hu H, Liu Z, An J (2020) Mining mobile intelligence for wireless systems: a deep neural network approach. *IEEE Comput. Intell. Mag.* 15:24–31
16. Wu Q, Ding K, Huang B (2018) Approach for fault prognosis using recurrent neural network. *J Intell Manuf.* 1–13
17. Tamura S, Tateishi M (1997) Capabilities of a four-layered feedforward neural network: Four layers versus three. *IEEE Trans. Neural Netw.* 8(2):251–255
18. Han J, Kamber M, Pei J (2011) Data mining: concepts and techniques. *Data Mining Concepts Models Methods & Algorithms*, Second Edition 5(4):1–18
19. Huang G, Huang GB, Song S, You K (2015) Trends in extreme learning machines: a review. *Neural Netw.* 61:32–48

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Affiliations

Fan Wu¹ · Si Hong¹ · Wei Zhao^{2,3} · Xiaoyan Wang⁴ · Xun Shao⁵  · Xiujun Wang^{2,3} · Xiao Zheng^{2,3}

Fan Wu
dragonwufan@126.com

Si Hong
Sihong2019@163.com

Wei Zhao
zhaoweistuart@gmail.com

Xiaoyan Wang
xiaoyan.wang.shawn@vc.ibaraki.ac.jp

Xiujun Wang
wxj@mail.ustc.edu.cn

Xiao Zheng
Xzheng@ahut.edu.cn

- ¹ School of Management Science and Engineering, Anhui University of Technology, Anhui, China
- ² School of Computer Science and Technology, Anhui University of Technology, Anhui, China
- ³ Anhui Engineering Laboratory for Intelligent Applications and Security of Industrial Internet, Maanshan 243032, China
- ⁴ Graduate School of Science and Engineering, Ibaraki University, Ibaraki, Japan
- ⁵ School of Regional Innovation and Social Design Engineering, Kitami Institute of Technology, Kitami, Japan