2019 International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)

L-RBF : A Customer Churn Prediction Model Based on Lasso+RBF

Anping Xiong, Ya You, Linbo Long[1]

Abstract

With the development of market economic, customer churn prediction play a critical role in the company management. However, customer information is complex, it contains multi-dimensional features. What's worse, the number of customer churn very small among the whole consumers, and the features of customer are dynamically changing, which is challenging for traditional statistical methods. Fortunately, as the rise of edge computing, more computing related to data-intensive applications will be decentralized to edge smart terminals. Moreover, edge computing focuses on real-time, short-cycle analysis, which is useful for dealing with dynamic changes problems in customer features. Therefore, to address these issues about customer messages, this paper proposes a simple and effective model to predictive customer churn with a higher accuracy, named *L-RBF*. The basic idea is to utilize Lasso Regression algorithm (Lasso) for optimizing Radial Basis Function Neural Network (RBF). At first, placing customer features on edge terminals, and Lasso is utilized to extract the correlated features of customer churn. Then, according to the correlated features and lasso regression equation, we can automatically set the topology and parameter information of RBF. Finally, experimental results indicate that *L-RBF* has a higher recall rate and stronger prediction classification ability compared with the previous work, included Logistic Regression (Log-R), RBF and Boosting.

Keywords: Lasso, RBF, Churn prediction, Weight initialization, Centers and Spread constant

1. Introduction

As market competition intensifies, developing new customers become tough for service industry. Thus, companies pay more attention to the current customers so as to prevent customer churn[1, 2, 3]. So it is important to predict the customer who might churn and propose marketing plans for these ones in order to retain them. However, the customer features is gradually increasing. There is no doubt that redundant features not only increase the scale of computation but also impact the accuracy of model. In addition, the number of the churned customers is smaller in the whole data set, which is challenging to predict customers churn[4, 5]. Meanwhile, the required parameters are calculated at the same terminal, which will reduce the efficiency and performance of the algorithm.Therefore, this paper will propose a novel and simple model to accurately predict the churn risk of customers.

In prediction management of customer churn, the problem of customer churn can be regarded as a binary classification problem. In recent years, many researcher have already focused on addressing customer churn prediction. At first, Bi et al.[2] proposed a semantic-driven subtractive clustering method (SD-SCM), which is combined with axiomatic fuzzy set (AFS) and subtractive clustering method (SCM) to calculate the K value of K-means. Then, Lu et al.[1] separated customers into two clusters according to the weight assigned by the boosting algorithm. Meanwhile, this work used Logistic Regression as a basis learner to train a churn prediction model on each cluster, respectively. After that, a meta heuristic based churn prediction technique was proposed by Ammar et al.[6], which used a hybridized form of firefly algorithm as the classifier. In addition, Castro et al.[7] propose a frequency analysis approach based on k-nearest neighbors' machine learning algorithm for feature representation from login records for churn prediction modeling. Their empirical evaluation results indicate that data mining techniques can effectively assist company to achieve the customers who might be lost.

Although these methods can deal with the problem of customer churn efficiently, we find that some issues might be incurred. On the one hand, since customer information is complex, and there is an imbalance class problem in customer data. When dealing with issue which is complex, the reliability and effectiveness of traditional statistical methods will be declined. On the other hand, the method selection is inappropriate, the prediction results will be not accurate enough. What's more, these methods are all transfer the total data to a single terminal for calculation, which will increase the storage bottleneck, execution efficiency and bandwidth resources of the terminal. Therefore, these observations motivate us to propose a new and effective predict-method, to obtain the accurate results.

More specifically, to address these issues, this paper proposes a new and simple prediction method, named *L-RBF*. The basic idea is to utilize Lasso Regression algorithm (Lasso) [8] for optimizing Radial Basis Function Neural Network (RBF) [9]. At first, placing customer features on edge terminals [10, 11], we use Lasso to extract the correlated features of customer churn. Then, we can obtain core points and neighborhood radius based on the idea of Density-Based Spatial Clustering of

[1]Corresponding author: Linbo Long, College of Computer Science and Technology, Chongqing University of Posts and Telecommunications, Chongqing, China. E-mail: longlb@cqupt.edu.cn

978-1-7281-2980-8/19/$31.00 ©2019 IEEE

DOI 10.1109/iThings/GreenCom/CPSCom/SmartData.2019.00121

Applications with Noise (DBSCAN) [12]. After that, combining with on the idea of Logistic Regression (Log-R), we can compute the init weight according to core points and lasso regression equation. Finally, we setup the RBF according to those parameters.

The evaluation results indicate that the proposed model *L-RBF* can effectively improve the precision and recall of classification results compared with previous work.

The contributions of this paper are summarized as follows.

- Combining with the complex customer information and the problem of setting parameters, we propose a simple and effective prediction model, called *L-RBF*, to predict the customer churn for improving the accuracy of classification and maintaining existing customers. Compared to previous work, *L-RBF* has a higher prediction precision and lower error influence.
- We can easily update the parameters required by the model through edge calculations, it is can improve the predictive validity of the model.
- Finally, a series of experiments based on a bank data set and a telecom data set have been conducted to evaluate the effectiveness of the proposed techniques in this paper.

The rest of the paper is organized as follows. Section 2 introduces the background and motivation. In Section 3,we proposed a novel and simple model, named *L-RBF*. Then, through experimental verification of a subset from a bank and a telecommunications, compared and analyzed with Logistic Regression (Log-R), Radial Basis Function Neural Network (RBF), Boosting Algorithm (Boosting) in Section 4. In the last section, we conclude this paper.

2. Background and Motivation

Nowadays, as the rise of edge computing, more computing related to data-intensive applications will be decentralized to edge smart terminals, which not only accelerates the training speed of the neural network, but also improves the utilization of the terminal resources. Thus, under the situation of complex customer information and irregular churn reason, this paper combined edge calculation, selects ANN [13] as a basis classification algorithm. The detail elaboration is brought in the next section.

2.1. Edge Computing

There are more and more methods on customer churn, but these methods always transfer the total data to a single terminal for calculation, which will increase the storage load and reduce the execution efficiency of a single terminal. Fortunately, with the size of data and the diversification of data processing continued to increase, edge computing was born. Edge computing refers to an open platform that integrates network, computing, storage, and application core capabilities on the side close to the source, providing near-end services [11]. In edge calculation, the terminal node performs a certain amount of calculation and data processing, and then passes the processed data to another terminal for intelligent calculation. This real-time and multi-terminal processing model not only improves the efficiency of data transfer and execution, but also reduces the scale of computation and storage costs [14]. Therefore, when processing customer information data, combined with edge calculation can improve the processing efficiency of the prediction method and save a lot of human and material costs.

2.2. Radial Basis Function Neural Network

RBF in ANN has simple structure, simple training and wide application. It is especially prominent in solving classification problems [15]. RBF is consisted of three layers of neurons [15, 16, 17, 18], it includes input layer, hidden layer and output layer.

Input layer. The first layer is input layer, it is made up of sample feature variables. But redundant features can reduce the accuracy of prediction.

Hidden layer. The second layer is hidden layer, it transforms the low-dimensional input data into high-dimensional space. The basis function of neurons usually uses gaussian kernel function.

$$G(\|X - C\|) = exp(-\frac{\|X - C\|^2}{2\sigma^2}) \tag{1}$$

where C is the vector describing the location of the center of this RBF unit and σ is the spread constant (i.e, named radius) of this RBF unit. They affect the accuracy of the model

Output layer. The third layer is output layer. It is made up of sample labels. The output is calculated by using weighted summation of out-up. Where the weight is the tunable parameter of RBF, which reflects the contribution of hidden layer neurons to output results.

Next, in the following part, we will state the motivation of the proposed algorithm in this paper.

2.3. Motivation

With the rapid development of the data, the customer information become complex. It will be very challenging for discovering the reasons of customer churn. This paper focuses on RBF to predict the customers who might churn. Generally, the performance of RBF depends on the parameter setting and topology, and the calculation of parameters in a single terminal will affect the efficiency of the algorithm. Therefore, this paper proposes a new algorithm called *L-RBF* combined with edge computing. The algorithm puts the calculation about extracting features and obtaining RBF parameters into the edge terminal, so that the execution efficiency of the algorithm can be improved, and the thought map is shown in Figure 1.

In the next section, the detail methods can be presented.

3. Methodology

In this Section, we will introduce the main work flow of the proposed *L-RBF*.

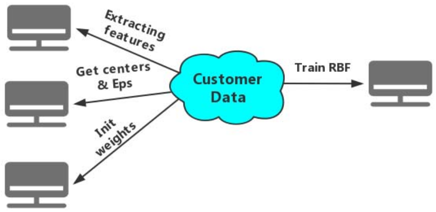


Figure 1: The thought of Edge calculation

3.1. Overview

According to the performance of RBF not only depend on a proper topology, but also lie on appropriate parameters. Therefore, this paper proposes a new method to setup RBF. The main work of the proposed method can be divided into computing the reasonable parameters of RBF and simplifying the topology of RBF. The process diagram about the two parts is shown in Figure 2.

Computing the reasonable parameters of RBF. We will calculate the data at the edge terminal. As shown in Figure 2(a), we first use Lasso [19] to extract the features of customer churn. Meanwhile, we can get the *lasso-func* in this process.

$$y^{lasso} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_k x_k \tag{2}$$

where β_1,β_2,...β_k is the non-zero regression coefficient, x_i represents the extracted *ith* feature of the sample. Then, combining with the idea of DBSCAN, we can obtain core points and *Eps*. After that, with the idea of Log-R, we put the center point into the *lasso-func*, and bring the value into the sigmoid function(3).

$$\frac{1}{1+e^{-x}} \tag{3}$$

The final result is calculated by formula (3) can be represented by p.

Simplifying the topology of RBF. Transfer the value calculated by the edge terminal and the processed data to another terminal, and prepare for intelligent calculation. As shown in Figure 2(b), we present the detail steps of simplifying the topology of RBF. Firstly, the features are extracted by Lasso can be used as input neurons. Secondly, we optimize the traditional self-organizing, and use the core points as the center points. What's more, we build a linear map between the *Eps* and the spread constant σ.Therefore, the formula can be defined as follows.

$$\sigma = \lambda * Eps \tag{4}$$

λ is a mapping constant and λ> 0. Finally, this paper uses the p and 1-p as the initial weight of the neuron with the output of 1 and the neuron with the output of 0.

According to the above description, for solving the issues about parameters and topology, we take advantage of Lasso and RBF and propose a novel and simple algorithm, named *L-RBF*.

Table 1: The description of the notation used by *L-RBF*

Symbol	Description
n	The number of sample points
Xi	The *ith* sample point
D	Raw data set
initW1	Initial weights for the *y1* neurons
initW0	Initial weights for the *y0* neurons
ci	The *ith* center point
ck	Number of center points
tn	Train sample size
η	Learning rate
epoch	The number of training branches
teamSize	The number of sample points per batch
Xtest	Test set
Xtrain	Train set
ytrain	The corresponding label of *Xtrain*

3.2. Algorithm of L-RBF

In this part, we will explain the details of *L-RBF* algorithm.

Step1 Extract Features with lasso. Training the Lasso with *D*, the *lasso-func* and the features with the non-zero coefficients are outputted.

Step2 Update D. Update *D* to a new data set D_{new} according to the feature extracted by Step1.

Step3 Get *Eps* and *MinPts*. According to equation (5) to calculate the Euclidean distance between sample.

$$d(i,j) = \sqrt{(x_1-y_1)^2 + (x_2-y_2)^2 + ... + (x_m-y_m)^2} \tag{5}$$

The ascending sort matches the obtained distances, and fit a curve with this sequence later. The first sharp changing position from the curve is marked *Eps*. Meanwhile, we take the corresponding index of *Eps* as *MinPts*.

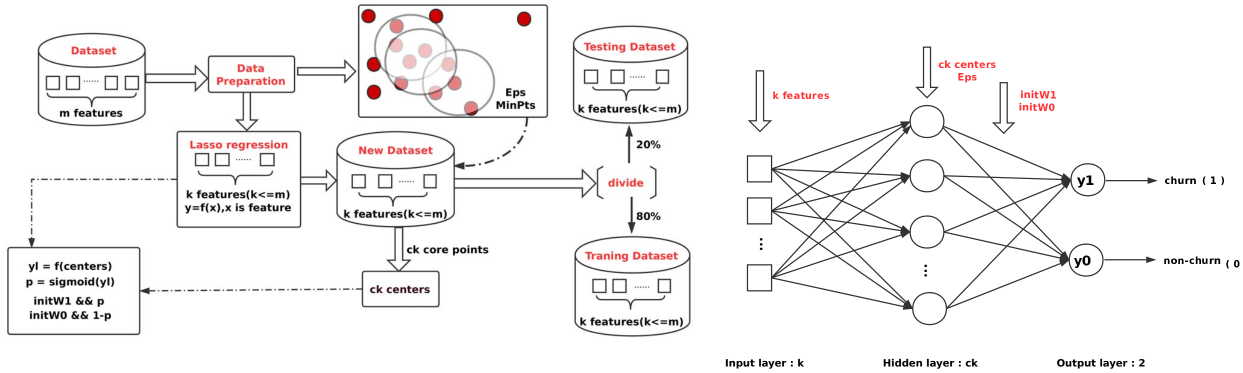
Step4 Get *Centers*. From the *Eps* and *MinPts* obtained by Step 3, the core point can selected.

Step5 Initialize Weight. Combining *lasso-func* with *centers* with the idea of Log-R to compute a result(p).

Step6 Train RBF. Setup RBF and train it following Step 6(A) to Step 6(F).

A. Setup parameters and topology.

- Set features which extracted by Lasso to input neurons.
- Calculate λ * *Eps* as the spread constant of the basis function.
- Set centers to the center of the basis function.
- Set two neurons to output neurons named *y1* and *y0*, representing churn *(1)* and non-churn *(0)*, respectively.
- Set *initW1* and *initW0* to initial weights for *y1* and *y0*, respectively.



(a) Setup the parameters of RBF (b) Setup the topology of RBF

Figure 2: The establishment of RBF in the entire churn prediction method

B. Setup input vectors, output vectors and prediction results.

- Set input vectors.

$$X = [x_1, x_2, ..., x_k]^T \tag{6}$$

where m is the number of input layer units.

- Set output vectors.

$$ypre = [ypre1_j, ypre0_j] \tag{7}$$

- Set expected output vector.

$$t = \begin{cases} 1, & [1, 0] \\ 0, & [0, 1] \end{cases} \tag{8}$$

- Set prediction result.

$$y = \begin{cases} 1, & ypre1_j > ypre0_j \\ 0, & ypre1_j < ypre0_j \end{cases} \tag{9}$$

Since there are two output neurons $y1$,$y0$, the output vector is expressed as $[ypre1_j, ypre0_j]$. Meanwhile, the t should also be a 2-dimensional vector. If the expectation is 1, it is represented as t=[1, 0], and if it is expected to be a 0, it is represented as t=[0, 1]. In addition, $ypre1_j$and $ypre0_j$ represents the output of $y1$, $y0$, separately. If $ypre1_j$>$ypre0_j$, the predict result of network is churn; if $ypre1_j$<$ypre0_j$, the result is non-churn.

C. Calculate the output of the jth *sample through the* ith *neuron of the hidden layer.*

$$ypre1_j = \sum_{i=1}^{ck} w1_i * G(\|X_j - C_i\|) \tag{10}$$

$$ypre0_j = \sum_{i=1}^{ck} w0_i * G(\|X_j - C_i\|) \tag{11}$$

D. Calculate the total output error of the output layer neurons named totalErr.

$$totalErr = \frac{1}{2} * \sum_{j=1}^{tn} [(t[0] - ypre1_j)^2 + (t[1] - ypre0_j)^2] \tag{12}$$

Use *totalErr* as the objective function.

E. Calculate weight parameters iteratively.

- Initialize network weights.

$$W1_i = initW1_i \tag{13}$$

$$W0_i = initW0_i \tag{14}$$

- Calculate the network weight change according to equations (13) and (14).

$$\begin{aligned} \Delta W1_i &= \eta * \frac{\partial totalErr}{\partial w1_i} \\ &= \eta * \frac{\partial totalErr}{\partial y1} * \frac{\partial y1}{\partial w1_i} \\ &= \eta * (ypre1_j - t[0]) * G(\|X_j - C_i\|) \end{aligned} \tag{15}$$

$$\begin{aligned} \Delta W0_i &= \eta * \frac{\partial totalErr}{\partial w0_i} \\ &= \eta * \frac{\partial totalErr}{\partial y0} * \frac{\partial y0}{\partial w0_i} \\ &= \eta * (ypre0_j - t[1]) * G(\|X_j - C_i\|) \end{aligned} \tag{16}$$

Finally,update weight.

$$W1_i = W1_i - \Delta\, W1_i \tag{17}$$

$$W0_i = W0_i - \Delta\, W0_i \tag{18}$$

Algorithm 1 Train RBF

Input: D_{new}, *epoch*, *teamSize*, *centers*,η, *Xtrain*, *ytrain*,ε, $W1_i$,$W0_i$

Output: The RBF

Calculate the number of training weight sample groups:

$$team = \lceil \frac{n}{teamSize} \rceil \tag{21}$$

for i=1,2,...,*epoch* **do**
    Randomly sort D_{new};
    for j=1,2,...,*team* **do**
        Calculate the total weight change for the *jth* batch according to formula (15),(16)
    Update weights according to formula (17) , (18)
    Calculate *totalErr* according to formula (12)
    if $totalErr \leq \varepsilon$ **then**
        break
return The RBF

F. Train neural network.

- Using batch gradient descent to train the neural network

Initialize the neural network according to Step6(B), and set the learning rate η and the iteration termination condition ε. As Algorithm 4 shows.

4. Experiment

This section will evaluate the classification effect of the *L-RBF*. Firstly, we introduce the classic evaluation indicators in Section 4.1. Secondly, we conduct experiments in a bank customer data set and a telecom company customer data set in Section 4.2. Finally, RBF, Log-R, and Boosting are compared and analyzed.

4.1. Measurement standard

In general, the evaluation indicators for the two classifications are usually used classical confusion matrices [20]. The model evaluates the algorithm depends on the following four indicators (Using 1 to represent churn).

4.1.1. Evaluation indicators

- *Accuracy rate*, which is the accuracy of the overall judgmentz
- *Precision rate*, which is the accuracy of the prediction 1 in the prediction result.
- *Recall rate*, which is the accuracy of the prediction 1 in the actual result.
- *F1-score*, which is a comprehensive measurement of *Precision* and *Recall*, and can be used to measure the overall accuracy of a binary model.

In addition, for the two-class model, there are ROC and AUC [21] as the evaluation of the classifier. A simple introduction to ROC and AUC is as follows.

4.1.2. ROC and AUC

ROC is a curve with a true positive rate as the ordinate and a false positive rate as the abscissa. AUC is defined as the area under the ROC curve, which is used to measure the accuracy of the model, and the classifier with a larger AUC value works better.

4.2. Experimentation

4.2.1. Schemes for comparison

RBF. The parameter information of RBF algorithm will be set according to data set.

Log-R. The Logistic Regression algorithm.

Boosting. The Boosting algorithm was proposed in 2014 by Lu N [1].

The following experiments were performed in python 3.6 and PCharm 2017.2.3. Meanwhile, using the same data set with 8:2 data volume as the train set and test set.

4.3. Experimental and analysis of results

4.3.1. A bank customer data set

The sample data comes from the anonymized real data set, downloaded from the superdatascience official website, which contains 10,000 data, 10 attributes. According to this data set, the experimental results are as follows.

The verification results of these four algorithms are shown in Table 4.

Table 2: Evaluation index of a bank customer data set

Algorithm	Accuracy	Precision	Recall	F1-score
L-RBF	80.95%	53.02%	52.10 %	52.55%
RBF	77.75%	43.55%	33.33 %	37.77%
Log-R	80.7%	57.6%	17.78 %	27.17%
Boosting	81.05%	60.32%	18.77 %	28.63%

It can be seen from Table 4, Boosting's *accuracy* and *precision* are slightly higher, but for the *recall* and *f1-score* values, the *L-RBF* is much higher than the other three algorithms, indicating that the *L-RBF* classifier is more sensitive to the real category of 1. Moreover, the *f1-score* value further shows that the overall classification of the *L-RBF* is the best. Next, this paper will supplement the analysis from AUC.

The AUC values are expressed in Table 5.

Table 3: The AUC values of these four algorithms

	L-RBF	RBF	Log-R	Boosting
AUC	0.7	0.61	0.57	0.58

It is clear from Table 5 that the AUC value of *L-RBF* is largest, the RBF is second, then Boosting, and finally is Log-R. Therefore, recognized from the AUC that *L-RBF* has the best classification effect. It is proved that the proposed method is effective. Next, we will test the effectiveness of our method through a telecom customer data set again.

4.3.2. A telecom company customer data set

A portion of the customer data set from a telecommunications company, contains 3,333 data and 19 attributes. In the *L-RBF*,13 neurons are inputted, and the *L-RBF* and RBF have 100 *centers*. The four algorithms used are identical to those of a bank customer data set. The experimental results are shown in Table 6, and the results are similar to a bank customer data set.

Table 4: Evaluation index of a bank customer data set

Algorithm	Accuracy	precision	Recall	F1-score
L-RBF	91.45%	76.47%	55.91 %	64.60%
RBF	86.96%	55%	35.48 %	43.14%
Log-R	87.11%	60%	22.59 %	32.81%
Boosting	87.56%	63.16%	25.81 %	36.64%

The AUC values are shown in Table 7:

Table 5: The AUC values of these four algorithms

	L-RBF	RBF	Log-R	Boosting
AUC	0.77	0.65	0.60	0.62

As shown in Table 6 and Table 7, the four indicators of *L-RBF* all are the largest. It is indicate that the *L-RBF*'s classification has a better classification prediction effect.

5. Conclusion

The purpose of this paper is to provide companies with a better method to predict customer churn, in order to provide operators with more effective decision-making solutions and greater economic benefits. We integrate Lasso and RBF to formed a new algorithm, named *L-RBF*. By edge calculating, the topology and parameter information of the RBF can be automatically set on the original data. At the same time, the edge calculation improves the execution efficiency and data transmission rate of the algorithm. The experimental results showed that the prediction effect of *L-RBF* is higher than that of RBF, Log-R and Boosting. Therefore, *L-RBF* improves the accuracy of classification.

Acknowledgment

This work was supported in part by the Science and Technology Research Program of Chongqing Municipal Education Commission under Grant KJ1704085 and in part by the Chongqing Research Program of Basic Research and Frontier Technology under Grant cstc2017jcyjAX0164.

References

[1] N. Lu, H. Lin, J. Lu, G. Zhang, A customer churn prediction model in telecom industry using boosting, IEEE Transactions on Industrial Informatics 10 (2) (2014) 1659–1665. doi:10.1109/TII.2012.2224355.

[2] W. Bi, M. Cai, M. Liu, G. Li, A big data clustering algorithm for mitigating the risk of customer churn, IEEE Transactions on Industrial Informatics 12 (3) (2016) 1270–1281. doi:10.1109/TII.2016.2547584.

[3] Y. Huang, F. Zhu, M. Yuan, K. Deng, Y. Li, B. Ni, W. Dai, Q. Yang, J. Zeng, Telco churn prediction with big data, in: ACM SIGMOD International Conference on Management of Data, 2015, pp. 607–618. doi:10.1145/2723372.2742794.

[4] K. Kim, C.-H. Jun, J. Lee, Improved churn prediction in telecommunication industry by analyzing a large network, Expert Systems with Applications 41 (15) (2014) 6575 – 6584. doi:https://doi.org/10.1016/j.eswa.2014.05.014.

[5] B. Zhu, B. Baesens, A. Backiel, S. K. L. M. V. Broucke, Benchmarking sampling techniques for imbalance learning in churn prediction, Journal of the Operational Research Society (17) (2017) 1–17. doi:10.1057/s41274-016-0176-1.

[6] A. A. Q. Ahmed, D. Maheswari, Churn prediction on huge telecom data using hybrid firefly based classification, Egyptian Informatics Journal 18 (3). doi:10.1016/j.eij.2017.02.002.

[7] E. Castro, M. Tsuzuki, Churn prediction in online games using players' login records: a frequency analysis approach, IEEE Transactions on Computational Intelligence & Ai in Games 7 (3) (2017) 255–265. doi:10.1109/tciaig.2015.2401979.

[8] C. M. O'Brien, Statistical learning with sparsity: The lasso and generalizations, International Statistical Review 84 (1) (2016) 156–157. doi:10.1111/insr.12167.

[9] B. M. Wilamowski, C. Cecati, J. Kolbusz, P. Rozycki, P. Siano, A novel rbf training algorithm for short-term electric load forecasting and comparative studies, IEEE Transactions on Industrial Electronics 62 (10) (2015) 6519–6529. doi:10.1109/TIE.2015.2424399.

[10] M. Satyanarayanan, Edge computing, Computer 50 (10) (2017) 36–38. doi:10.1109/MC.2017.3641639.

[11] M. Satyanarayanan, The emergence of edge computing, Computer 50 (1) (2017) 30–39. doi:10.1109/MC.2017.9.

[12] Y. Lv, T. Ma, M. Tang, J. Cao, Y. Tian, A. Al-Dhelaan, M. Al-Rodhaan, An efficient and scalable density-based clustering algorithm for datasets with complex structures, Neurocomputing 171 (C) (2016) 9–22. doi:10.1016/j.neucom.2015.05.109.

[13] M. Tavana, A.-R. Abtahi, D. D. Caprio, M. Poortarigh, An artificial neural network and bayesian network model for liquidity risk assessment in banking, Neurocomputing 275 (2018) 2525 – 2554. doi:https://doi.org/10.1016/j.neucom.2017.11.034.

[14] W. Yu, F. Liang, X. He, W. G. Hatcher, C. Lu, J. Lin, X. Yang, A survey on the edge computing for the internet of things, IEEE Access 6 (99) (2018) 6900–6919. doi:10.1109/ACCESS.2017.2778504.

[15] Y. Xie, Y. Xing, A prediction method for the wax deposition rate based on a radial basis function neural network, Petroleum 3 (2) (2017) 237 – 241. doi:https://doi.org/10.1016/j.petlm.2016.08.003.

[16] G. Du, S. He, H. Yu, The pattern recognition of space debris hypervelocity impact based on rbf neural network, Spacecraft Environment Engineering.

[17] A. Addeh, A. Khormali, N. A. Golilarz, Control chart pattern recognition using rbf neural network with new training algorithm and practical features, ISA Transactions 79 (2018) 202 – 216. doi:https://doi.org/10.1016/j.isatra.2018.04.020.

[18] D. Zhang, G. Zang, J. Li, K. Ma, H. Liu, Prediction of soybean price in china using qr-rbf neural network model, Computers and Electronics in Agriculture 154 (2018) 10 – 17. doi:https://doi.org/10.1016/j.compag.2018.08.016.

[19] Y. Luo, Y. Liu, Y. Zhang, C. Yue, Speech bottleneck feature extraction method based on overlapping group lasso sparse deep neural network, Speech Communication 99 (2018) 56 – 61. doi:https://doi.org/10.1016/j.specom.2018.02.005.

[20] X. Deng, Q. Liu, Y. Deng, S. Mahadevan, An improved method to construct basic probability assignment based on the confusion matrix for classification problem, Information Sciences 340-341 (2016) 250 – 261. doi:https://doi.org/10.1016/j.ins.2016.01.033.

[21] X. Zhang, X. Li, Y. Feng, Z. Liu, The use of roc and auc in the validation of objective image fusion evaluation metrics, Signal Processing 115 (2015) 38 – 48. doi:https://doi.org/10.1016/j.sigpro.2015.03.007.