

A Bifurcation Deep Neural Network for Electricity Meter Error Prediction under Actual Conditions

Zichao Kou, Yanjun Fang*, Laurent Bleszinski

School of Electrical Engineering, Wuhan University, Wuhan 430072, China

Corresponding Author Email: yjfang@whu.edu.cn

<https://doi.org/10.18280/ejee.210604>

ABSTRACT

Received: 10 May 2019

Accepted: 19 August 2019

Keywords:

*convolutional neural network (CNN),
autoencoder, measuring errors,
electricity meters*

The environmental conditions have a great impact on the measuring accuracy of electricity meters, once they are installed. This paper aims to find a way to accurately evaluate the measuring errors of electricity meters under actual conditions. Specifically, a novel bifurcation deep neural network (BDNN) model was designed and tested. The BDNN consists of a subnetwork and a fully-connected network. The subnetwork is a deep autoencoder-convolutional neural network (DAE-CNN) dedicated to processing harmonic features. The fully-connected network takes the subnetwork output and the environmental conditions as its inputs, and generates the output of the entire model by softmax. Then, the BDNN was trained on a dataset generated by real experiments with electricity meters. Three hyperparameters, namely, the activation function, the number of hidden layers and the autoencoder structure, were optimized through several experiments. Through the optimization, the rectified linear unit (ReLU) was adopted as the activation function, the number of hidden layers was set to 4, and the autoencoder structure was determined as 256-128-64-32. Each numerical figure refers to the number of nodes in the corresponding hidden layer. Finally, the BDNN was compared with the least squares support vector machine (LS-SVM), the fully-connected MLP (FCP) and the original CNN, and found to outshine the contrastive methods in prediction error and computing cost. The research results shed important new light on the field calibration and error prediction of electricity meters.

1. INTRODUCTION

The effective calibration of electricity meters is a growing concern among metrologists, who have sought continually to improve the accuracy of different measurement approaches. This has been accentuated by new international directives highlighting the need for measuring instruments that achieve designed functions in actual conditions, and adopt highly traceable measuring methods [1, 2]. Meanwhile, the development of smart grids has raised the concern with the precision of state estimation for electricity meters across distributed management systems. Together, these concerns have motivated the power industry to find effective ways to ensure the accuracy of electricity meters [3]. To date, the relevant studies have focused on creating error models for electricity meters, and predicting errors under dynamic conditions. However, there is not yet a comprehensive model for the errors of electricity meters. For electricity meters with multiple components, the error situation in multi-dimensional conditions remains a black box problem.

Fortunately, there is another way to improve the measuring accuracy of electricity meters [4]: treating the error estimation as a regression and forecasting problem [5]. In this way, a different set of techniques can be brought to bear, many of which are grounded on artificial intelligence (AI) [6, 7]. The rise of the AI has given birth to various effective forecasting models [8, 9].

The deep learning approaches have a unique advantage in

error estimation of electricity meters under actual conditions [10], namely, the algorithms are trained by real-world data [11]. Once enough data has been acquired, machine learning will take over for further error prediction [12]. Thus, the data collection task becomes scalable, facilitating the management of diverse and complex errors that may occur [13].

Therefore, this paper explores on how the errors of electricity meters are affected by the various changes in operating conditions, and, on this basis, establishes an error estimation method using a specially-designed artificial neural network (ANN). This paper offers three major contributions:

(1) A novel bifurcation deep neural network (BDNN) model was designed to estimate the errors of electricity meters under actual conditions. When there are known environmental parameters, this model can accurately determine the measuring accuracy of metering devices under specific environment conditions and certain constraints. This is the first time that machine learning has been specifically applied to the error prediction of electricity meters.

(2) To verify its performance, our BDNN model, plus various machine learning approaches, was applied to the evaluate initial measurement data. The comparison shows that our approach is an effective strategy for error estimation of electricity meters.

(3) Our approach offers a novel yet viable solution to estimation of measuring errors of electricity meters under actual conditions.

2. MODEL CONSTRUCTION

Deep learning has been successfully applied in many domains [14, 15]. It has clear potential to assist with the error estimation for electricity meters [16]. This section explains the construction of the BDNN model, and provides some background information about our approach [17].

As shown in Figure 1, the proposed BDNN consists of a subnetwork and a fully-connected network [18]. The subnetwork is a deep autoencoder-convolutional neural network (DAE-CNN) dedicated to processing harmonic features. The fully-connected network takes the subnetwork output and the environmental conditions as its inputs, and generates the output of the entire model by softmax.

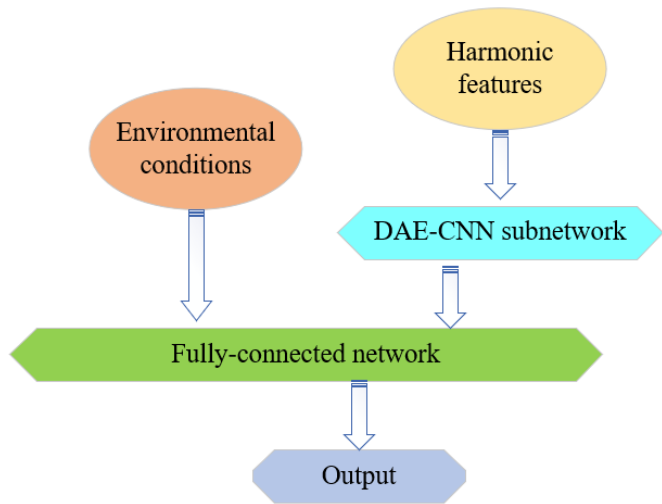


Figure 1. The architecture of the BDNN

The BDNN has two types of inputs, namely, environmental conditions (A) and harmonic features (B). The environmental conditions include temperature, humidity, voltage, current, and power factor, while the harmonic features refer to the frequency, amplitude and phase angle of each harmonic in the test system [19]. The BDNN needs to learn how the measuring errors of electricity meters are influenced by the harmonic features. This learning task is similar to the training in natural language processing (NLP) and image recognition [20]. That is why a DAE-CNN (Figure 2) was specifically designed as the subnetwork to process the harmonic features, and output a vector [21-23]. The output vector, together with the environmental conditions, was received by the fully-connected network, which then output the predicted changes of measuring errors.

The DAE-CNN is essentially an unsupervised pre-training machine, namely, the deep autoencoder (DAE), superimposed on the basic CNN. As shown in Figure 2, the autoencoder encodes the inputs as a reduced-dimensional map through convolution operations. Then, the reduced-dimensional map is decoded through deconvolution operations back to the original inputs. In this way, the reconstruction error is minimized, and the network weights are preadjusted through the training [24, 25]. Finally, a backpropagation algorithm is adopted to finetune the parameters of the pretrained network. Each layer in the DAE-CNN is an autoencoder, consisting of an encoder and a decoder. Each encoder contains three 3×3 convolutional layers, followed by a max-pooling layer, while each decoder contains a max-pooling layer, followed by three 3×3 convolutional layers.

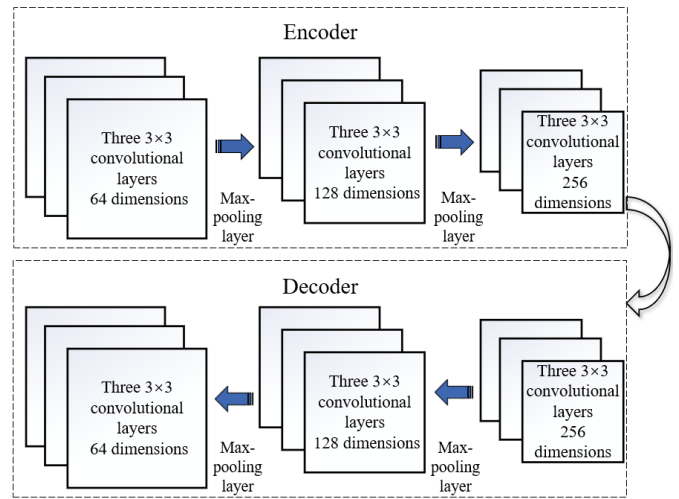


Figure 2. The architecture of the DAE-CNN

The inputs of the BDNN are the influencing factors of the measuring errors of electricity meters. In this research, the entire dataset of the influencing factors contains 17,710 samples. First, only the harmonic features (1,610 samples) were inputted to the subnetwork. To train the subnetwork, the 1,610 samples were divided into a pretraining dataset and a finetuning dataset at the ratio of 3:7. As their names suggest, the pretraining dataset was used to pretrain the subnetwork and the finetuning dataset to finetune to subnetwork. Then, the remaining samples in the entire dataset were adopted to train the entire BDNN.

3. PREDICTION EXPERIMENTS

To verify its effectiveness, the proposed BDNN was compared with other error prediction methods through four experiments on actual datasets of different sizes. All the experiments were carried out on Google's TensorFlow framework, using Python codes.

The prediction result of each method was evaluated by a test set, which is a part of the sample data. The sample data were divided into three parts, a training set, a verification set and a test set. Each method was trained the training set, and the training results were verified using the verification set. Through the training, the parameters of each method were improved continuously. Finally, the performance of each method was verified using the test set.

The prediction result of each method was evaluated by three metrics: the mean absolute error (MAE), the mean squared error (MSE) and the root mean square error (RMSE):

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - y'_i| \quad (1)$$

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - y'_i)^2 \quad (2)$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - y'_i)^2} \quad (3)$$

where, n is the sample size; y_i is the real value; y'_i is the predicted value.

The MAE reflects the deviation between the real value and the predicted value. This metric is very robust for large errors. Meanwhile, the MSE and the RMSE effectively reflect the dispersion of each method. However, these two metrics might magnify the large errors. Together, the three metrics demonstrate the performance of each prediction method from multiple angles.

3.1 Hyperparameter optimization

For a neural network model, there is a set of hyperparameters to be adjusted, in addition to network weights and bias parameters. These hyperparameters are related to activation functions, number of hidden layers and autoencoder structure, exerting a great impact on the training results. Therefore, the hyperparameters must be optimized before evaluating any neural network. Hence, three preliminary experiments were conducted to assess how three hyperparameters, including the activation function, the number of hidden layers, and autoencoder structure, affect the error prediction ability of a neural network. During the experiments, the learning rate was set to 0.001 and the batch size to 100.

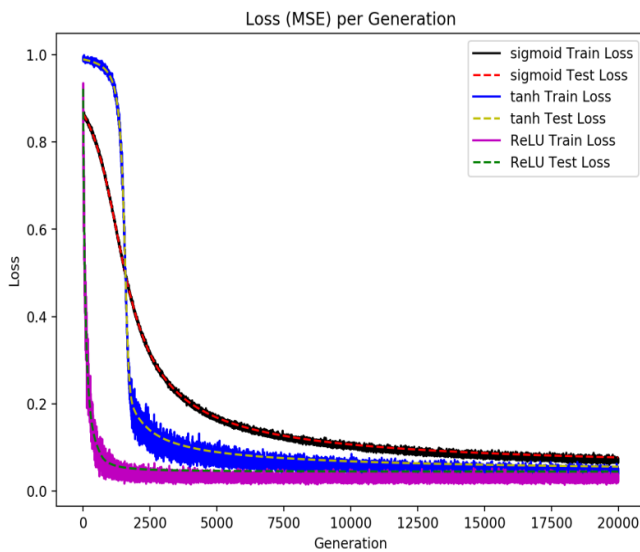


Figure 3. Prediction errors of neural networks trained by different activation functions

Table 1. Performance of neural network trained by different activation functions (mean of the last 100 iterations)

| Activation function | MSE | MAE | RMSE |
|---------------------|------------|-----------|-----------|
| ReLU | 0.00069591 | 0.0204587 | 0.0263801 |
| Sigmoid | 0.00502961 | 0.0671789 | 0.0709197 |
| Tanh | 0.00248868 | 0.0327884 | 0.0498867 |

Firstly, the neural network was trained with three different activation functions in turn: rectified linear unit (ReLU), sigmoid and tanh. The training results after 20,000 iterations are displayed in Figure 3 and Table 1. It can be seen that the neural network trained with sigmoid had the poorest results in terms of MSE, MAE and RMSE, while that trained with ReLU achieved the best results. In addition, the neural network trained with ReLU saw the fastest decline in training and test errors, i.e. the smallest overall error. Hence, ReLU was

adopted as the activation function for the verification experiments. This activation function also boasts fast convergence and accurate error feedbacks in multilayer perceptron (MLP) training.

Next, the number of hidden layers in the neural network was adjusted from 1 to 5 before prediction. The experimental results are recorded in Figure 4 and Table 2. Judging by MSE, MAE and RMSE, the different neural networks could be ranked in descending order of prediction performance as: the neural network with 4 hidden layers, that with 5 hidden layers, that with 2 hidden layers, that with 3 hidden layers and that with 1 hidden layer. The results can be explained as follows: If there are too few layers, the neural network will have insufficient ability to realize nonlinear fitting; if there are too many layers, the neural network will easily face overfitting. Therefore, the number of hidden layers was set to 4 in the verification experiments.

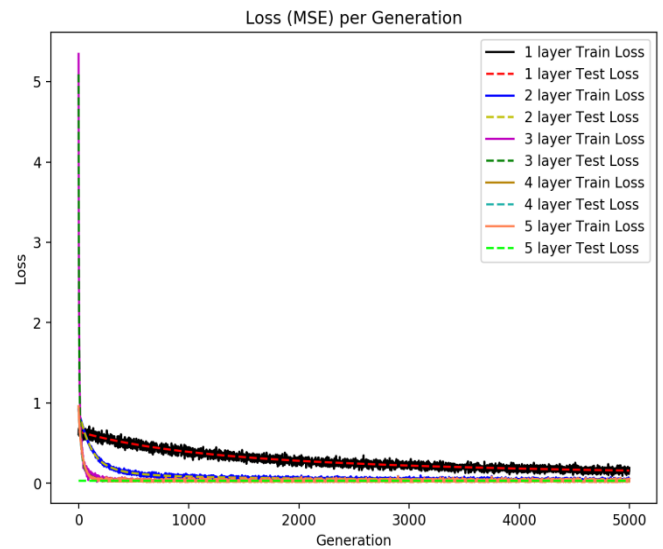


Figure 4. Prediction errors of neural networks with different number of hidden layers

Table 2. Performance of neural network with different number of hidden layers (mean of the last 100 iterations)

| Number of layers | MSE | MAE | RMSE |
|------------------|-------------|-----------|-----------|
| 1 | 0.00649618 | 0.0625282 | 0.0805988 |
| 2 | 0.00126128 | 0.0249926 | 0.0355145 |
| 3 | 0.00348837 | 0.0388211 | 0.0590624 |
| 4 | 0.000428885 | 0.0160663 | 0.0207095 |
| 5 | 0.000600928 | 0.0172511 | 0.0245138 |

Finally, the neural network was coupled with four different autoencoder structures before prediction: 100-50-25-12, 128-64-32-16, 200-100-50-25 and 256-128-64-32. Each numerical figure refers to the number of nodes in the corresponding hidden layer. The experimental results in Figure 5 and Table 3 show that the performance of the neural network increased with the number of hidden layer nodes. In terms of the MAE, the neural network with the autoencoder structure of 200-100-50-25 had a slight edge over the other neural networks. Overall, however, the neural network with a stacked autoencoder (256-128-64-32) provided the best outcome. Hence, the 256-128-64-32 was adopted as the autoencoder structure for the verification experiments.

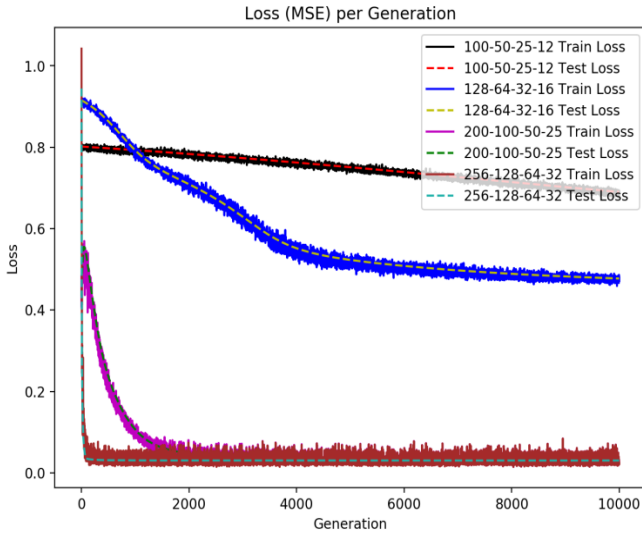


Figure 5. Prediction errors of neural networks with different autoencoder structures

Table 3. Performance of neural network with different autoencoder structures (mean of the last 100 iterations)

| Architecture | MSE | MAE | RMAE |
|---------------|-------------|----------|-----------|
| 100-50-25-12 | 0.459762 | 0.526034 | 0.678058 |
| 128-64-32-16 | 0.223107 | 0.332401 | 0.472342 |
| 200-100-50-25 | 0.000533585 | 0.015187 | 0.0230995 |
| 256-128-64-32 | 0.00051362 | 0.017557 | 0.0226631 |

3.2 Comparison between BDNN and other methods

Based on the optimized hyperparameters, the proposed BDNN was compared with the least squares support vector machine (LS-SVM), the fully-connected MLP (FCP) and the original CNN. All the four methods were trained with the same dataset. The prediction errors and error rates of the four methods are displayed in Figure 10 and Table 5, respectively. The comparison in Table 5 shows that the BDNN achieved the minimum prediction error by any metric, with a significant lead over the LS-SVM and the FCP. It can also be seen from Figure 6 that the predicted values of the BDNN were closely correlated with the actual errors. This result confirms that the BDNN can predict errors based on environmental conditions, and offer the best solution to error estimation despite the variation in environmental conditions.

The computing time in the training phase is another important indicator of error prediction effect. The total computing time of the BDNN was much shorter than that of any other method, especially compared with the LS-SVM. Hence, the BDNN outperforms the contrastive methods in the forecast of metering errors.

Of course, any neural network approach has a high initial overhead, because of the required amount of training. Despite the initial overhead, the experimental results suggest that the BDNN can accurately predict the errors of electricity meters under multi-dimensional conditions. Then, a key issue is to provide enough data for initial training. Once trained, the machine learning will save lots of time through the iterations.

In calibration, the standard measured value should be two orders of magnitude higher than the measured value, i.e. the prediction error in our experiments should be controlled within 1%. The BDNN obviously satisfies this requirement, with a

prediction error of only 0.002%. Therefore, our approach can fully support onsite calibration of electricity meters.

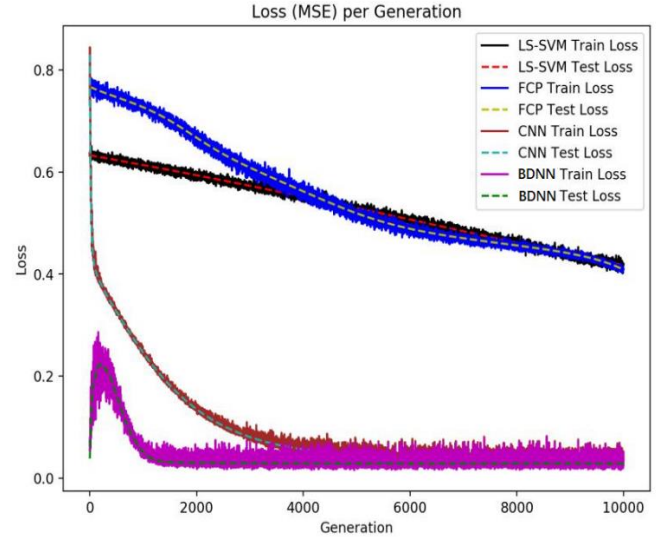


Figure 6. Prediction errors of LS-SVM, FCP, CNN and BDNN

Table 4. Performance of LS-SVM, FCP, CNN and BDNN

| Architecture | MSE | MAE | RMAE |
|--------------|------------|-----------|-----------|
| LS-SVM | 0.175321 | 0.324835 | 0.418713 |
| FCP | 0.167789 | 0.288262 | 0.409621 |
| CNN | 0.00262098 | 0.0360277 | 0.0511955 |
| BDNN | 0.00235818 | 0.0101225 | 0.0485611 |

4. CONCLUSIONS

From the perspective of deep learning, this paper aims to effectively predict the measuring errors of electricity meters under actual conditions. For this purpose, it is necessary to optimize the inputs and hyperparameters for the task, and compare the established approach with other methods in terms of prediction error.

To achieve the above goals, a BDNN was designed and tested with an DAE-CNN subnetwork, and a fully-connected network. The subnetwork was specially developed to process the harmonic features and provide an input vector to the fully-connected network. The proposed model was trained on a dataset generated by real experiments with electricity meters. Then, three hyperparameters were optimized through another set of experiments: ReLu was selected as the activation function, the number of hidden layers was set to 4, and the autoencoder structure was determined as 256-128-64-32. Finally, our model was compared with several deep learning methods through experiment. The results show that the BDNN outperformed the other methods in prediction error and computing cost. This means our model can effectively combine unsupervised pretraining with a stacked denoising autoencoder (SDA) with a supervised finetuning strategy, when there are enough data.

In addition, as mentioned in the Introduction, the current error prediction methods for electricity meters involve onsite calibration. However, these approaches cannot perform consistently in the field, because the calibration techniques are all designed in lab conditions. This defect is overcome in our research. The proposed BDNN can correctly verify the results

collected onsite, and effectively evaluate the errors of electricity meters.

REFERENCES

- [1] Muscas, C., Pau, M., Pegoraro, P.A., Sulis, S. (2018). Smart electric energy measurements in power distribution grids. *IEEE Instrumentation & Measurement Magazine*, 18(1): 17-21. <https://doi.org/10.1109/MIM.2015.7016676>.
- [2] Brandolini, A., Faifer, M., Ottoboni, R. (2009). A simple method for the calibration of traditional and electronic measurement current and voltage transformers. *IEEE Transactions on Instrumentation and Measurement*, 58(5): 1345-1353. <https://doi.org/10.1109/TIM.2008.2009184>
- [3] Femine, A.D., Gallo, D., Landi, C., Luiso, M. (2009). Advanced instrument for field calibration of electrical energy meters. *IEEE Transactions on Instrumentation and Measurement*, 58(3): 618-625. <https://doi.org/10.1109/TIM.2008.2005079>
- [4] Mazza, P., Bosonetto, D., Cherbaucich, C., De Donà, G., Franchi, M., Gamba, G., Gentili, M., Milanello, C.D. (2014). On-site verification and improvement of the accuracy of voltage, current and energy measurements with live-line working methods. New equipment, laboratory and field experience, perspectives 2014 11th International Conference on Live Maintenance (ICOLIM), pp. 1-8. <https://doi.org/10.1109/ICOLIM.2014.6934377>
- [5] Cai, B.S., Gao, M.T., Zhou, W.W., He, Z.W. (2013). A design of error computing system for multifunctional watt-hour meter. 2013 Third International Conference on Instrumentation, Measurement, Computer, Communication and Control, pp. 715-718. <https://doi.org/10.1109/IMCCC.2013.158>
- [6] Granderson, J., Touzani, S., Custodio, C., Sohn, M. D., Jump, D., Fernandes, S. (2016). Accuracy of automated measurement and verification (M&V) techniques for energy savings in commercial buildings. *Applied Energy*, 173: 296-308. <https://doi.org/10.1016/j.apenergy.2016.04.049>
- [7] Otdomański, P., Zazula, P. (2013). The experimental verification of metrological properties of direct current Watt-hour meter. *Journal of Physics: Conference Series*, 459(1): 012058. <https://doi.org/10.1088/1742-6596/459/1/012058>
- [8] Femine, A.D., Member, S., Gallo, D., Landi, C., Luiso, M. (2007). Measurement equipment for on-site calibration of energy meters. 2007 IEEE Instrumentation & Measurement Technology Conference IMTC 2007, pp. 1-6. <https://doi.org/10.1109/IMTC.2007.379144>
- [9] Avram, S., Plotenco, V., Paven, L.N. (2017). Design and development of an electricity meter test equipment. 2017 International Conference on Optimization of Electrical and Electronic Equipment (OPTIM) & 2017 Intl Aegean Conference on Electrical Machines and Power Electronics (ACEMP), pp. 96-101. <https://doi.org/10.1109/OPTIM.2017.7974954>
- [10] Elena, K., Sukalo, A. (2016). Energy measurements verification based on evaluation residues. 2016 International Symposium on Industrial Electronics (INDEL), pp. 1-5. <https://doi.org/10.1109/INDEL.2016.7797779>
- [11] Zhou, N.C., Wang, J., Wang, Q.G. (2017). A novel estimation method of metering errors of electric energy based on membership cloud and dynamic time warping. *IEEE Transactions on Smart Grid*, 8(3): 1318-1329. <https://doi.org/10.1109/TSG.2016.2619375>
- [12] Su, H., Qi, C.R., Li, Y.Y., Guibas, L.J. (2015). Render for CNN: Viewpoint estimation in images using CNNs trained with rendered 3d model views. 2015 IEEE International Conference on Computer Vision (ICCV), pp. 2686-2694. <https://doi.org/10.1109/ICCV.2015.308>
- [13] Wang, H.F., Hu, D.J. (2005). Comparison of SVM and LS-SVM for regression. *Proceedings of 2005 International Conference on Neural Networks and Brain Proceedings*, 1: 279-283. <https://doi.org/10.1109/icnnb.2005.1614615>
- [14] Berriel, R.F., Lopes, A.T., Rodrigues, A., Varejão, F.M., Oliveira-Santos, T. (2017). Monthly energy consumption forecast: A deep learning approach. 2017 International Joint Conference on Neural Networks (IJCNN), pp. 4283-4290. <https://doi.org/10.1109/IJCNN.2017.7966398>
- [15] Dong, B., Li, Z.X., MahboburRahman, S.M., Vega, R. (2016). A hybrid model approach for forecasting future residential electricity consumption. *Energy and Buildings*, 117: 341-351. <https://doi.org/10.1016/j.enbuild.2015.09.033>
- [16] LeCun, Y., Bengio, Y., Hinton, G. (2015). Deep learning. *Nature*, 521(7553): 436-444. <https://doi.org/10.1038/nature14539>
- [17] Schmidhuber, J. (2015). Deep Learning in neural networks: An overview. *Neural Networks*, 61: 85-117. <https://doi.org/10.1016/j.neunet.2014.09.003>
- [18] Liu, W.B., Wang, Z.D., Liu, X.H., Zeng, N.Y., Liu, Y.R., Alsaadi, F.E. (2017) A survey of deep neural network architectures and their applications. *Neurocomputing*, 234: 11-26. <https://doi.org/10.1016/j.neucom.2016.12.038>
- [19] Szegedy, C., Liu, W., Jia, Y.Q., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A. (2015). Going deeper with convolutions. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1-9. <https://doi.org/10.1109/CVPR.2015.7298594>
- [20] Wu, Q.X., Gui, Z.J., Li, S.Q., Ou, J. (2017). Directly connected convolutional neural networks. *International Journal of Pattern Recognition and Artificial Intelligence*, 32(5): 1859007. <https://doi.org/10.1142/S0218001418590073>
- [21] Li, C.K., Zhao, D.F., Mu, S.J., Zhang, W.H., Shi, N., Li, L.N. (2019). Fault diagnosis for distillation process based on CNN-DAE. *Chinese Journal of Chemical Engineering*, 27(3): 598-604. <https://doi.org/10.1016/j.cjche.2018.12.021>
- [22] Vincent, P., Larochelle, H., Bengio, Y., Manzagol, P.A. (2008). Extracting and composing robust features with denoising autoencoders. *Proceedings of the 25th international conference on Machine learning*, 1096-1103. <http://dx.doi.org/10.1145/1390156.1390294>
- [23] Hinton, G.E., Osindero, S., Teh, Y.W. (2006). A fast learning algorithm for deep belief nets. *Neural computation*, 18(7): 1527-1554. <http://dx.doi.org/10.1162/neco.2006.18.7.1527>
- [24] Hinton, G.E., Salakhutdinov, R.R. (2006). Reducing the

dimensionality of data with neural networks. *Science*,
313(5786): 504-507.
<http://dx.doi.org/10.1126/science.1127647>
[25] Xue, B., Zhang, M.J., Browne, W.N., Yao, X. (2015). A

survey on evolutionary computation approaches to
feature selection. *IEEE Transactions on Evolutionary
Computation*, 20(4): 606-626.
<http://dx.doi.org/10.1109/TEVC.2015.2504420>