Expert Systems With Applications 199 (2022) 116982

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

A new oversampling method and improved radial basis function classifier for customer consumption behavior prediction

Yue Li [a], Xiaoyun Jia [b], Ruili Wang [b], Jianfang Qi [a], Haibin Jin [a], Xiaoquan Chu [a], Weisong Mu [a,c,*]

[a] College of Information and Electrical Engineering, China Agricultural University, Beijing 100083, China
[b] School of Natural and Computational Sciences, Massey University, Auckland 0745, New Zealand
[c] Key Laboratory of Viticulture and Enology, Ministry of Agriculture, Beijing 100083, China

ARTICLE INFO

Keywords:
Imbalanced data
Oversampling technique
RBF neural network
Immune algorithm
Consumer consumption behavior prediction

ABSTRACT

In practical applications, imbalanced data has brought great challenges to classification problems. In this paper, we propose two new methods: (1) a new oversampling method, named Tomek-CASUWO, to address the issue of class imbalance; (2) a new classifier, named ILS-RBFNN, to increase the accuracy of prediction of customer consumption behavior. Since current ASUWO algorithm is easily affected by noise samples and fuzzy class boundaries, we propose Tomek-CASUWO to address these problems: (i) the Tomek links algorithm is used to filter noise samples; (ii) CASUWO is used to avoid overlapping class boundaries; (iii) Tomek-CASUWO is used to synthesize new samples. Also, we propose a new classifier based on RBFNN, named ILS-RBFNN, to improve the prediction accuracy: (i) the hybrid kernel is developed by combining Gaussian and Polynomial; (ii) an Immune Algorithm (IA) is used to optimize the centers of RBFNN; (iii) Least-Squares (LS) is used to calculate the biases and weights. Wine-consumer behavior data is used to compare our Tomek-CASUWO with other oversampling methods. We compare ILS-RBFNN with several well-known kernel functions and parameter update methods. The experimental results show that Tomek-CASUWO can significantly improve the prediction accuracy of a classifier, and ILS-RBFNN outperforms other classification methods. We also conduct experiments on the extended real-world dataset. Finally, the robustness and applicability of ILS-RBFNN are verified on the eleven UCI datasets. All results show that the proposed two methods outperform existing models. The experimental results also demonstrate the effectiveness and practicability of ILS-RBFNN for predicting customer behavior.

1. Introduction

With the rapid development of economy, customers have multiple choices when purchasing products. Customers' consumption behavior appears to have an important impact on the survival and development of the enterprises. As a result, it is crucial to establish a proper prediction model of customer consumption behavior, which can guide enterprises with specialized marketing strategies (Vijayaragavan et al., 2018). With the growth of the Internet, data of consumer consumption behavior are growing as well. Hence, how to mine valuable information from such big data and use them to predict customer behavior has become particularly important (Koehn et al., 2020). Therefore, there is a need to adopt a proper prediction method to predict customer consumption behavior.

There have been relatively few studies on customer consumption behavior. Several deficiencies have been identified as follows: (1) Some current studies only use historical behavioral data to do the traditional regression analysis to analyze customers' behavior. This can lead to rough prediction results. (2) Many studies only investigate the demographic characteristics of customers. Hence, the obtained results are superficial, and it is hard to guide enterprises on decision making (Chu et al., 2019). (3) With the increasing amount of data in the databases, traditional analysis methods are not effective enough to meet the needs of diversified customer behavior analysis. Therefore, it is necessary to improve the traditional customer analysis methods to provide modern enterprises with more effective marketing decisions.

Data mining is a useful tool to study customer behavior, since it can

* Corresponding author at: College of Information and Electrical Engineering, China Agricultural University, 537#, No. 17, Qinghua East Road, Haidian District, Beijing 100083, China.

E-mail addresses: liyueleona@foxmail.com (Y. Li), sophiajianz@gmail.com (X. Jia), Ruili.wang@massey.ac.nz (R. Wang), qijianfangly@163.com (J. Qi), cnjinhaibin@gmail.com (H. Jin), chuxiaoquan1994@163.com (X. Chu), wsmu@cau.edu.cn (W. Mu).

https://doi.org/10.1016/j.eswa.2022.116982
Received 19 March 2021; Received in revised form 9 March 2022; Accepted 23 March 2022
Available online 26 March 2022
0957-4174/© 2022 Elsevier Ltd. All rights reserved.

discover hidden and potentially valuable customer information. Classification models in data mining have been widely used in customer behavior prediction, including decision trees, support vector machines, random forests, and artificial neural networks (Chiang et al., 2019; Kruzlicova et al., 2009; López-González et al., 2020; Lu et al., 2020; Rubio, 2021; Rubio et al., 2021; Rubio et al., 2022; Vargas, 2021). For example, Suh et al. (2004) utilized association rules to mine data on customer purchasing patterns and predicted customer consumption intentions to achieve precision marketing. Lee et al. (2008) predicted consumers' behavior in choosing hospitals based on the Back Propagation Neural Network (BPNN) model. Based on the results of this research, the evidence is enough to suggest that the neural network methods are useful in identifying customer behavior.

However, classification problems can reduce the effectiveness of prediction (Wang et al., 2018; Shen et al., 2019). Conventional classification methods usually assume that each category has the same number of samples (Anyfantis et al., 2007). The problem of class imbalance is inevitable in many real-world applications (Zhao & Li, 2017). The main reason for the imbalance is that the number of samples between classes are not balanced. The number of samples in one class is obviously less than that in other classes. Traditional classification methods use the overall classification accuracy as an evaluation index. Generally, the higher the overall classification accuracies, the better the prediction performances of the classifiers. For the imbalanced dataset, the predicted results of traditional classifiers tend to the majority classes, which will reduce the classification accuracies of the minority classes. Due to the imbalance of the data distributions in the real dataset, it is difficult to effectively evaluate a classification model. As a result, it is of great significance to study an effective classification method for imbalanced data (Gupta et al., 2019; Shilaskar & Ghatol, 2019; Gao et al., 2020).

The rest of this paper is organized as follows: Sections 2 and 3 describe the motivation and contribution of the work, respectively. Section 4 presents a review of related work. Section 5 describes the proposed oversampling technique and the improved RBFNN. Section 6 analyzes and discusses the experimental results of the proposed methods. Section 7 suggests the practical implications based on customer behavior prediction results. Section 8 summarizes the paper and gives the directions of future work.

2. Motivation

The objective of the present study is to develop an accurate predictor for customer consumption behavior based on an optimized neural network. To solve the problem of class imbalance, we propose a Tomek-CASUWO oversampling method. In this paper, the Tomek links algorithm is used to improve the quality of the synthetic samples, and the oversampling formula is modified based on ASUWO to reduce the possibility of fuzzy class boundaries. To improve the accuracy of consumer behavior prediction, we propose an ILS-RBFNN with the hybrid kernel model. The hybrid kernel is designed by combining Gaussian and Polynomial (Poly) to solve the function fitting problem. An Immune Algorithm (IA) is used to optimize the centers of the Radial Basis Function Neural Network (RBFNN), and Least-Squares (LS) is used to calculate the biases and weights to further improve the prediction accuracy. Our study demonstrates the advantages of designing the RBFNN kernel function and optimizing the parameters by IA for predicting consumer consumption behavior.

3. Contribution

The main contributions of this study include: (1) a new Tomek-CASUWO oversampling method, which can dramatically improve the prediction accuracy of a classifier, and (2) a new ILS-RBFNN classifier with the hybrid kernel, which can improve training efficiency, generalization ability, and global optimization performance. The prediction performances of our Tomek-CASUWO and ILS-RBFNN are analyzed and compared using wine-customer behavior data and eleven UCI datasets. The experimental results show that the proposed methods have better predictive performances than the compared methods. Predicting customer consumption behavior can help to understand customers' purchasing intentions, and is useful to guide enterprises with personalized marketing strategies.

4. Related work

In this section, we review the related work of imbalanced classification briefly, especially the classification problems in the imbalanced wine-consumer behavior dataset. In the proposed methods, the impact of data imbalance on the classifiers is reduced from two aspects, namely via data preprocessing and algorithm improvement. The purpose of data preprocessing is to generate artificial samples to form the balanced class distributions. Regarding algorithm improvement, focus is placed on the RBFNN model with parameter learning to further improve the overall performance of the RBFNN model.

4.1. Data preprocessing

From the perspective of data preprocessing, an existing solution to the problem of data imbalance is resampling. Resampling can change the imbalance ratio by adding minority samples or reducing the majority samples to construct a balanced synthetic dataset. The essence of resampling lies in its ability to help design classification methods (Sáez et al., 2016). Resampling techniques include undersampling and oversampling. Undersampling is an imbalanced data processing method that achieves equalization by removing samples from the majority classes. Undersampling methods include the compressed nearest-neighbor method and random deletion method (Lin et al., 2017). However, undersampling simply removes samples, and this can lead to the loss of important information. Compared with undersampling, oversampling improves the number of minority samples by adding samples to the minority classes. The simplest oversampling method is random oversampling, in which the synthesized minority samples are duplicated. However, this can lead to model overfitting. Another common method is the synthesis of samples from the data. The Synthetic Minority Oversampling Technique (SMOTE) is the most popular oversampling method, which adds genuinely new samples (Chawla et al., 2002). The idea of this algorithm is to synthesize artificial samples by interpolating between the minority samples and their nearest neighbors. The SMOTE can avoid the overfitting problem caused by mere replication and improve the generalization abilities of the classifiers on the minority classes (Douzas et al., 2018; Jian et al., 2016; Xu et al., 2020). Because SMOTE does not differentiate each sample for selection, redundant samples can be generated.

To improve the quality of the synthesized samples, researchers have proposed many solutions to solve the disadvantages of SMOTE (López et al., 2012). In 2005, Han et al. (2005) proposed a novel oversampling method, called Borderline-SMOTE, to oversample only the borderline examples of the minority classes, as these minority samples are the most

easily misclassified. This method effectively avoids generating redundant samples and enhances the class boundary information. He et al. (2008) proposed an Adaptive Synthetic Sampling Technology (ADASYN) to expand the minority samples based on the probability distributions of the minority samples. On the basis of the distribution characteristics of the original data, ADASYN can adaptively synthesize the minority samples to reduce the bias caused by the class imbalance. However, the ADASYN algorithm does not analyze the correlations between minority samples. The characteristics of the minority samples are not fully utilized, which can affect the prediction performances of the classification models. After that, Safe-Level-SMOTE generates minority samples along the weighted lines, and each synthetic sample is generated in a safe position by considering the safe level ratios of instances (Bunkhumpornpat et al., 2009). The samples synthesized by this method are concentrated in areas in which the minority samples are dense, thereby avoiding the random synthesis of SMOTE. In 2016, Nekooeimehr and Lai-Yuen (2016) proposed an Adaptive Semi-unsupervised Weighted Oversampling (ASUWO) technique. This algorithm mainly oversamples hard-to-learn samples close to the class boundaries based on the hierarchical clustering. However, in practical applications, human errors are not avoidable. Many features are not related to the target feature. If those samples are not filtered, the predictive performances of the classifiers can be affected. ASUWO has three main steps: semi-unsupervised clustering, adaptive subcluster sizing, and synthetic instance generation. In the process of sample generation, random linear interpolation can cause fuzzy class boundaries (Beyan & Fisher, 2015).

Although most methods can address the weaknesses of oversampling, few methods can simultaneously avoid noises and reduce the imbalance between and within classes. In addition, many techniques blindly synthesize new samples, and do not conduct reasonable sampling according to the characteristics of the class distributions. Therefore, there is still room for improvement in noise samples and generation methods in real-world applications (García et al., 2012).

4.2. Algorithm improvement

Predicting customer consumption behavior is a key link in the generation of incomes for enterprises. Many established prediction models have low accuracies, and improving the accuracies of prediction models is therefore an urgent issue that currently requires a resolution. In particular, the neural network-based methods have shown a stronger learning ability than other conventional classification methods (Yang et al., 2020). In many neural network models, the RBFNN is a single hidden layer of a three-front network. RBFNN has fewer control parameters and a higher training speed owing to its simple topology structure and strong approximation abilities for nonlinear behaviors. It has been widely used in research in predicting customer behavior (Lin et al., 2009; Liu et al., 2018). However, selecting the kernel and parameters of RBFNN is a difficult task. It is also quite challenging to properly select the network structure and optimize RBFNN parameters. The prediction accuracy of the RBFNN is usually affected by the fitting degree of its kernel functions, centers, biases, and weights (Dey et al., 2019). To prevent overfitting or underfitting of prediction results, it is essential to select the proper RBFNN parameters. Most improvement methods include the following two aspects: kernel function design and network parameter optimization.

For various network kernel functions, the Gaussian kernel is the most commonly used method (Kitayama & Yamazaki, 2011). However, recently, the Cosine kernel has provided more complementary information than the Gaussian kernel, and it makes the feature vectors more prominently distinguishable (Moinuddin et al., 2017). In some studies, it has been suggested that combining multiple kernels is more efficient than using individual kernels (Gu et al., 2016; Bucak et al., 2014). Aftab et al. (2014) proposed a multikernel function combining Gaussian and Cosine, which has some advantages over traditional single kernels. However, when a multikernel is used, the weights of the two kernels must usually be artificially adjusted to be equal. To resolve this problem, Khan et al. (2017) proposed an adaptive framework for an automatic fusion of the kernels. This approach tunes the kernel weights at every iteration. In 2020, Atif et al. (2020) proposed a novel RBFNN with an adaptive multikernel, where each kernel function has its own weight. Their experimental results showed that the flexibility of the multikernel, which can increase the convergence speed of RBFNN and make better local optimization.

In optimizing the network parameters, the selection of RBF centers directly affects the discrimination power of the RBFNN model. To locate the centers of those hidden layer neurons, random and K-means methods are normally used. The RBF centers are randomly selected from the training samples. However, this method only suits data with regular distributions. Also, the centers obtained by K-means are sensitive to the initial cluster centers (Oh et al., 2012). The classification results are very random due to the different initial randomly selected centers. Recently, evolutionary methods have been used to optimize the RBFNN model (Han et al., 2018). For example, Feng (2006) proposed a self-generation RBFNN by using the Particle Swarm Optimization (PSO) algorithm. In that study, PSO was used to adjust the centers, biases, and connection weights of RBFNN. The PSO algorithm is relatively simple and easy to implement, and there is no need to adjust too many parameters. The combination of PSO and RBFNN can optimize the parameters and structure of the RBFNN model, which can improve the prediction accuracy of the RBFNN classifier. Compared with the PSO, ant colony, and other intelligence algorithms, the IA is easy to understand, convergences rapidly, and is not likely to fall into a local optimum due to its memory and self-learning abilities (Duma & Twala, 2018; Wei et al., 2020). Because of those advantages, IA has been well applied to optimize the parameters of RBFNN (Peng et al., 2014). In the literature (Muling et al., 2018), IA was used to determine the centers and the number of hidden nodes of RBFNN. The experimental results demonstrate that this improved RBFNN has better prediction accuracy and generalization ability. In the training process of RBFNN, the selection of the parameters can seriously affect the prediction accuracy. How to choose the efficient parameters is still under investigation.

As mentioned previously, the ASUWO algorithm synthesizes the same number of new samples for each sample. It does not make full use of those more representative samples, which can generate redundant and noise samples. Furthermore, the RBFNN model has been widely used in classification problems. However, due to the influences of the kernel function and parameter learning, the prediction accuracy of the RBFNN can be further improved. Therefore, this paper focuses on methods that combine ASUWO oversampling technique with the RBFNN model.

5. Methods

This section introduces the related data preprocessing and classification method. The proposed oversampling technique and dimensionality reduction method are described in Section 5.1. The concept and optimization process of RBFNN are presented in Section 5.2.

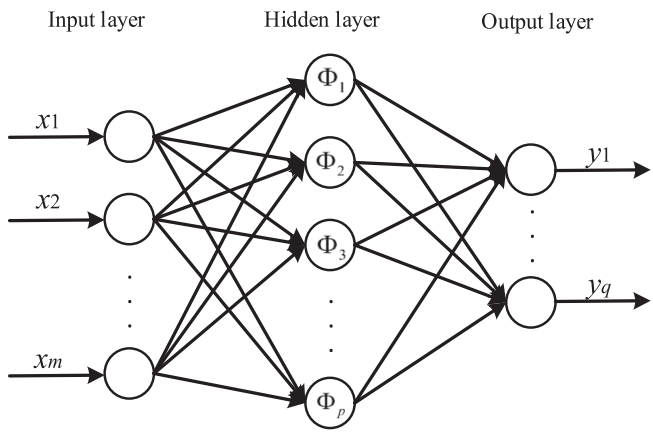


Fig. 1. Structure diagram of RBFNN.

5.1. Oversampling and feature processing

5.1.1. Tomek-CASUWO-based oversampling method

A balanced dataset can be obtained by data preprocessing. The ASUWO aims to balance class distributions by increasing the number of samples belonging to a minority group. On the one hand, the ASUWO can increase the majority class sample noises if a minority sample or nearest sample is a noise. On the other hand, the ASUWO synthesizes minority samples by linear difference, and it ignores the majority samples near the root data and increases the overlap of classes (Rivera, 2017). Therefore, the ASUWO algorithm is sensitive to noise samples and easy to cause fuzzy class boundaries. In order to solve these problems, we propose the following solutions:

(1) The combination of oversampling and undersampling technologies (hybrid sampling) is an effective way to balance a dataset, which can improve the generalization ability of the RBFNN classifier and reduce the probability of overfitting (Ren et al., 2019; García et al., 2006). We therefore introduce the Tomek links algorithm to eliminate the outliers and redundant samples before oversampling the minority classes. Tomek Links can be defined as follows: Suppose $\{E_1, E_2, \cdots, E_n\} \subset R$ is a dataset, given two examples $E_i = (x_i, y_i)$ and $E_j = (x_j, y_j)$, where $y_i \neq y_j$. A pair (E_i, E_j) is a Tomek link if E_i and E_j have different labels, and there is not a sample E_l satisfying the following conditions: $d(E_i, E_l) \langle d(E_i, E_j)$ or $d(E_j, E_l) \langle d(E_i, E_j)$, where $d(\cdot)$ is the distance between samples. If so, then one of the two examples is a noise, or they are close to the class boundaries. Filtering noises and outliers can ensure the quality of the synthesized samples, and greatly contribute to the prediction results.

(2) To reduce the probability of fuzzy class boundaries, we use minority centroids to modify the oversampling formula so that the synthetic samples are in the triangle region and close to the centroids. The oversampling formula is modified in Equation (1).

$$\begin{cases} x_{new1} = x_{cen} + sin\alpha|x_1 - x_{cen}| \\ x_{new2} = x_{cen} + sin\beta|x_2 - x_{cen}| \\ x_{new} = x_{cen} + sin\gamma|(x_{new1} + r|x_{new2} - x_{new1}|) - x_{cen}| \end{cases} \quad (1).$$

where x_1, x_2 are instances in the subcluster; x_{cen} represents the centroid of the subcluster; $\alpha, \beta \in \left[\frac{\pi}{4}, \frac{\pi}{2}\right]; \gamma \in [0, \frac{\pi}{4}]$; r is a random number between [0,1]. Our Tomek-CASUWO algorithm can eliminate the noise samples and reduce the probability of fuzzy class boundaries. Using our proposed Tomek-CASUWO, we can obtain a relatively clean balanced dataset and improve the quality of the synthetic samples.

5.1.2. Reduced dimensionality

Because of the overfitting of the RBFNN model after data oversampling, irrelevant and redundant features can reduce the network training speed and even make the prediction results less accurate. Principal Component Analysis (PCA) is a statistical method commonly used for dimensionality reduction (Nadkarni & Neves, 2018; Ahmadi et al., 2019). Its main purpose is to extract important information and compress the dataset dimensions. PCA can ensure that the extracted information between the principal components are uncorrelated (Jafarzadegan et al., 2019). The main steps of PCA are as follows:

(1) Standardize the raw data to eliminate the dimensional differences of all indicators.

$Z_{ij} = \frac{X_{ij} - \overline{X}_j}{S_j}, i = 1, 2, ..., n; j = 1, 2, ..., p$ (2).where p is the number of features; n is the number of samples; Z_{ij} is the standardized data of X_{ij}; $\overline{X}_j$ and S_j are the average value and standard deviation of X_j, respectively.

(2) Calculate the correlation coefficient matrix $R = [r_{ij}]_{p \times p}$.

$r_{ij} = \frac{\sum_{k=1}^{n} Z_{ki}^T \cdot Z_{kj}}{n-1}, (i, j = 1, 2, ..., p)$ (3).

(3) Calculate eigenvalues and eigenvectors. The p eigenvalues are obtained by calculating $|\lambda E - R| = 0$. These eigenvectors $(\mu_1, \mu_2, \cdots, \mu_p)$ are sorted according to their eigenvalues $(\lambda_1 > \lambda_2 > \cdots > \lambda_p \geq 0)$.

$U_j = \sum_{k=1}^{p} \mu_{kj} Z_k, j = 1, 2, ..., p$ (4).

(4) Calculate the cumulative contribution rate a_m of U_1, U_2, …, U_m. If $a_m \geq 90\%$, the first m principal components are selected, where m is smaller than p.

$a_m = \frac{\sum_{k=1}^{m} \lambda_k}{\sum_{k=1}^{p} \lambda_k}$ (5).

5.2. ILS-RBFNN-based customer consumption behavior prediction method

5.2.1. Basic principle of RBFNN

The conventional RBFNN is a typical three-layer neural network comprising an input layer, a hidden layer, and an output layer. The transformation from the input space to the hidden layer space is nonlinear, whereas the transformation from the hidden layer space to the output layer space is linear (Zounemat-Kermani et al., 2020). RBFNN has a strong nonlinear fitting ability, and can map any complex nonlinear functions. Its parameters can be learned separately. Also, RBFNN has strong robustness and self-learning ability and high convergence speed, and therefore is widely applied in the field of prediction (Yu et al., 2017). The topology of RBFNN is shown in Fig. 1. The RBFNN structure is *m-p-q*. m, p and q are the total number of input, hidden and output nodes, respectively; x_1, x_2, …, x_m are the inputs of RBFNN; $\varnothing_1, \varnothing_2, \cdots, \varnothing_p$ are the radial basis functions; y_1, …, y_q are the outputs of RBFNN. The matching relationship of the network is defined in Equation (6).

$y_i = f_i(x) = \sum_{k=1}^{p} \Phi_k(x, c_k) w_{ki}, i = 1, 2, ..., q$ (6).where w_{ki} is the weight from the kth hidden node to the ith output node; c_k is the center

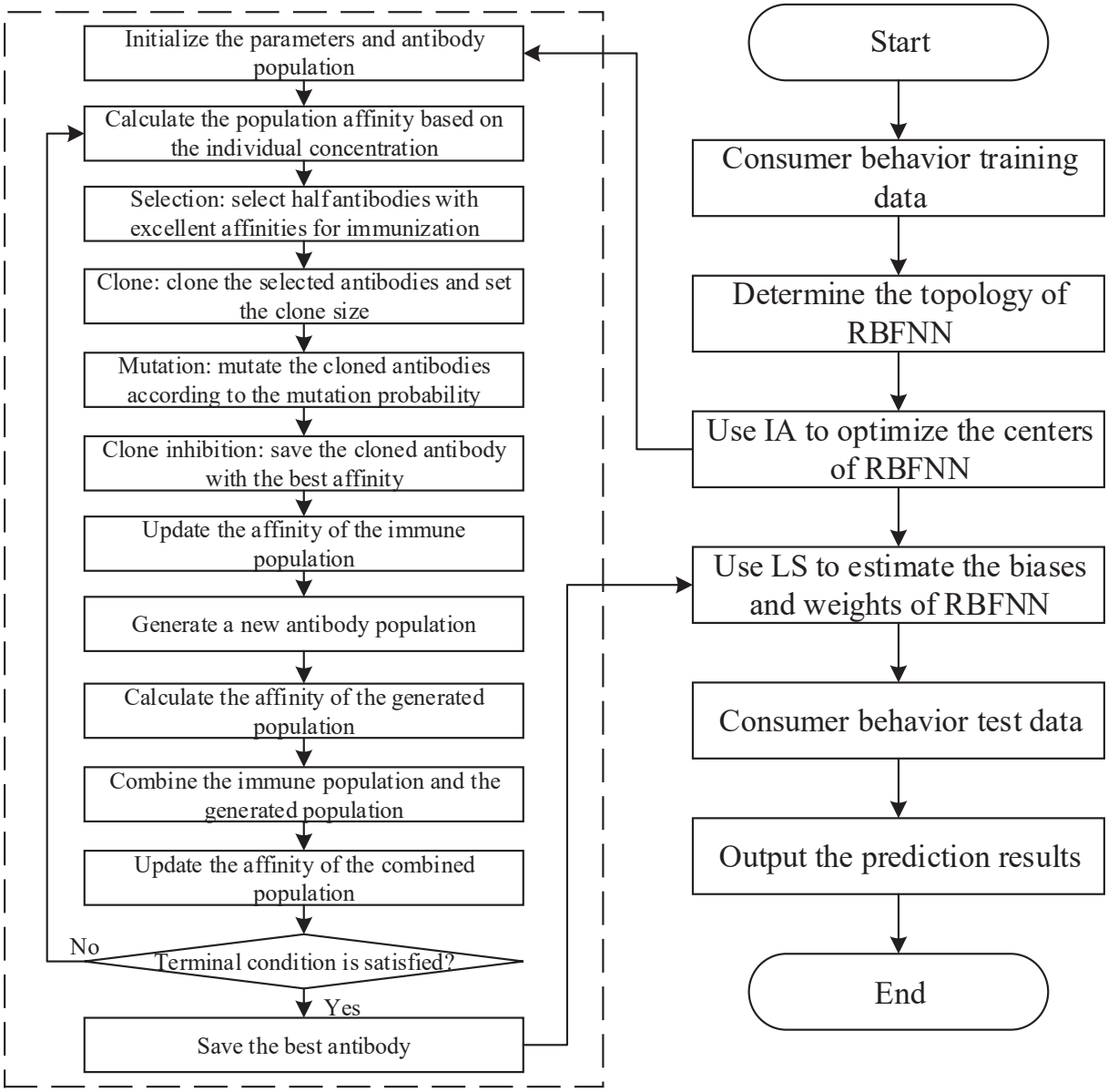


Fig. 2. The flowchart of ILS-RBFNN-based customer consumption behavior prediction model.

of the kth hidden node; $\varnothing_k(\cdot)$ is the kernel function of the kth hidden node. Generally, the Gaussian kernel function is used as the activation function of RBF, expressed as: $\varnothing_k(x, c_k) = \exp(-\frac{\|x-c_k\|^2}{2\sigma_k^2})$. σ is the expansion constant of neuron, which reflects the width of the function. Gaussian kernel function, also known as the RBF kernel, is a radially symmetric scalar function. It is usually defined as a monotonic function of the Euclidean distance between the sample and the data center. The essence of the Gaussian kernel is to measure the "similarity" between samples, and its effect is to find a new space to achieve nonlinear mapping (Yuan et al., 2009). In this study, the Gaussian kernel function is used to map each finite-dimensional data to a high-dimensional feature space, which allows the linearly inseparable data to be linearly separable.

In the RBFNN model, the number of network nodes is determined by the actual problem. Nevertheless, the hidden basis functions, centers, widths, and connection weights all play crucial roles in obtaining the desirable properties of RBFNN (Duvvuri & Anmala, 2019). Therefore, to find an accurate and stable customer behavior prediction model, we need to redesign the network structure and optimize the parameters of RBFNN, as described in Section 5.2.2.

5.2.2. Optimization and selection of network parameters

Before optimizing the network parameters, the number of hidden layer nodes must be determined because of its effect on the training speed and prediction accuracy of RBFNN (Li et al., 2019). In this study, the empirical Equation (7) is used to calculate the number of hidden nodes, where a is an integer between 1 and 10.

$$p = \sqrt{m+q} + a \quad (7).$$

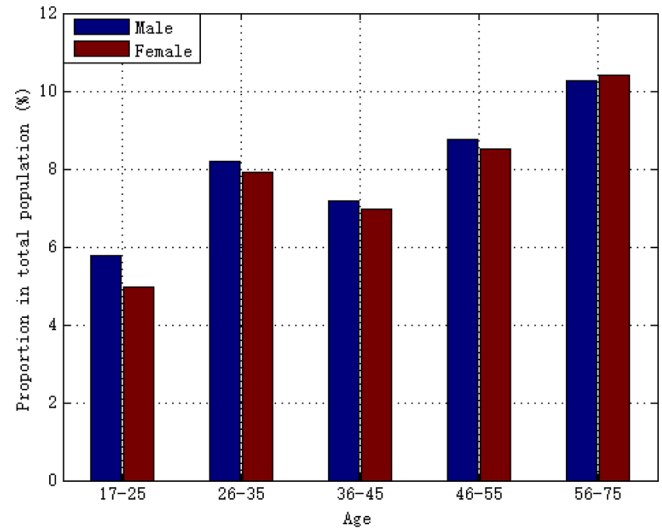
5.2.2.1. Kernel function design. To solve the problem of insufficient computing power caused by high-dimensional data, a kernel function is generally used (Hussain, 2019). It can map the eigenvectors of the low-dimensional space to the separable high-dimension space (Shastry et al., 2017). It is necessary to construct a suitable kernel function for a real-world dataset. Generally, kernel functions can be divided into local and global kernels. A local Gaussian kernel with good learning ability is used the most to train RBFNN, whereas a global Poly kernel has stronger generalization ability (Zhang et al., 2016). For practical applications with complex data, a hybrid kernel function combining a series of kernels is more suitable than a single kernel function (Monney et al., 2020). Here, to create an adaptive hybrid kernel and improve the generalization ability of RBFNN, we combine Gaussian and Poly kernels. The nonlinear hybrid kernel is defined in Equation (8), and the overall outputs of RBFNN can be written as Equation (9).

$$\Phi_k(x, c_k) = \alpha_{1_k} exp(-\frac{\|x-c_k\|^2}{2\sigma_k^2}) + \alpha_{2_k}(x \cdot c_k + t)^d = \sum_{l=1}^{2} \alpha_{l_k} \Phi_{l_k}(x, c_k) \quad (8).$$

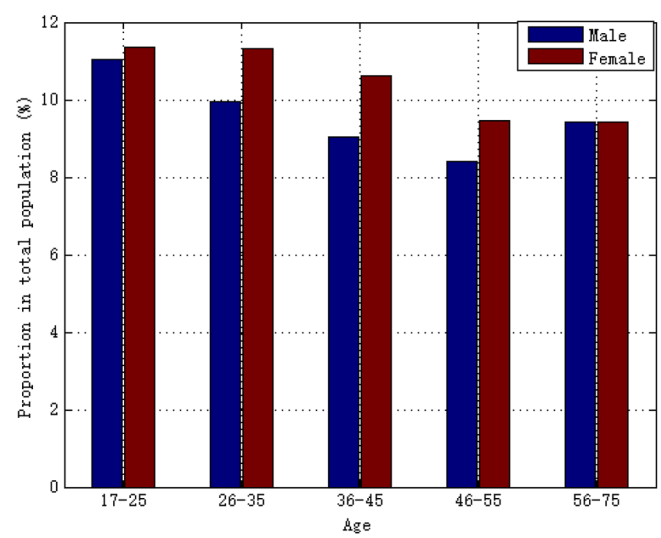
$$y = (\sum_{l=1}^{2} \alpha_{l_k} \Phi_{l_k}(x, c_k)) \sum_{k=1}^{p} w_k = \sum_{k,l} \Phi_{l_k}(x, c_k)(w_k \alpha_{l_k}) = \sum_{k,l} \Phi_{l_k}(x, c_k) w_{k,l} \quad (9).$$

where x is a sample; σ_k is the bias of the kth hidden node; t is a Poly constant ($t > 0$); d is the Poly order ($d = 1,2,3,\ldots$); α_{1_k} and α_{2_k} represent the weights assigned to Gaussian and Poly kernels of the kth radial basis, respectively; $w_{k,l}$is the substitute form of the weight of lth participating kernel in the kth radial basis.

5.2.2.2. Center, bias, and weight update rules. The selection of kernel parameters also plays a key role in RBFNN optimization (Wu et al., 2016). To improve the prediction accuracy, we use the IA algorithm to optimize the centers of RBF. IA is a heuristic search algorithm that imitates a biological immune system to solve the optimization problems. IA is usually used to find a global optimal solution that has a good balance between population diversity and convergence, and restrains the immature convergence (Duma & Twala, 2019). An antigen is defined as the objective function of the actual problem, and an antibody can be regarded as a candidate solution to the actual problem (Chiu et al., 2009; Samigulina & Samigulina, 2019). Due to the diversities of antibodies, IA can prevent the RBFNN model from falling into a local optimum in determining the RBFNN centers (Khan et al., 2020). The main steps of the IA optimized RBFNN centers are described as follows.



(a) China Statistical Yearbook data.

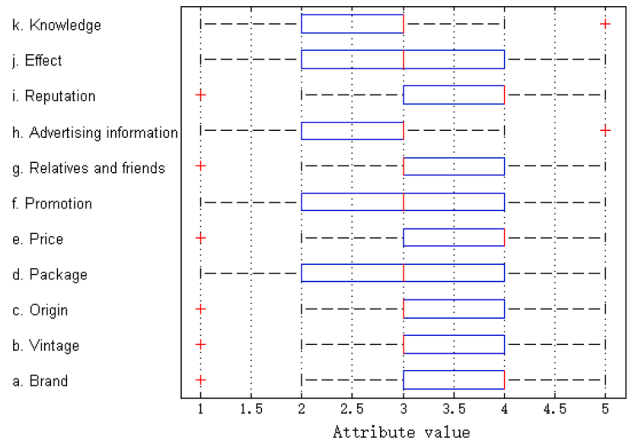


(b) Sample data.

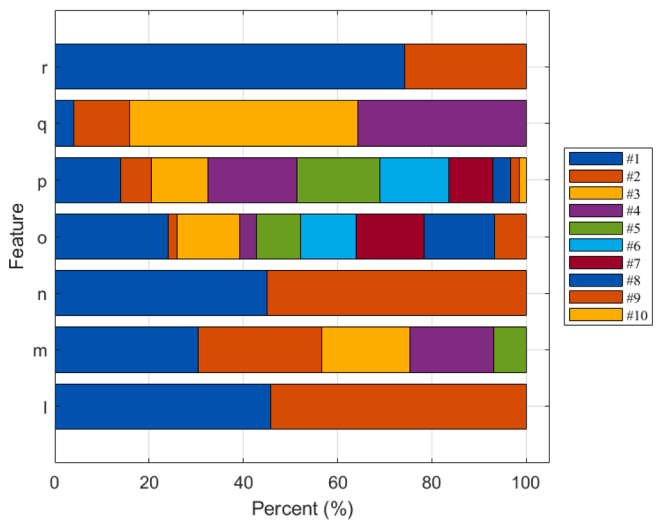
Fig. 3. Gender-age distribution between the national census data and the sample data.

Step 1: Initialize the antibody population. The real-coded population $A=\{c_1,c_2,c_M\}$ is randomly initialized to have M individuals, as shown in Equation (10). Antibodies $c_i=\{c_{1i}, c_{2i}, c_{pi}\}$ are the centers of the network hidden layer; D is the dimension of each antibody; $X_{\max}$ and $X_{\min}$ are the upper and lower limits of antibody, respectively.

$$c = X_{min} + rand(D, M) \times (X_{max} - X_{min}) \quad (10).$$



(a) Box plot of numerical variables.



(b) Proportion of categorical variables.

Fig. 4. Visualization of data feature distributions.

Step 2: Calculate the affinity (*AFF*) of the antibody population, as shown in Equation (11). α and β are the affinity coefficients. The output error (Fit_i) of RBFNN is used as the measurement for evaluating the *i*th antibody, shown in Equation (12). Acc_i is the accuracy of the test data of the *i*th antibody. The method for calculating the concentration (Nd_i) of the *i*th antibody, shown in Equation (13). $nd_{i,j}$ is the concentration between the *i*th antibody c_i and the *j*th antibody c_j; δ is the similarity threshold between antibodies. This method not only maintains the diversity of the population, but also speeds up convergence.

$$AFF = \alpha \cdot Fit - \beta \cdot Nd \quad (11).$$

$$Fit_i = 1 - Acc_i \quad (12).$$

$$Nd_i = \frac{\sum_{j=1}^{M} nd_{i,j}}{M} \quad (13).$$

$$nd_{i,j} = \begin{cases} 1, \sqrt{(c_i - c_j)^2} < \delta \\ 0, \sqrt{(c_i - c_j)^2} \geqslant \delta \end{cases}, i, j = 1, 2, \ldots, M \quad (14).$$

Table 1
Description of the basic characteristics of wine customers.

#	Feature	Feature description	Average value
a	Brand	a1. Unimportant; a2. Slightly important; a3. Generally important; a4. Very important; a5. Especially important	3.395
b	Vintage	b1. Unimportant; b2. Slightly important; b3. Generally important; b4. Very important; b5. Especially important	3.080
c	Origin	c1. Unimportant; c2. Slightly important; c3. Generally important; c4. Very important; c5. Especially important	3.276
d	Package	d1. Unimportant; d2. Slightly important; d3. Generally important; d4. Very important; d5. Especially important	3.025
e	Price	e1. Unimportant; e2. Slightly important; e3. Generally important; e4. Very important; e5. Especially important	3.452
f	Promotion	f1. Unimportant; f2. Slightly important; f3. Generally important; f4. Very important; f5. Especially important	2.916
g	Relatives and friends	g1. Unimportant; g2. Slightly important; g3. Generally important; g4. Very important; g5. Especially important	3.290
h	Advertising information	h1. Unimportant; h2. Slightly important; h3. Generally important; h4. Very important; h5. Especially important	2.536
i	Reputation	i1. Unimportant; i2. Slightly important; i3. Generally important; i4. Very important; i5. Especially important	3.684
j	Effect	j1. Unimportant; j2. Slightly important; j3. Generally important; j4. Very important; j5. Especially important	2.717
k	Knowledge	k1. Unimportant; k2. Slightly important; k3. Generally important; k4. Very important; k5. Especially important	2.799
l	Gender	l1. Male; l2. Female	1.542
m	Age	m1. 18–25; m2. 26–35; m3. 36–45; m4. 46–55; m5. > 55 years old	2.445
n	Marital status	n1. Unmarried; n2. Married	1.549
o	Occupation	o1. Student; o2. Farmer; o3. Freelancer; o4.Retiree; o5. State-owned enterprise; o6. Foreign / private enterprise; o7. Public institution; o8. Scientific research institution; o9. Other	4.801
p	Monthly income	p1. < 1500; p2. 1500–2000; p3. 2001–3000; p4. 3001–5000; p5. 5001–7000; p6. 7001–10000; p7. 10001–15000; p8. 15001–20000; p9. 20001–30000; p10. > 30000 yuan	4.411
q	Education	q1. Junior high school or below; q2. High school; q3. Undergraduate; q4. Postgraduate or above	3.160
r	Regional economic development	r1. Urban; r2. Rural	1.257

Step 3: Introduce the following operators to generate the immune population, and the size of the immune population is $M/2$. (1) Selection: select the first $M/2$ individuals with excellent affinities from the current population; (2) Clone: clone the selected individuals, and set the clone number for each individual. The clone operator searches for the antibodies with good affinities in order to get a better solution; (3) Mutation: mute the cloned antibodies according to the mutation probability, which can maintain the diversity of the population. As shown in Equation (15). $cc_{d,i}$ is the *d*-dimensional data of the *i*th sub-antibody of the antibody *c*; τ is the initial value of the neighborhood; *t* is the current iteration number; (4) Clone inhibition:

Table 2
Confusion matrix.

	Positive class	Negative class
Positive prediction	True Positive (TP)	False Positive (FP)
Negative prediction	False Negative (FN)	True Negative (TN)

update the affinity of the cloned population of each individual, and save the antibody with the best affinity.

$cc_i = cc_{d,i} + (rand - 0.5)\frac{\tau}{t}, i = 1, 2, ..., M_2, d = 1, 2, ..., D$ (15).

Step 4: Regenerate the individuals randomly, and the size of the new population is $M/2$.
Step 5: Update the antibody population. The immune population and the generated population are combined to get a new population. Then update the affinity of the combined population.
Step 6: Check whether the maximum number of iterations is reached. If satisfied, select the optimal antibody of the population, which is the central set of the hidden layer; otherwise, jump to Step 2 for further evolution.

After the RBFNN centers are determined by IA, the widths and connection weights are carried out through the LS method, as shown in Equations (16) and (17), and the Equation (17) represents the weight update rule.

$\sigma_k = \frac{d_{max}}{\sqrt{2p}} = \frac{max(||x - c_k||)}{\sqrt{2p}}, k = 1, 2, ..., p$ (16)

$w = (\Phi^T\Phi)^{-1}\Phi^T L$ (17)where L is the target outputs of the RBFNN model. Fig. 2 shows the improved RBFNN flowchart of this study. As a consequence, we propose a novel RBFNN learning model by combining the IA with memory and self-organization capabilities with LS. The hidden centers are designed by the IA, which reaches the optimal structure of RBFNN and improves its classification accuracy.

6. Procedures and results

6.1. Data collection

We collected data from the survey of wine-customer consumption behavior done by the Chinese Grape Industry Technology System from June to August 2020. The respondents consisted of ordinary urban consumers classified by gender, age, marital status, occupation, monthly income, education, and the degree of regional economic development. The quota sampling method commonly used in market research is used to determine the sample distribution (Li et al., 2021). Quota indicators and the range of values refer to the ratio of gender to age in the 2020 China Statistical Yearbook (https://www.stats.gov.cn/tjsj/ndsj/2020/indexch.htm) (Zhang et al., 2021). Determined by the degree of regional economic development, different sample sizes were assigned in each region. The data obtained from the statistics are representative and can reflect the characteristics of wine consumers in China to a certain extent. The comparison results between the national census data and the sample data are presented in Fig. 3. Fig. 3 shows that the sample gender ratio is balanced, and the age distribution is similar to the data in the China Statistical Yearbook.

The dataset comprises 879 customer records of three types of consumption purposes: 190 customers like domestic wine, 135 customers like imported wine, and 554 customers have no preferences on the wine producing areas. The behavior data of each consumer comprises 18

Table 3
Evaluation results for different classification methods on the real dataset.

Oversampling method	Index	LR	DT	RF	NB	KNN	SVM	BPNN	RBFNN
No oversampling	Accuracy	0.5282	0.6322	0.5322	0.5897	0.6322	0.6782	0.5552	0.6667
	Precision	0.4146	0.4846	0.4235	0.5012	0.4904	0.4911	0.4484	0.4723
	Recall	0.4285	0.4614	0.5670	0.4152	0.5456	0.5654	0.5874	0.5251
	F-measure	0.4054	0.4727	0.6006	0.4732	0.5165	0.5510	0.4274	0.4987
	G-means	0.4092	0.5155	0.4953	0.5416	0.5276	0.3617	0.5776	0.6003
	AUC	0.5086	0.6293	0.7155	0.6810	0.6810	0.7586	0.7414	0.6845
	Running time	0.7713	1.4943	1.8783	0.2367	0.9300	1.1810	0.3935	0.3862
SMOTE	Accuracy	0.5091	0.6788	0.6485	0.6848	0.6066	0.7212	0.6424	0.7364
	Precision	0.4762	0.5060	0.6305	0.5239	0.4633	0.5674	0.4798	0.6693
	Recall	0.4874	0.5109	0.5331	0.5311	0.6075	0.5939	0.5983	0.6732
	F-measure	0.3817	0.5084	0.5318	0.5275	0.6272	0.5804	0.4889	0.5712
	G-means	0.5150	0.6211	0.5973	0.6366	0.7190	0.6797	0.6051	0.5907
	AUC	0.6318	0.7591	0.6864	0.7636	0.7818	0.7309	0.7318	0.7273
	Running time	0.9091	1.5895	2.6187	0.3841	1.2109	1.2818	0.7201	0.5128
ASUWO	Accuracy	0.5802	0.8272	0.7889	0.8333	0.8272	0.7580	0.8148	0.8399
	Precision	0.4294	0.6948	0.7775	0.6963	0.6908	0.7501	0.7808	0.8375
	Recall	0.5333	0.6967	0.8043	0.7130	0.7610	0.7538	0.7886	0.8572
	F-measure	0.4313	0.6957	0.7907	0.7046	0.7242	0.7519	0.7847	0.8772
	G-means	0.5575	0.7695	0.7447	0.7773	0.7961	0.8130	0.7612	0.8528
	AUC	0.6852	0.8704	0.8167	0.8750	0.8704	0.8935	0.8611	0.8844
	Running time	0.8247	1.5154	2.1931	0.3441	1.2091	1.2275	0.5476	0.4889
Tomek-CASUWO	Accuracy	0.6221	0.8240	0.8870	0.9201	0.8481	0.8610	0.8610	0.9351
	Precision	0.7556	0.8175	0.8727	0.8043	0.8002	0.8242	0.8242	0.8697
	Recall	0.6611	0.8095	0.8747	0.7562	0.8021	0.8242	0.8242	0.8853
	F-measure	0.6583	0.8185	0.8737	0.7990	0.8012	0.8242	0.8242	0.8774
	G-means	0.6949	0.8120	0.8808	0.8102	0.8267	0.8437	0.8437	0.9104
	AUC	0.7416	0.8805	0.8190	0.9069	0.8610	0.8708	0.8708	0.9616
	Running time	0.6208	1.3019	1.9784	0.2939	0.8242	1.1217	0.4518	0.4279

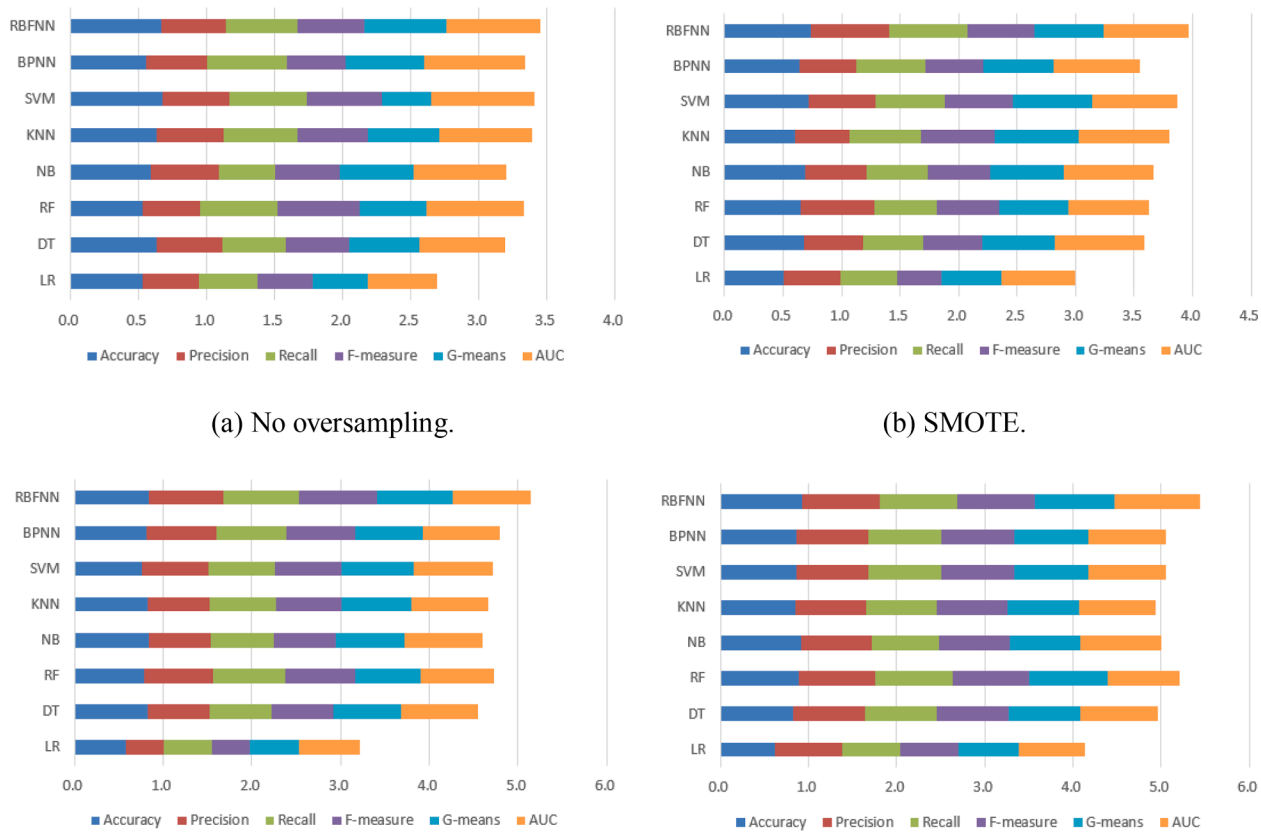
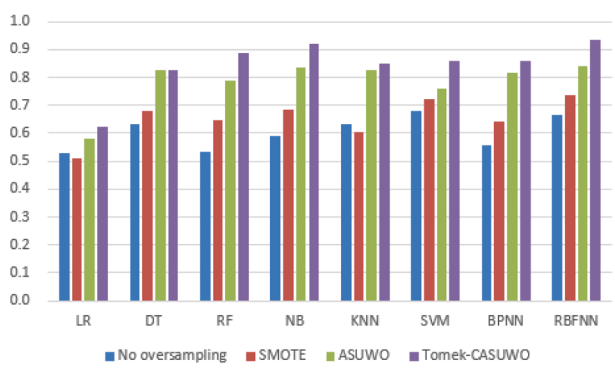
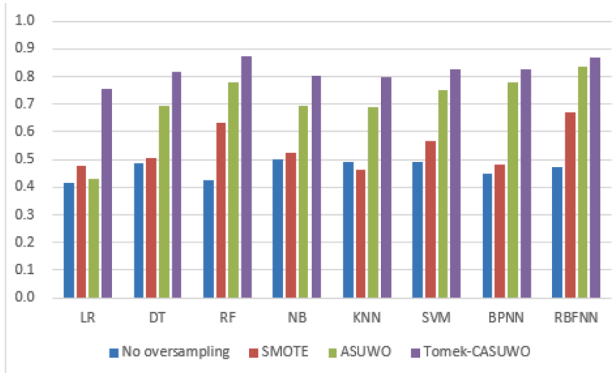


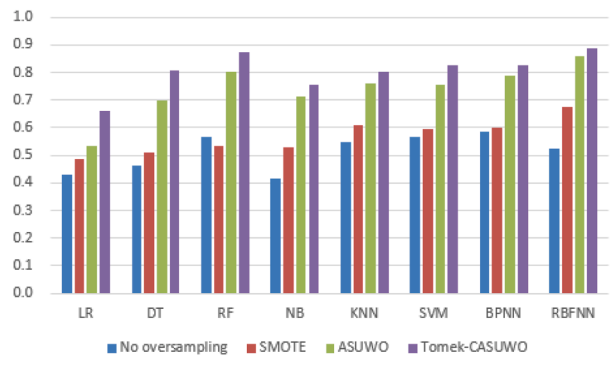
Fig. 5. Overlay diagrams of evaluation indicators in different classifiers.



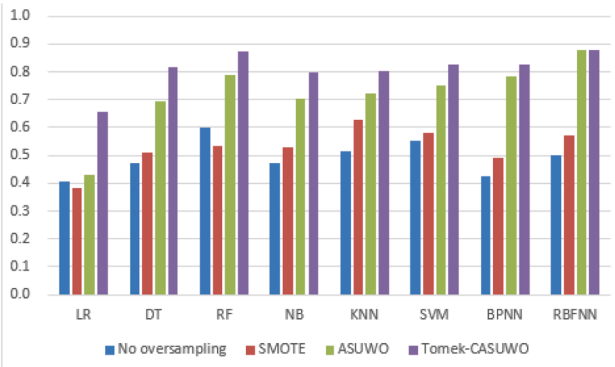
(a) Accuracy.



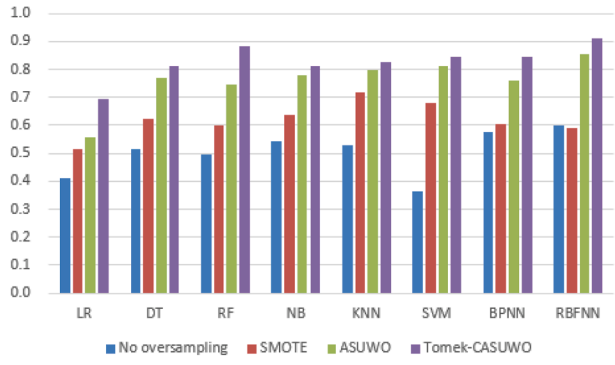
(b) Precision.



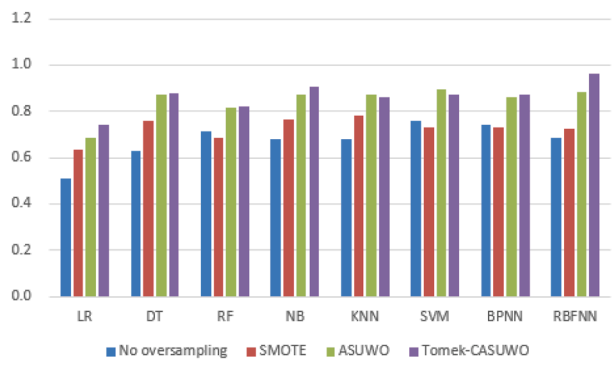
(c) Recall.



(d) F-measure.



(e) G-means.



(f) AUC.

Fig. 6. Evaluation indicators of different oversampling methods.

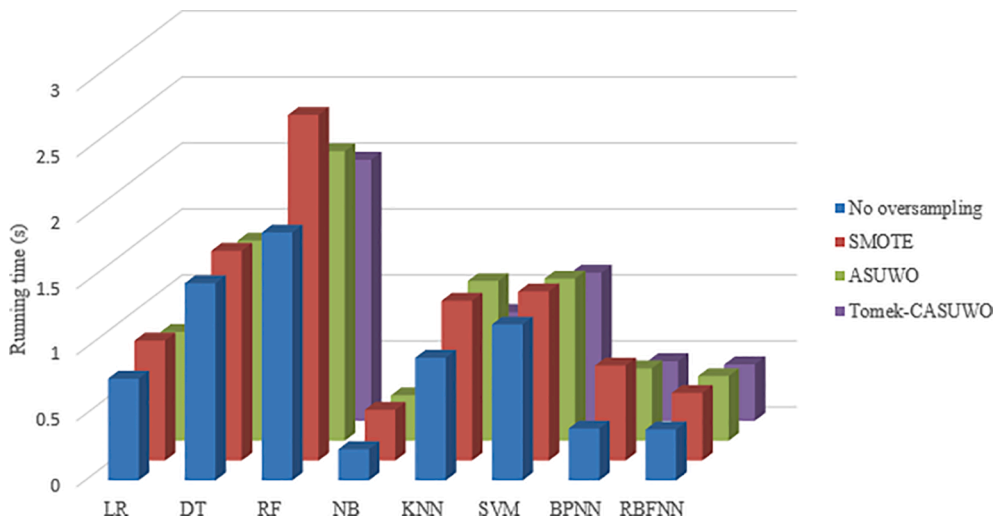


Fig. 7. Running times of the eight classifiers under different oversampling methods.

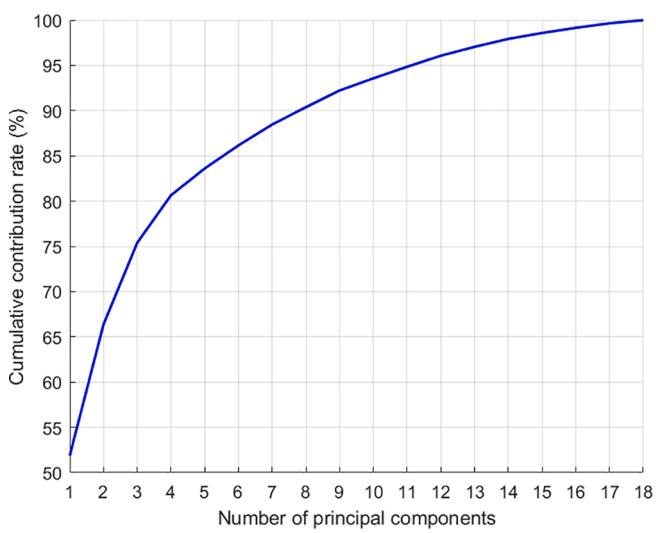


Fig. 8. PCA-based principal component information.

characteristics, including 11 numerical attributes and 7 categorical attributes. Fig. 4(a) shows a box plot of the numerical attributes. It indicates the degree of influence of each characteristic on consumer behavior: unimportant, slightly important, generally important, very important, and especially important. There are some outliers and there is no sampling deviations between the numerical variables. Fig. 4(b) shows the proportions of categorical variables per feature. All the features are encoded into numerical features for subsequent customer behavior prediction. The characteristics of numerical and categorical attributes are described in Table 1.

We implement the models using MATLAB R2014a based on the personal information and consumption behavior preferences of wine customers. To ensure the models are generalized and not overfitting, we use ten-fold cross-validation to calculate the experimental results.

6.2. Performance criteria

To solve the imbalanced classification problems, seven evaluation indexes are used to comprehensively evaluate the predictive performances of our methods: Accuracy, Precision, Recall, F-measure, G-means, AUC, and running time (Li & Mao, 2014; Galar et al., 2012). Higher values of Accuracy, Precision, Recall, F-measure, G-means, and AUC indexes represent better predictive performances of the classifiers. These evaluation indexes are calculated through the confusion matrix, as shown in Table 2. TP is the number of positive samples that are predicted as positive; FN is the number of positive samples that are predicted as negative; FP is the number of negative samples that are predicted as positive; TN is the number of negative samples that are predicted as negative. Six evaluation metrics are calculated as follows:

$$\text{Accuracy} = \frac{\text{TP}+\text{TN}}{\text{TP}+\text{TN}+\text{FP}+\text{FN}} \quad (18).$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP}+\text{FP}} \quad (19).$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP}+\text{FN}} \quad (20).$$

$$\text{F - measure} = \frac{(1+\beta^2)\text{Precision}\times Recall}{\beta^2\text{Precision}+Recall} \quad (21).$$

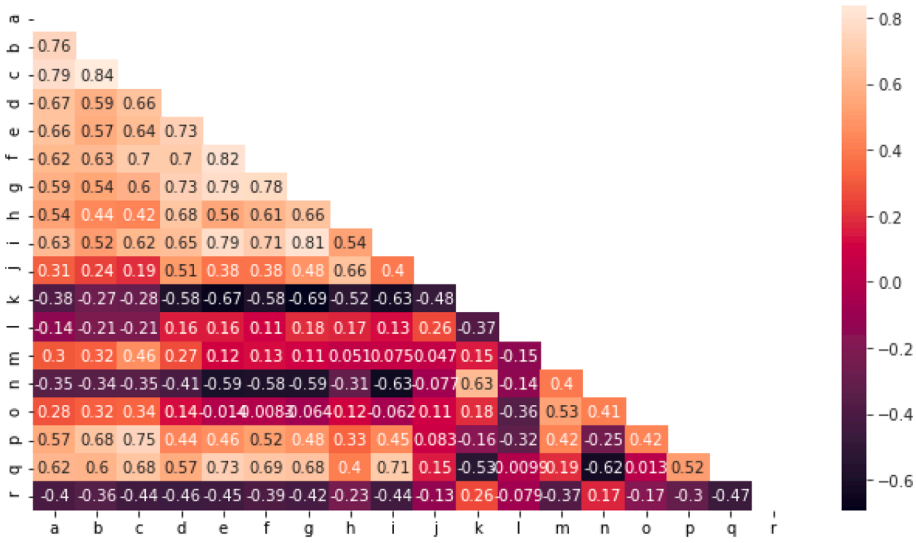
$$\text{G - means} = \sqrt{\frac{\text{TP}}{\text{TP}+\text{FN}}\times\frac{\text{TN}}{\text{FP}+\text{TN}}} \quad (22).$$

$$\text{AUC} = \frac{S_p-n_p\times(n_p+1)\times 0.5}{n_p n_n} \quad (23).$$

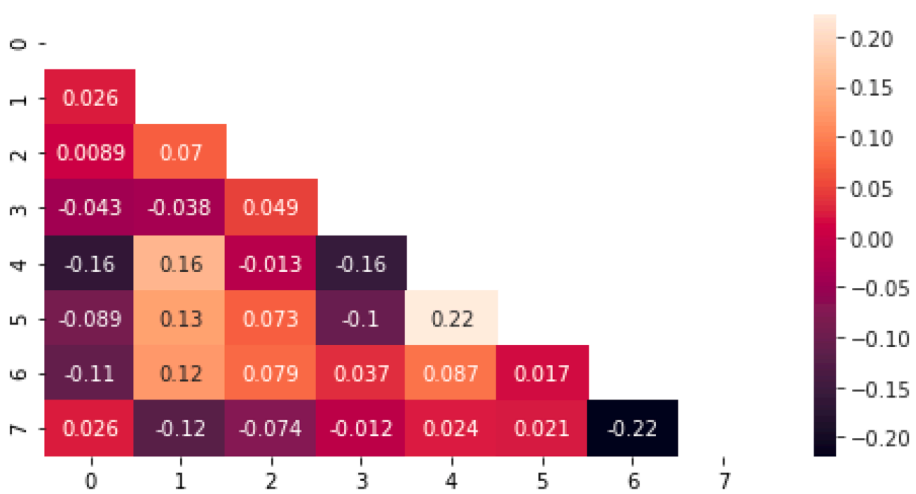
where β is a tunable parameter (usually $\beta = 1$); S_p is the sum of the positive ranks; n_p is the number which belongs to positive in the test data; n_n is the number which belongs to negative in the test data. Receiver Operation Characteristic (ROC) consists of a series of TP-FP pairs to demonstrate the predictive performances of the classifiers. The advantage of ROC is that it is not affected by the sample distributions and can objectively evaluate the classifiers. Sometimes, it is difficult to distinguish the prediction results from the ROC curve. The area under the ROC curve (AUC) is usually used to assess the imbalance classification by calculating the ratio of both true positive and false positive (Huang & Kechadi, 2013).

6.3. Performance analysis of the proposed methods based on the real-world dataset

To verify the methods of consumer behavior prediction, two experiments are compared. The first experiment is for evaluating the advantages of the Tomek-CASUWO oversampling technique, and the second experiment is for comparing the proposed ILS-RBFNN with the hybrid kernel to well-known existing kernel functions and parameter update methods.

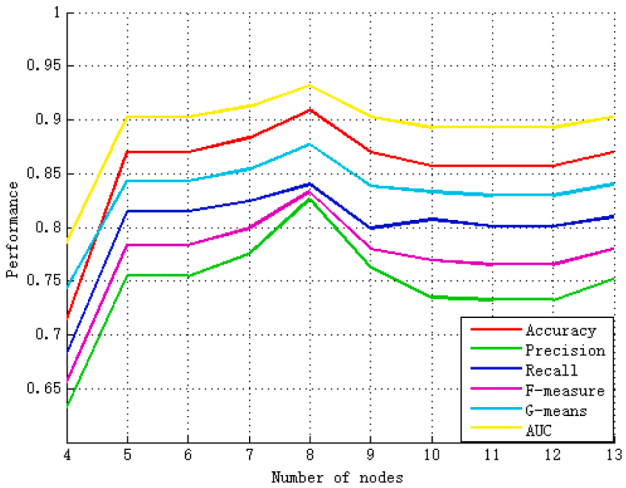


(a) Correlations of raw data after oversampling.

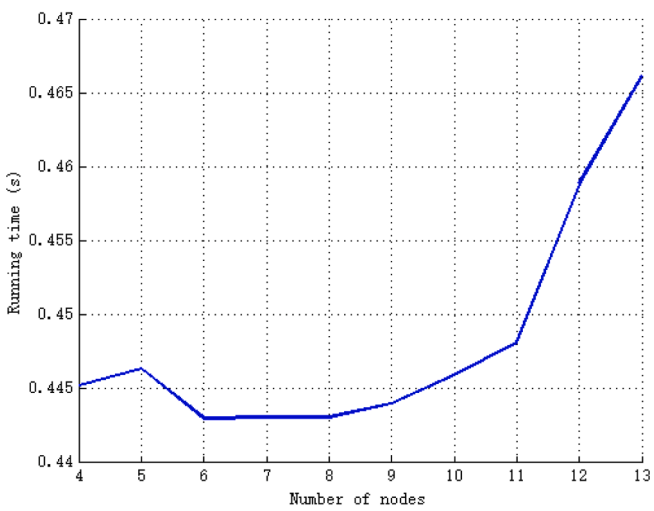


(b) Correlations between different features after PCA.

Fig. 9. Spearman correlation coefficients in the heat maps.



(a) Accuracy, Precision, Recall, F-measure, G-means, and AUC.



(b) Running time.

Fig. 10. Evaluation metrics corresponding to different number of hidden nodes.

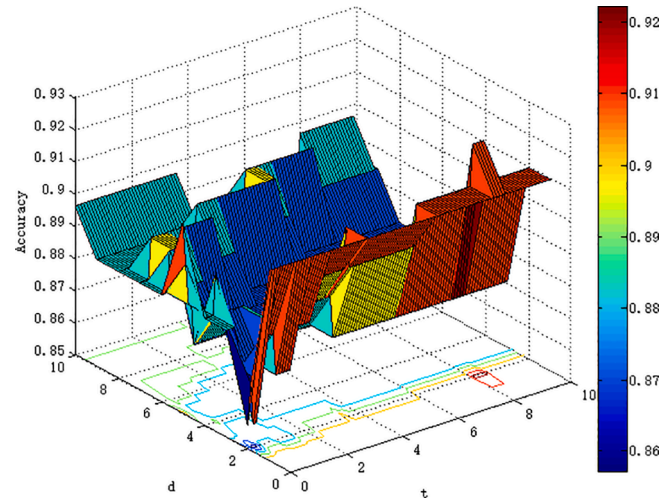


Fig. 11. Poly parameters in the proposed kernel method.

6.3.1. *Comparative experiments of oversampling methods*

In this experiment, we use 8 well-known classifiers to evaluate the models for predicting the consumption behavior of wine customers, such as Logistic Regression (LR), Decision Tree (DT), Random Forest (RF), Naive Bayes (NB), K-Nearest Neighbor (KNN), Support Vector Machine (SVM), BPNN, and RBFNN (kernel: Gaussian; centers: K-means; biases and weights: LS). Due to the class imbalance in the customer consumption behavior data (the imbalance ratio is 190:135:554 = 1.41:1:4.10), the prediction accuracies of classifiers are seriously affected. Hence, it is necessary to oversample the raw data. The related oversampling techniques are separately used to improve classification performances of the classifiers. After SMOTE, ASUWO, and Tomek-CASUWO oversampling, the feature count before and after oversampling is the same as the original, and the sample sizes are 1662, 1632, and 776, respectively. Table 3 compares the proposed Tomek-CASUWO with no oversampling, SMOTE (Chawla et al., 2002), and ASUWO (Nekooeimehr & Lai-Yuen, 2016) methods. To select the optimal classifier, Fig. 5 visually shows the prediction results of different classifiers. To compare the proposed Tomek-CASUWO and other oversampling methods, Fig. 6 visualizes the Accuracy, Precision, Recall, F-measure, G-means, and AUC evaluation indicators of different oversampling methods. Fig. 7 shows the running times of the eight classifiers under different oversampling methods.

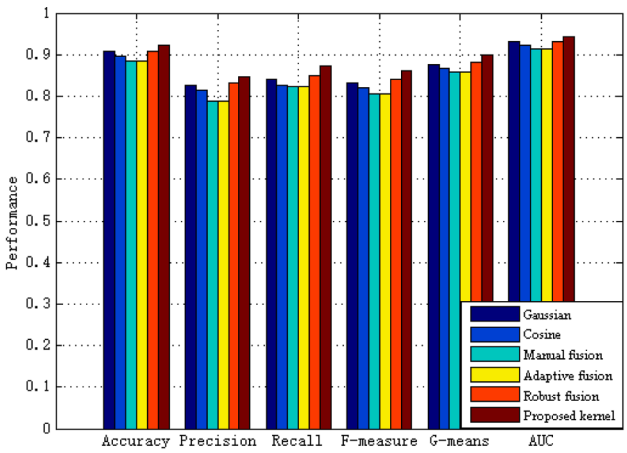
As described in Table 3, we can conclude as follows: (1) Fig. 5 illustrates that the RBFNN has the highest cumulative performance. The experimental results demonstrate that the RBFNN model outperforms the other compared methods for the imbalanced dataset and has better practical value. (2) Fig. 6 indicates that the Tomek-CASUWO can significantly improve the performances of the classifiers compared with no oversampling, SMOTE, and ASUWO oversampling methods. This is because the Tomek Links algorithm is used to filter noises and outliers in the original data, and then the subcluster centroids are used to synthesize the new samples. Hence, overlapping class boundaries can be effectively prevented. (3) A suitable prediction algorithm is usually selected according to the actual application problem to analyze the running times of the classifiers. The running efficiency of the algorithm is related to the number and dimension of samples. After balancing the real-world dataset using the SMOTE, ASUWO, and Tomek-CASUWO algorithms, the sample sizes are 1662, 1632, and 776, respectively. Fig. 7 shows that the RBFNN model has good time efficiency, which meets the needs of this case. Therefore, RBFNN is used to predict customer consumption behavior, and the proposed Tomek-CASUWO is used to synthesize new samples.

6.3.2. *Comparative experiments between ILS-RBFNN with the hybrid kernel and state-of-the-art methods*

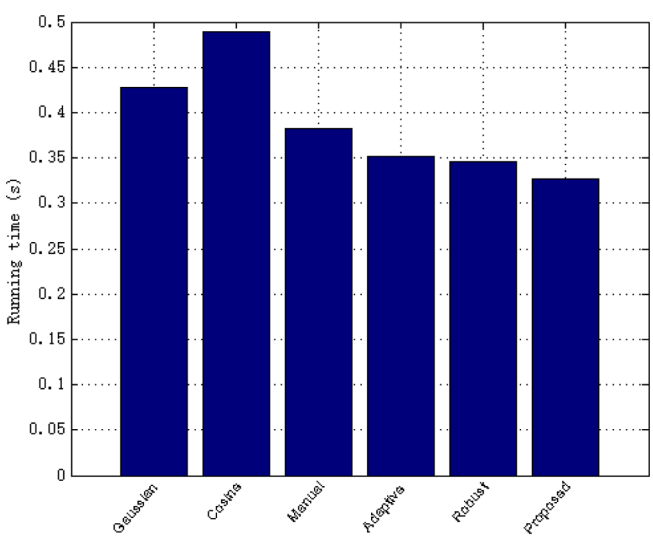
After data oversampling, all the original features are highly correlated. It is important to reduce its dimensions, otherwise the training efficiency of the RBFNN model and the prediction effect can be affected. Compared to the full features, although PCA loses some useful information, its computational cost to train the RBFNN is reduced, and it can avoid model overfitting caused by oversampling. Thus, PCA is a good choice for dimensionality reduction. Fig. 8 displays the proportions of principal components, which shows that the cumulative contribution rate reaches 90.38% when the number of principal components is 8. Fig. 9(a) shows the correlations between the original features. Fig. 9(b) visualizes the values of the pair-wise Spearman correlation coefficients between different features after PCA. In this figure, the first eight

Table 4
Evaluation metrics of different kernel function methods.

Kernel function	Gaussian	Cosine	Manual fusion	Adaptive fusion	Robust fusion	Proposed kernel
Accuracy	0.9091	0.8961	0.8831	0.8831	0.9091	**0.9221**
Precision	0.8261	0.8150	0.7883	0.7883	0.8309	**0.8469**
Recall	0.8400	0.8253	0.8232	0.8232	0.8484	**0.8737**
F-measure	0.8330	0.8201	0.8054	0.8054	0.8396	**0.8601**
G-means	0.8767	0.8666	0.8577	0.8577	0.8820	**0.8985**
AUC	0.9318	0.9221	0.9123	0.9123	0.9318	**0.9416**



(a) Accuracy, Precision, Recall, F-measure, G-means, and AUC.



(b) Running time.

Fig. 12. Evaluation metrics of different kernel function methods.

principal components are the basic matrices, and there are no obvious correlations between the extracted features.

After that, we analyze the proposed kernel function and parameter update method based on the oversampled and selected features. Fig. 10 reveals the evaluation metrics when the number of hidden nodes is between 4 and 13. Fig. 10(a) clearly shows that when the number of hidden nodes is 8, Accuracy, Precision, Recall, F-measure, G-means, and AUC values of the RBFNN classifier achieve the largest. Fig. 10(b) illustrates that the RBFNN classifier has a faster running time when the number of hidden nodes is 8. Therefore, the structure of RBFNN determined in this study is 8–8-3.

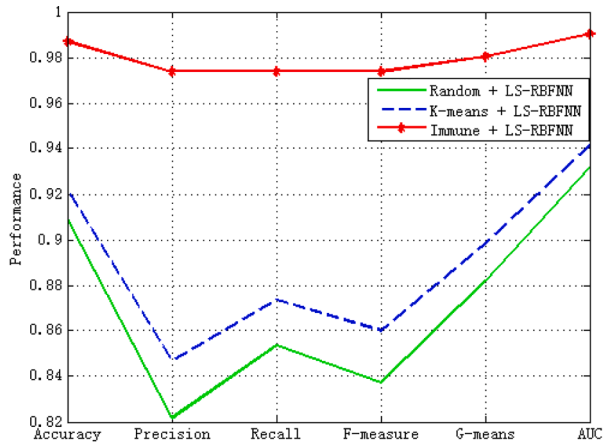
Poly kernel function has a good global generalization ability. The constant t and order d parameters have an impact on the generalization ability of the RBFNN model. In order to find the optimal parameter combination, a grid search procedure is used to determine the Poly's hyperparameters. Fig. 11 shows all the possible combinations, which indicates that the prediction accuracy of RBFNN is the best when the value of t is 8~8.4 and d is 2.

Table 4 compares the prediction results between our proposed kernel function and the Gaussian, Cosine, Manual (Aftab et al., 2014), Adaptive (Khan et al., 2017), Robust (Atif et al., 2020) kernel functions. The optimal values obtained by the proposed kernel function are marked in bold. The intuitive results are shown in Fig. 12. We use K-means to select the radial basis centers, and LS is used to update the biases and connection weights. Fig. 12 illustrates that our proposed hybrid kernel function outperforms other selected state-of-the-art kernel functions. Compared with the other kernel functions, the proposed kernel function shows higher prediction performance and less running time.

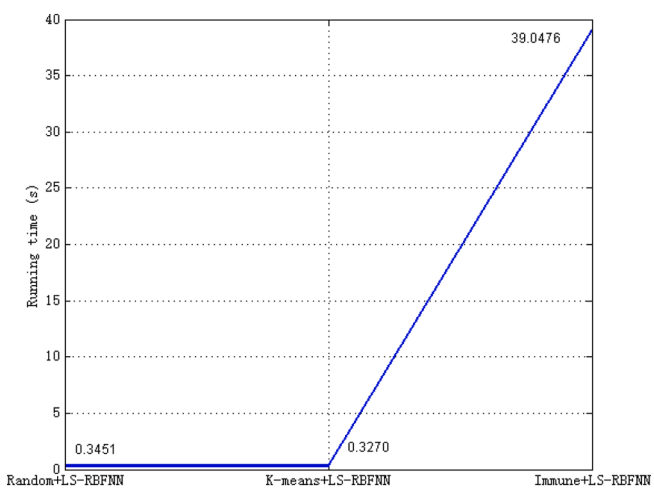
In order to verify the effectiveness of our proposed parameter optimization method, three different parameter update rules (centers: random, K-means, immune; biases and weights: LS) are chosen to identify the best combination based on the proposed kernel. The relevant parameters of IA are set as follows: the dimension of each individual is 64, the number of antibodies is 20, the number of selected individuals is 10, the clone size of each selected individual is 10, the mutation probability is 0.7, the incentive coefficient is 1, the similarity threshold is 0.2, and the maximum iteration number is 100. Table 5 illustrates the prediction results. The optimal values of all indexes are marked in bold. Fig. 13 plots the evaluation metrics of the three parameter update methods, which shows that the ILS-RBFNN model has the significant advantages compared with other combination strategies in Accuracy, Precision, Recall, F-measure, G-means, and AUC evaluation metrics, but the running time is increased. This is because the IA is used

Table 5
Evaluation metrics of the three parameter update methods.

Parameter update method	Accuracy	Precision	Recall	F-measure	G-means	AUC	Running time
Random + LS-RBFNN	0.9091	0.8217	0.8537	0.8374	0.8818	0.9318	0.3451
K-means + LS-RBFNN	0.9221	0.8469	0.8737	0.8601	0.8985	0.9416	**0.3270**
Immune + LS-RBFNN	**0.9870**	**0.9737**	**0.9737**	**0.9737**	**0.9805**	**0.9903**	39.0476



(a) Accuracy, Precision, Recall, F-measure, G-means, and AUC.



(b) Running time.

Fig. 13. Evaluation metrics of the three parameter update methods.

to optimize the centers of RBFNN, which increases the computational cost. Although the ILS-RBFNN algorithm has a longer running time than that of Random + LS-RBFNN and K-means + LS-RBFNN, its running time is acceptable in this case. ILS-RBFNN can effectively overcome the dependence of K-means on the initial centers. Hence, the algorithm combining proposed kernel function and parameter update method can create an effective RBFNN model, which can be used as the final classifier to predict the customer consumption behavior.

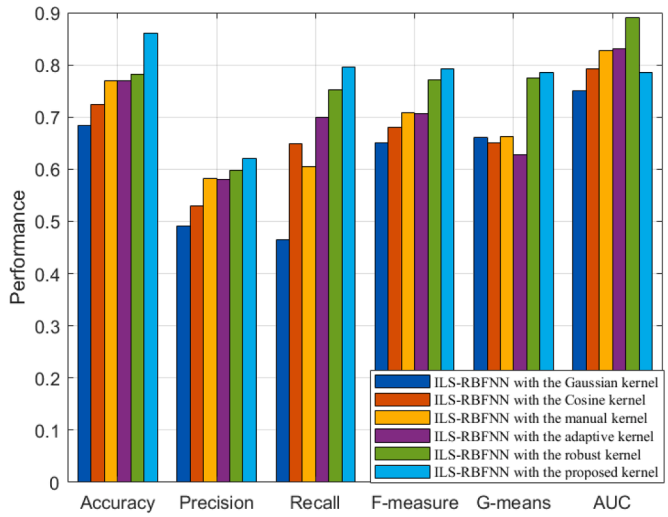
Based on the above experimental results, it is proved that the proposed kernel function and ILS-RBFNN classifier are superior to the compared methods. To verify the effectiveness of the kernel function proposed in the present study, the prediction results of six kernel functions before and after oversampling (Tomek-CASUWO) are compared based on the ILS-RBFNN model. The statistical results are shown in Table 6. The bold values indicate the optimal average metrics for all methods before and after oversampling. Fig. 14 shows the classifier performance for each kernel before and after oversampling based on the real-world dataset.

From Table 6 and Fig. 14, it can be determined that, for both before and after oversampling, the prediction performance of the ILS-RBFNN with the proposed kernel is significantly better than that of classifiers with different kernel functions. At the same time, the overall performance of the classifier for each kernel is also significantly improved after oversampling. Based on the Tomek-CASUWO oversampling, since the sample size of the real dataset is smaller after oversampling, the overall average running time of the classifier after oversampling is better. In contrast, the running times of the ILS-RBFNN with the robust and proposed kernels are relatively long. On the other hand, the classification error rate of the test data is used as an error measure. The training convergences of different kernels before and after oversampling are shown in Fig. 15, which shows that although the Gaussian and Cosine functions converge fast, the ILS-RBFNN model easily falls into a local minima. For our proposed kernel, after oversampling, the error rate is the lowest, and it can effectively converge to the global optimum. This demonstrates the feasibility of our proposed kernel function for predicting the consumption behavior of wine customers.

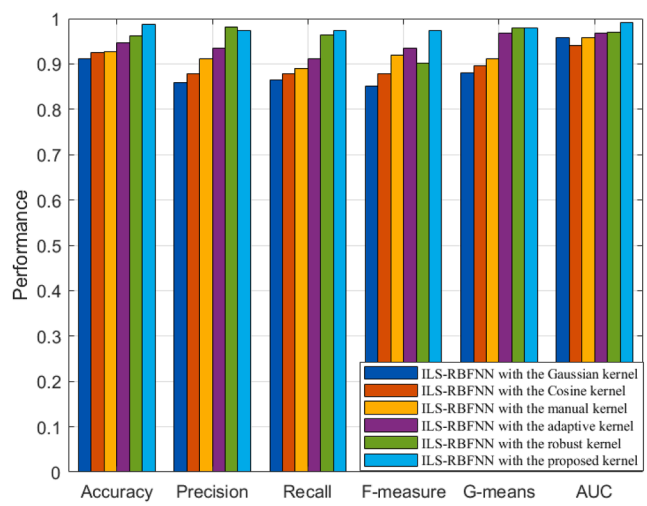
In order to verify the feasibility of the proposed ILS-RBFNN classifier, experiments are carried out by comparing the original RBFNN (Gaussian kernel + random and LS update parameters), LS-RBFNN with the robust kernel (robust kernel + K-means and LS update parameters), and our ILS-RBFNN with the proposed kernel (proposed kernel + IA and LS

Table 6
Classifier performance for each kernel before and after oversampling based on the real-world dataset.

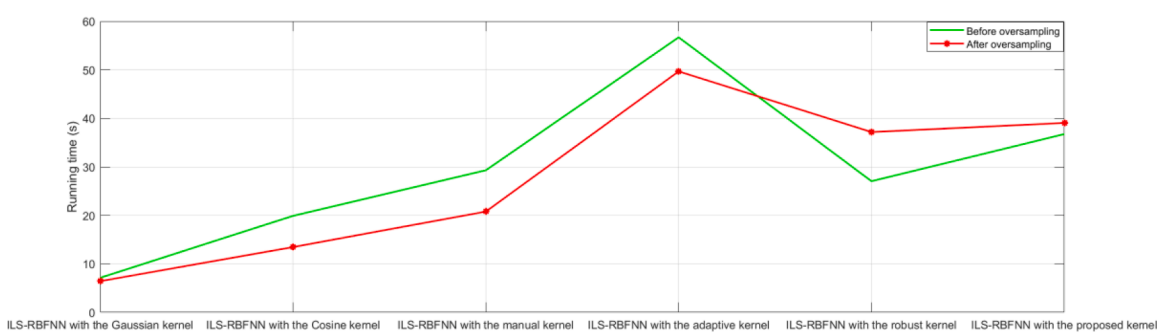
Oversampling	Classifier	Accuracy	Precision	Recall	F-measure	G-means	AUC	Running time
Before oversampling	ILS-RBFNN with the Gaussian kernel	0.6832	0.4911	0.4654	0.6510	0.6617	0.7500	7.1305
	ILS-RBFNN with the Cosine kernel	0.7241	0.5294	0.6493	0.6797	0.6507	0.7931	19.8955
	ILS-RBFNN with the manual kernel	0.7701	0.5826	0.6054	0.7090	0.6631	0.8276	29.3062
	ILS-RBFNN with the adaptive kernel	0.7701	0.5808	0.6997	0.7059	0.6284	0.8311	56.6782
	ILS-RBFNN with the robust kernel	0.7826	0.5987	0.7516	0.7714	0.7751	0.8896	27.0503
	ILS-RBFNN with the proposed kernel	0.8614	0.6212	0.7952	0.7930	0.7855	0.7845	31.7613
	Average value	0.7653	0.5673	0.6611	0.7183	0.6941	0.8127	28.6370
After oversampling	ILS-RBFNN with the Gaussian kernel	0.9122	0.8589	0.8642	0.8505	0.8812	0.9589	6.4315
	ILS-RBFNN with the Cosine kernel	0.9258	0.8790	0.8776	0.8792	0.8953	0.9403	13.4560
	ILS-RBFNN with the manual kernel	0.9269	0.9112	0.8903	0.9201	0.9114	0.9582	20.7827
	ILS-RBFNN with the adaptive kernel	0.9473	0.9348	0.9118	0.9348	0.9677	0.9673	49.6685
	ILS-RBFNN with the robust kernel	0.9625	0.9823	0.9637	0.9011	0.9803	0.9699	37.1763
	ILS-RBFNN with the proposed kernel	0.9870	0.9737	0.9737	0.9737	0.9805	0.9903	39.0476
	Average value	**0.9436**	**0.9233**	**0.9136**	**0.9099**	**0.9361**	**0.9642**	**27.7604**



(a) Before oversampling.

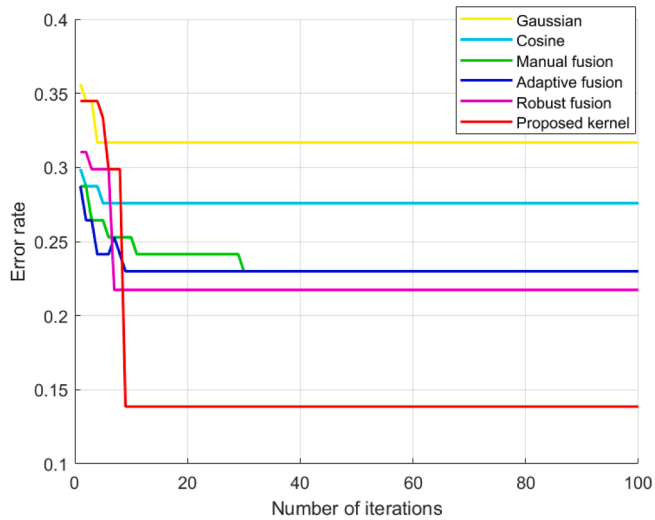


(b) After oversampling.

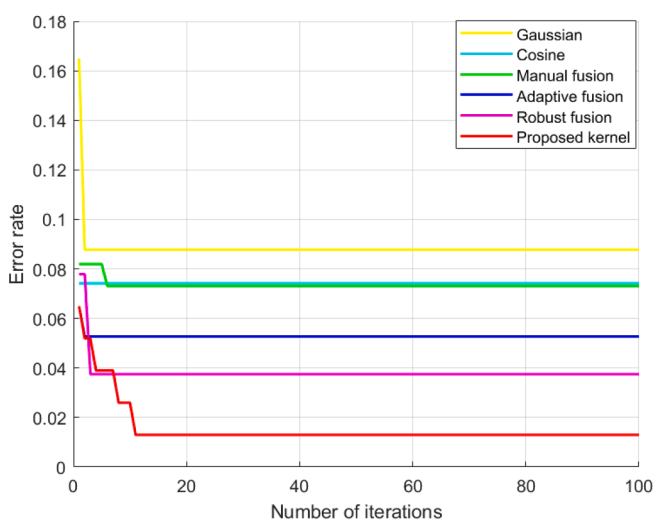


(c) Running time.

Fig. 14. Classifier performance for each kernel before and after oversampling based on the real-world dataset.



(a) Before oversampling.



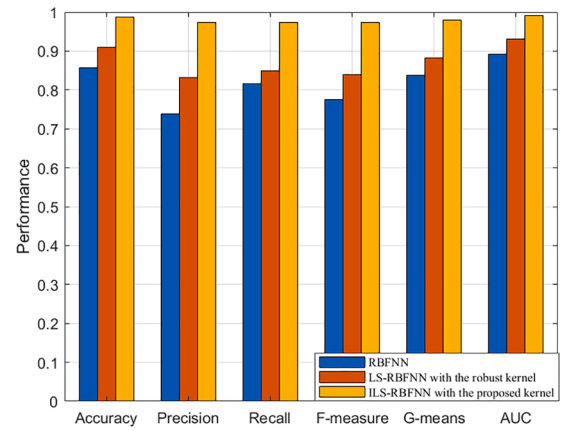
(b) After oversampling.

Fig. 15. Error convergence curves of different kernels before and after oversampling based on the real-world dataset.

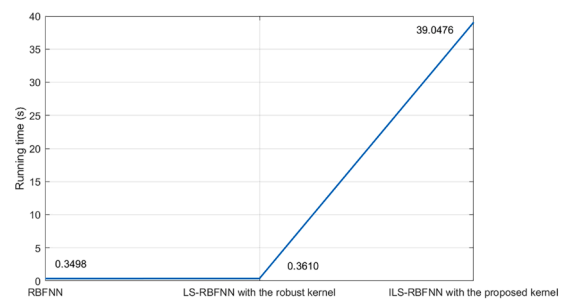
update parameters). Fig. 16(a) shows the comparison results, which shows that our ILS-RBFNN with the proposed kernel has the highest prediction performance. While the RBFNN and LS-RBFNN with the robust kernel have relatively low prediction performances. Also, the prediction performance of LS-RBFNN with the robust kernel is greater than that of RBFNN. Fig. 16(b) illustrates that the ILS-RBFNN with the proposed kernel model has a longer running time than that of RBFNN and LS-RBFNN with the robust kernel, but its running efficiency is acceptable in this paper. Hence, the application of IA and hybrid kernel can optimize the network parameters and enhance the learning ability of RBFNN.

6.4. Robustness analysis of the proposed methods

Two sets of experiments are conducted to verify the robustness and



(a) Accuracy, Precision, Recall, F-measure, G-means, and AUC.



(b) Running time.

Fig. 16. Prediction performances of the three target models.

application capabilities of the proposed two methods. The first experiment is used to verify the practical application of the proposed ILS-RBFNN model based on the extended real-world dataset, and the second experiment is used to evaluate the generalities of Tomek-CASUWO and ILS-RBFNN based on the UCI datasets.

6.4.1. Performance analysis of the improved ILS-RBFNN algorithm based on the extended real-world dataset

To verify the robustness of the proposed ILS-RBFNN algorithm, the real-world wine-customer behavior data was expanded from September to October 2020, and it shows the characteristics of high dimensionality and large sample data. The questionnaire is mainly divided into three parts: consumer behavior and values, consumer preferences, and basic customer information (Qi et al., 2021). The variables for consumer behavior and values include annual purchase frequency, annual drinking volume, product packaging, packaging capacity, purchasing motivations, production area, product brand, product source, and factors affecting consumption behavior. In the second part, the variables of consumer preferences include product color, sugar content, carbon dioxide content, taste, flavor, and acidity. The variables in the basic customer information include gender, age, marital status, occupation, monthly income, education, and regional economic development.

Some features are multiple-choice items, for example, the color of the wine is divided into red, white, and peach. Therefore, for the classification algorithms mentioned in this paper, there are 92 variable features involved in the modeling, with a total of 3621 records, and the number of each category is 754, 748, and 2119, respectively. After

Table 7
Evaluation metrics of the three parameter update methods.

Parameter update method	Accuracy	Precision	Recall	F-measure	G-means	AUC	Running time
RBFNN	0.5165	0.3813	0.4095	0.3949	0.5239	0.6374	**0.5749**
LS-RBFNN with the robust kernel	0.5827	0.4303	0.4473	0.4386	0.5620	0.6870	0.7912
ILS-RBFNN with the proposed kernel	**0.7697**	**0.6530**	**0.7534**	**0.6532**	**0.7264**	**0.7699**	152.4776

Table 8
Classifier performance for each kernel before and after oversampling based on the extended real-world dataset.

Oversampling	Classifier	Accuracy	Precision	Recall	F-measure	G-means	AUC	Running time
Before oversampling	ILS-RBFNN with the Gaussian kernel	0.4360	0.3931	0.4325	0.4564	0.4365	0.3459	123.2792
	ILS-RBFNN with the Cosine kernel	0.5349	0.3915	0.6592	0.4913	0.4709	0.4767	284.6102
	ILS-RBFNN with the manual kernel	0.6919	0.5501	0.5857	0.5674	0.6356	0.7169	334.7222
	ILS-RBFNN with the adaptive kernel	0.7209	0.5722	0.6468	0.6072	0.6665	0.7291	672.6815
	ILS-RBFNN with the robust kernel	0.7517	0.5861	0.6743	0.6491	0.6454	0.7847	359.1629
	ILS-RBFNN with the proposed kernel	0.7500	0.6259	0.6878	0.6554	0.6632	0.7864	100.9558
	Average value	0.6476	0.5198	**0.6144**	0.5711	0.5864	0.6400	**312.5686**
After oversampling	ILS-RBFNN with the Gaussian kernel	0.4338	0.4332	0.4235	0.4988	0.4857	0.5032	147.7424
	ILS-RBFNN with the Cosine kernel	0.6286	0.4610	0.4755	0.5468	0.5864	0.5215	443.2133
	ILS-RBFNN with the manual kernel	0.7313	0.5444	0.5948	0.5646	0.5792	0.7094	509.5826
	ILS-RBFNN with the adaptive kernel	0.7411	0.6435	0.6646	0.6449	0.6570	0.7301	1042.0002
	ILS-RBFNN with the robust kernel	0.7608	0.6443	0.6845	0.6448	0.7057	0.7858	550.9100
	ILS-RBFNN with the proposed kernel	0.7697	0.6530	0.7534	0.6532	0.7264	0.7699	152.4776
	Average value	**0.6776**	**0.5632**	0.5994	**0.5922**	**0.6234**	**0.6700**	474.3210

balancing the extended real-world dataset using the Tomek-CASUWO algorithm, the feature count before and after oversampling does not change. The final sample number is 6219, and the number of each category is 2054, 2065, and 2100, respectively. Table 7 shows the comparison results of the three parameter update methods. For the original RBFNN, the Gaussian kernel, random, and LS are implemented to update the parameters. For the LS-RBFNN with the robust kernel method, the robust kernel has been proposed in previous studies (Atif et al., 2020), and K-means and LS are used to update the parameters. In this study, an adaptive hybrid kernel based on Gaussian and Poly is proposed, which can improve the generalization ability of RBFNN (IA and LS are used to update parameters).

From Table 7, it can be seen that the prediction Accuracy, Precision, Recall, F-measure, G-means, and AUC of the ILS-RBFNN with the proposed kernel are the highest compared with RBFNN and LS-RBFNN with the robust kernel. Moreover, the running time of the ILS-RBFNN with the proposed kernel increases based on the high dimensionality and large sample dataset. Although the ILS-RBFNN with the proposed kernel has a longer running time than that of RBFNN and LS-RBFNN with the robust kernel, the research on this case is acceptable. Based on the above experimental results, it proves that the customer behavior prediction method proposed in this paper is practical and feasible using the extended real-world dataset.

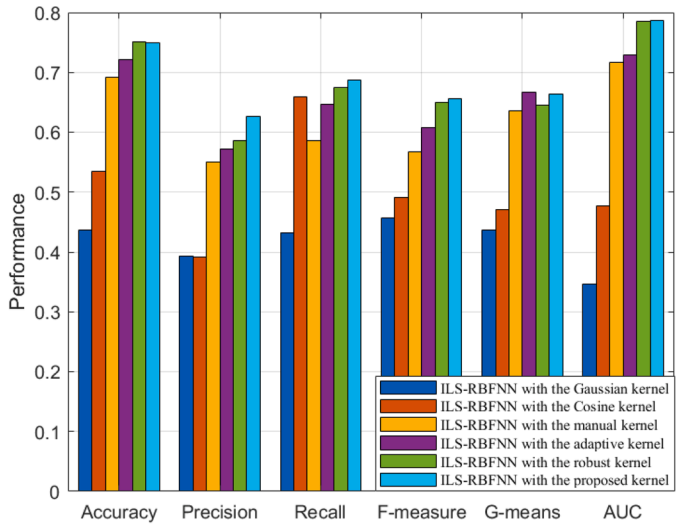
In addition, for the extended real-world dataset with a large sample size and high dimensionality, Table 8 shows the classifier performance for each kernel before and after oversampling. The visual effects of the results in Table 8 are shown in Fig. 17. From Table 8 and Fig. 17, similar conclusions can be drawn to those presented in Section 6.3.2 concerning the analysis of the classifier performance for each kernel before and after oversampling based on the real-world dataset. For the extended real-world dataset, the prediction performance of the ILS-RBFNN with the proposed kernel is greater than the ILS-RBFNN with existing kernels regardless of whether it is before or after oversampling. Furthermore, the overall performances of classifiers are better than those before oversampling. However, the running times of all classifiers are longer due to the increased number of samples after oversampling. Fig. 18 describes the error convergence curves of different kernels before and after oversampling based on the extended real-world dataset. The ILS-RBFNN with the proposed kernel has the optimal convergence fitness, signifying it has the best prediction accuracy and that the prediction accuracy is better than before oversampling.

6.4.2. Performance analysis of the proposed methods based on the UCI datasets

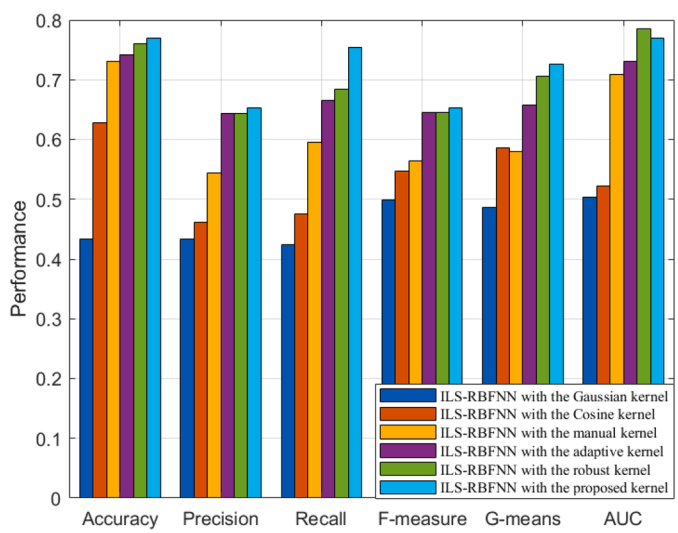
In this section, we compare the oversampling technique and proposed prediction model with the eleven common UCI datasets which have similar characteristics to the real-world dataset in this paper. Table 9 describes all the used UCI datasets, along with important characteristics, where #1∼#6 belong to the low-dimensional feature samples. We also select the multi-dimensional "Ultrasonic flowmeter diagnostics" and "Dermatology" datasets (Gyamfi et al., 2018). The flowmeter dataset is represented in terms of the following four feature sets (files): Meter A contains 87 instances and 36 attributes of diagnostic parameters; Meter B contains 92 instances and 51 attributes of diagnostic parameters; Meter C contains 181 instances and 43 attributes of diagnostic parameters; Meter D contains 180 instances and 43 attributes of diagnostic parameters. Meanwhile, the Dermatology dataset contains 33 attributes, and family history is used as an attribute to be classified.

Table 10 shows the prediction results. Each row represents the metric measures acquired from the three oversampling algorithms, and each column indicates the metric measures obtained from the three classification algorithms. Fig. 19 shows the Accuracy, Precision, Recall, F-measure, G-means, and AUC evaluation results of each dataset for the different oversampling and classification methods. Fig. 20 visualizes the running times of the three classification methods under different oversampling techniques.

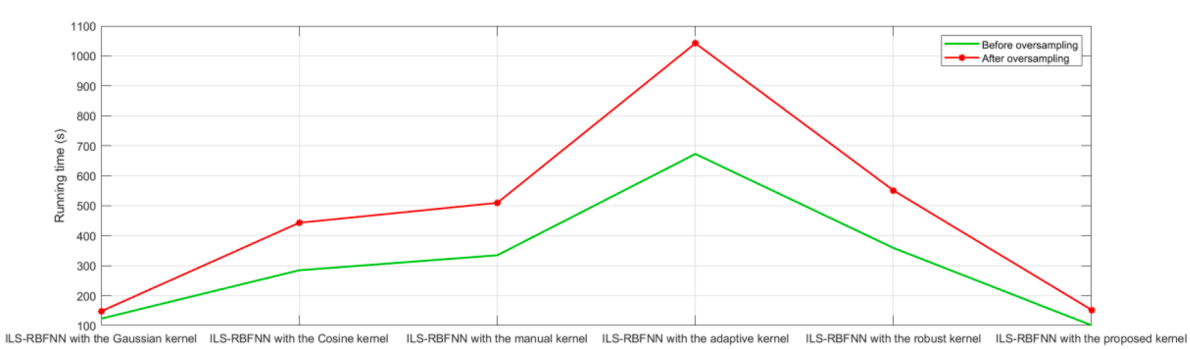
Hence, (1) regarding the target oversampling technologies, we compare the proposed Tomek-CASUWO algorithm with the mentioned



(a) Before oversampling.

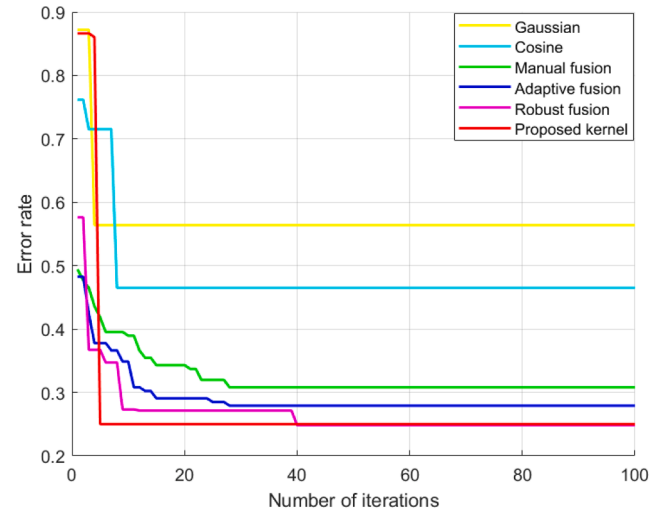


(b) After oversampling.

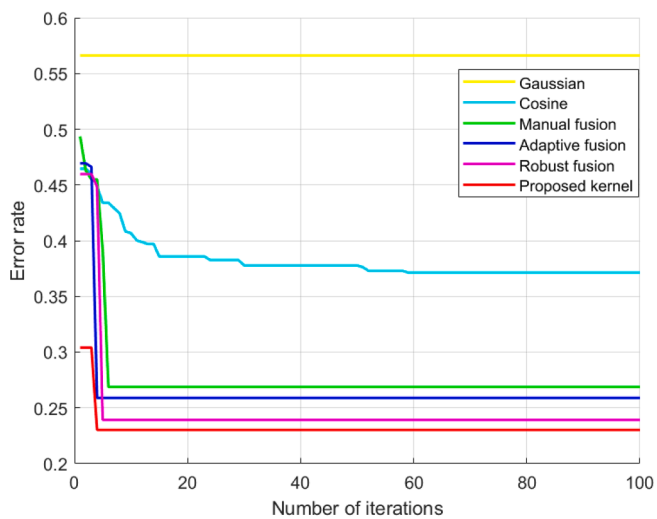


(c) Running time.

Fig. 17. Classifier performance for each kernel before and after oversampling based on the extended real-world dataset.



(a) Before oversampling.



(b) After oversampling.

Fig. 18. Error convergence curves of different kernels before and after oversampling based on the extended real-world dataset.

SMOTE and ASUWO methods. For the low-dimensional UCI datasets, after Tomek-CASUWO oversampling, the prediction performance of each classifier is significantly improved. However, for the multi-dimensional UCI datasets, the prediction performance of each classifier improves slightly. In a word, our proposed Tomek-CASUWO has better generalization ability than other methods based on each classification method; (2) as to the target classification methods, we compare our proposed ILS-RBFNN model, the RBFNN, and the LS-RBFNN with the robust kernel. The curve of each prediction method is plotted by considering all indicators of the three oversampling techniques. It is clear that the prediction accuracy of our proposed ILS-RBFNN model is the highest among the three target methods no matter what the oversampling models are; (3) for the running times of the three classifiers, the ILS-RBFNN with the proposed kernel has a longer running time than that of RBFNN and LS-RBFNN with the robust kernel. The main reason is that the radial basis centers are optimized by IA, which increases the running time of the proposed ILS-RBFNN.

The prediction results show that the Tomek-CASUWO is better than the other tested methods. Tomek-CASUWO can improve the prediction ability on the imbalanced datasets. Also, our ILS-RBFNN with the proposed kernel model achieves the highest prediction performance. The UCI numerical experiments demonstrate the feasibility and generalization abilities of our methods.

7. Practical implications

In the actual application environment, to solve the problem of imbalanced sample data categories, the Tomek links algorithm is used to remove the noise samples, and ASUWO is used to modify the oversampling formula. To improve the prediction accuracy of RBFNN, a hybrid kernel is designed to solve the overfitting problem. IA is used to optimize the centers of RBFNN, and LS is used to calculate the biases and connection weights. Finally, the Tomek-CASUWO and ILS-RBFNN with the hybrid kernel are applied to the prediction of consumer consumption behavior. Through the simulation experiments of the proposed two methods, the results show that the Tomek-CASUWO can significantly improve the prediction performances of the classifiers compared with the traditional oversampling algorithms. Furthermore, the ILS-RBFNN with the hybrid kernel improves the prediction accuracy compared with other selected state-of-the-art methods. We propose the theoretical analysis mechanism and modeling method suitable for the customer consumption behavior prediction.

To establish a long-term and pleasant supply-demand relationship with customers, it is necessary to establish an accurate prediction model to predict customer consumption behavior (Kytö et al., 2019; Sivasankar & Vijaya, 2019). The main practical implications of this study include:

Table 9
Summary description of UCI datasets.

#	Dataset	Number of samples	Number of features	Number of categories	Data distribution	Imbalance ratio	Data distribution after SMOTE	Data distribution after ASUWO	Data distribution after Tomek-CASUWO
1	Haberman	306	3	2	225/81	2.78:1	450 (225/225)	375 (189/186)	318 (160/158)
2	Bupa	345	6	2	145/200	1:1.38	400 (200/200)	263 (130/133)	213 (104/109)
3	Wisconsin	683	9	2	444/239	1.86:1	888 (444/444)	847 (432/415)	845 (429/416)
4	Newthyroid	215	5	3	150/35/30	5:1.17:1	450 (150/150/150)	446 (150/148/148)	426 (143/141/142)
5	Balance	625	4	3	288/49/288	5.88:1:5.88	864 (288/288/288)	863 (288/287/288)	752 (254/251/247)
6	Wine	178	13	3	59/71/48	1.23:1.48:1	213 (71/71/71)	207 (69/71/67)	144 (49/48/47)
7	Meter A	87	36	2	35/52	1:1.49	104 (52/52)	82 (40/42)	79 (39/40)
8	Meter B	92	51	3	19/24/49	1:1.26:2.58	147 (49/49/49)	143 (48/46/49)	143 (48/46/49)
9	Meter C	181	43	4	54/23/54/50	2.35:1:2.35:2.17	216 (54/54/54/54)	211 (54/53/54/50)	211 (54/53/54/50)
10	Meter D	180	43	4	51/23/55/51	2.22:1:2.39:2.22	220 (55/55/55/55)	204 (51/47/55/51)	210 (51/53/55/51)
11	Dermatology	366	33	2	320/46	6.96:1	640 (320/320)	581 (291/290)	542 (271/271)

Table 10
Prediction performances of the three target oversampling and classification methods based on the UCI datasets.

	SMOTE							ASUWO							Tomek-CASUWO						
#	Index1	Index2	Index3	Index4	Index5	Index6	Index7	Index1	Index2	Index3	Index4	Index5	Index6	Index7	Index1	Index2	Index3	Index4	Index5	Index6	Index7
RBFNN																					
1	0.587	0.493	0.518	0.505	0.505	0.587	0.027	0.584	0.543	0.550	0.547	0.566	0.584	0.023	0.806	0.767	0.738	0.722	0.725	0.806	0.023
2	0.425	0.405	0.429	0.436	0.423	0.525	0.020	0.554	0.508	0.590	0.546	0.528	0.654	0.022	0.673	0.613	0.618	0.615	0.631	0.672	0.021
3	0.759	0.747	0.747	0.747	0.757	0.759	0.020	0.777	0.800	0.789	0.786	0.785	0.807	0.022	0.800	0.815	0.810	0.822	0.844	0.860	0.023
4	0.667	0.622	0.629	0.625	0.659	0.700	0.026	0.633	0.639	0.700	0.668	0.675	0.800	0.023	0.833	0.696	0.821	0.753	0.827	0.875	0.022
5	0.636	0.583	0.598	0.590	0.689	0.702	0.023	0.690	0.615	0.651	0.633	0.643	0.767	0.021	0.827	0.685	0.697	0.691	0.768	0.870	0.021
6	0.564	0.516	0.524	0.520	0.573	0.576	0.021	0.567	0.561	0.587	0.574	0.687	0.750	0.021	0.700	0.733	0.719	0.771	0.822	0.820	0.021
7	0.423	0.333	0.333	0.333	0.408	0.431	0.029	0.512	0.423	0.469	0.365	0.344	0.403	0.021	0.524	0.438	0.617	0.456	0.421	0.431	0.023
8	0.512	0.423	0.469	0.365	0.344	0.403	0.025	0.546	0.523	0.561	0.504	0.469	0.527	0.024	0.574	0.552	0.577	0.503	0.525	0.540	0.022
9	0.650	0.481	0.625	0.544	0.716	0.767	0.021	0.700	0.441	0.750	0.556	0.748	0.800	0.021	0.650	0.380	0.675	0.487	0.691	0.767	0.023
10	0.750	0.519	0.675	0.587	0.750	0.833	0.021	0.700	0.507	0.900	0.649	0.831	0.800	0.023	0.806	0.741	0.786	0.785	0.856	0.842	0.024
11	0.672	0.545	0.545	0.545	0.552	0.672	0.022	0.828	0.701	0.714	0.707	0.710	0.828	0.025	0.824	0.645	0.698	0.752	0.743	0.869	0.026
LS-RBFNN with the robust kernel																					
1	0.652	0.525	0.542	0.533	0.536	0.652	0.032	0.632	0.571	0.602	0.551	0.618	0.632	0.028	0.875	0.762	0.775	0.769	0.774	0.875	0.027
2	0.500	0.457	0.463	0.460	0.469	0.600	0.029	0.592	0.535	0.590	0.561	0.557	0.692	0.025	0.727	0.711	0.702	0.691	0.684	0.727	0.023
3	0.839	0.821	0.821	0.821	0.833	0.859	0.027	0.858	0.856	0.856	0.856	0.857	0.858	0.024	0.851	0.882	0.892	0.887	0.888	0.941	0.029
4	0.756	0.706	0.718	0.712	0.736	0.767	0.026	0.749	0.758	0.803	0.780	0.841	0.917	0.024	0.924	0.868	0.900	0.850	0.899	0.943	0.025
5	0.693	0.648	0.715	0.680	0.762	0.775	0.025	0.805	0.675	0.699	0.687	0.765	0.853	0.027	0.880	0.776	0.779	0.778	0.834	0.910	0.026
6	0.792	0.741	0.735	0.758	0.768	0.754	0.028	0.805	0.788	0.825	0.806	0.864	0.889	0.025	0.833	0.839	0.839	0.839	0.884	0.890	0.026
7	0.507	0.414	0.453	0.409	0.438	0.500	0.031	0.500	0.375	0.500	0.429	0.433	0.500	0.026	0.625	0.429	0.750	0.546	0.433	0.625	0.026
8	0.552	0.466	0.458	0.430	0.421	0.442	0.032	0.566	0.573	0.580	0.542	0.531	0.557	0.029	0.586	0.577	0.589	0.583	0.581	0.563	0.028
9	0.750	0.577	0.750	0.652	0.800	0.733	0.033	0.650	0.418	0.700	0.523	0.720	0.767	0.031	0.762	0.546	0.763	0.785	0.768	0.806	0.032
10	0.766	0.593	0.695	0.658	0.767	0.833	0.028	0.650	0.423	0.750	0.541	0.738	0.733	0.030	0.810	0.761	0.793	0.787	0.890	0.873	0.029
11	0.734	0.596	0.602	0.599	0.604	0.734	0.032	0.879	0.780	0.781	0.780	0.784	0.879	0.027	0.926	0.850	0.869	0.859	0.860	0.926	0.029
ILS-RBFNN with the proposed kernel																					
1	0.804	0.663	0.696	0.679	0.678	0.804	4.523	0.768	0.758	0.760	0.759	0.765	0.868	3.732	0.954	0.938	0.941	0.946	0.949	0.983	3.708
2	0.662	0.607	0.616	0.612	0.615	0.721	4.387	0.746	0.698	0.769	0.732	0.730	0.846	3.149	0.909	0.818	0.818	0.818	0.826	0.909	2.286
3	0.929	0.927	0.937	0.937	0.937	0.949	5.466	0.957	0.951	0.955	0.953	0.954	0.977	5.181	0.988	0.976	0.977	0.976	0.977	0.988	5.315
4	0.888	0.860	0.880	0.881	0.883	0.883	3.954	0.916	0.876	0.918	0.912	0.936	0.967	3.941	0.987	0.957	0.987	0.967	0.984	0.985	3.870
5	0.843	0.788	0.795	0.792	0.820	0.857	4.647	0.897	0.798	0.809	0.803	0.854	0.922	4.658	0.987	0.972	0.973	0.973	0.980	0.990	4.570
6	0.914	0.872	0.896	0.917	0.901	0.929	2.491	0.947	0.931	0.953	0.940	0.966	0.961	2.971	0.954	0.967	0.951	0.949	0.969	0.965	2.734
7	0.697	0.651	0.701	0.665	0.506	0.589	3.573	0.762	0.743	0.737	0.698	0.654	0.710	2.708	0.875	0.762	0.804	0.711	0.693	0.736	2.695
8	0.587	0.543	0.526	0.497	0.465	0.496	3.258	0.609	0.673	0.669	0.684	0.690	0.758	3.240	0.632	0.676	0.693	0.680	0.625	0.657	3.225
9	0.825	0.763	0.782	0.701	0.822	0.796	4.196	0.912	0.773	0.850	0.810	0.891	0.933	4.664	0.900	0.781	0.800	0.790	0.867	0.874	4.392
10	0.790	0.679	0.747	0.789	0.793	0.897	3.372	0.900	0.800	0.800	0.800	0.870	0.933	3.442	0.952	0.872	0.911	0.891	0.937	0.936	3.523
11	0.906	0.819	0.833	0.826	0.827	0.906	7.895	0.944	0.839	0.869	0.920	0.902	0.896	7.657	0.965	0.854	0.895	0.943	0.896	0.917	7.251

(a) Haberman.

(b) Bupa.

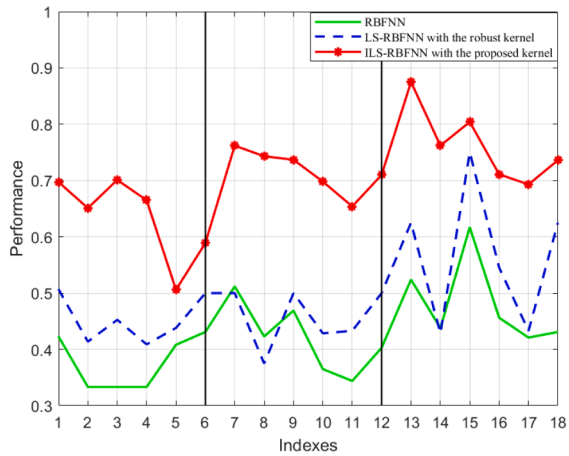
(c) Wisconsin.

(d) Newthyroid.

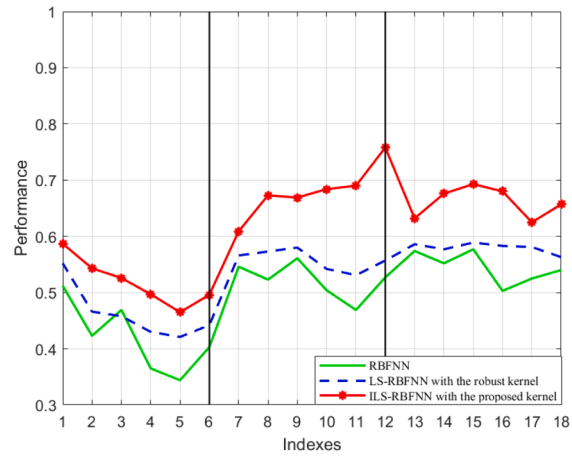
(e) Balance.

(f) Wine.

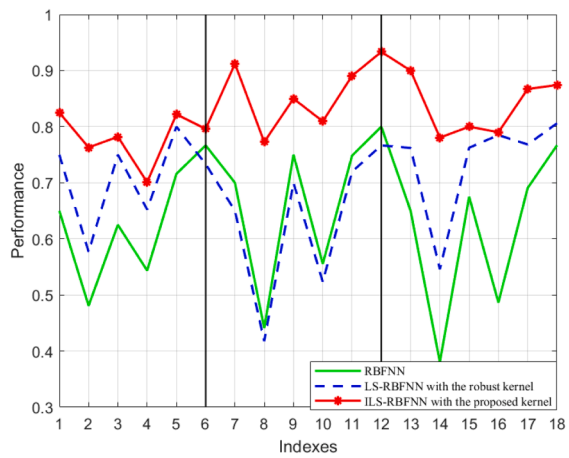
Fig. 19. Prediction performances of the three target oversampling and classification methods based on the UCI datasets.



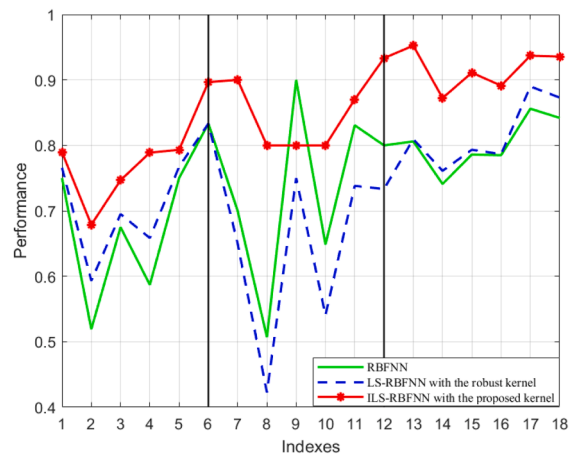
(g) Meter A.



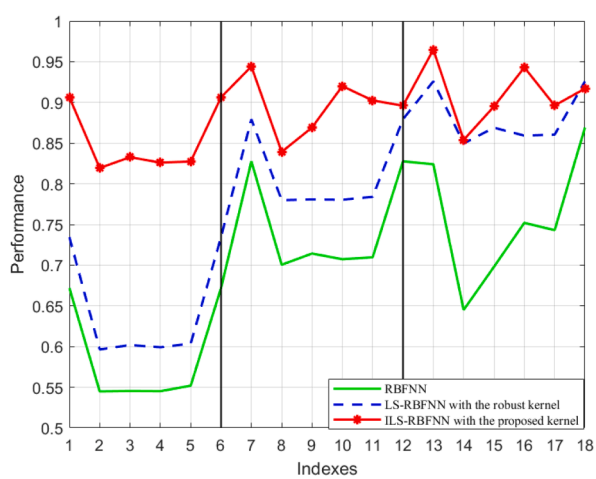
(h) Meter B.



(i) Meter C.



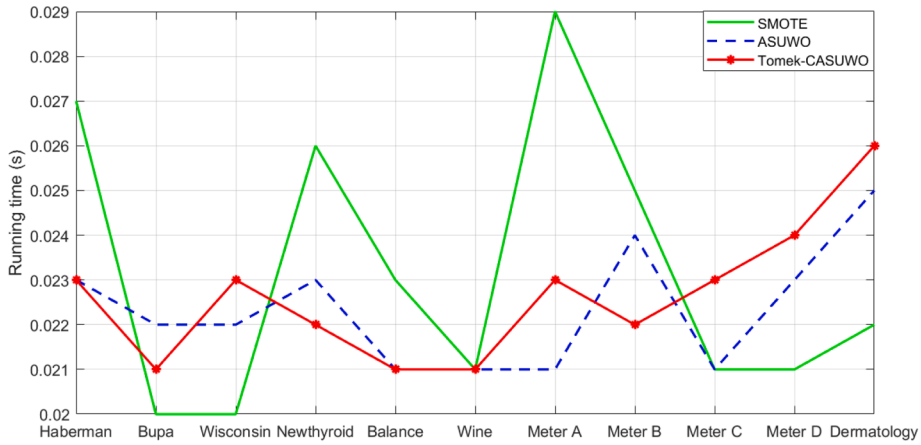
(j) Meter D.



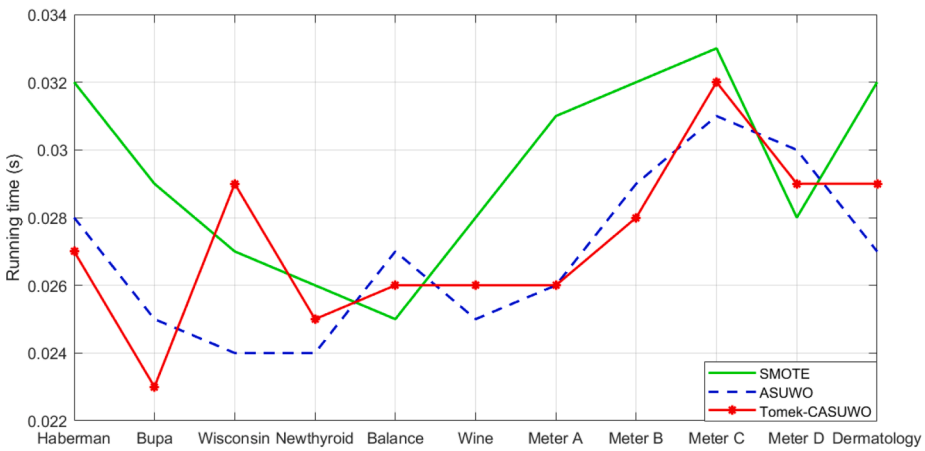
(k) Dermatology.

Fig. 19. (*continued*).

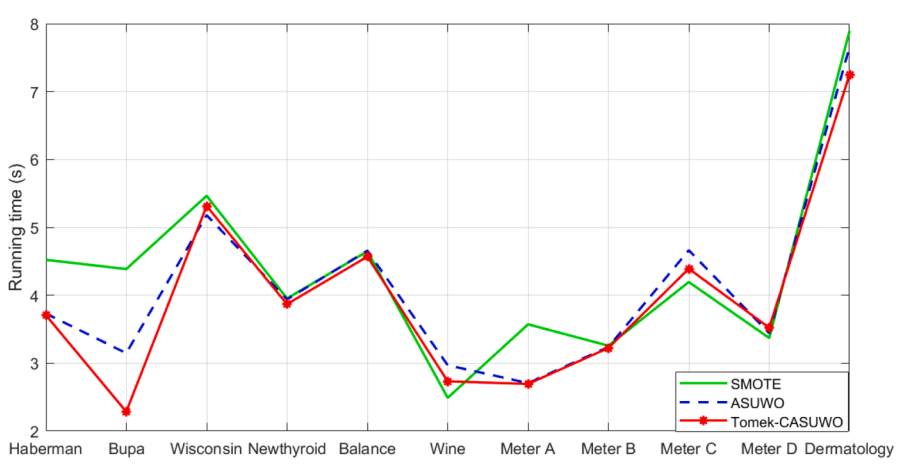
(1) Reduce the marketing costs of enterprises. Enterprises can quickly formulate targeted marketing strategies and tactics to improve the demand matching between products and customers. (2) Improve customers' satisfaction. Enterprises can accurately recommend personalized products and services for the individual needs of target customers. (3) Provide suitable products for target customer groups. The



(a) RBFNN.

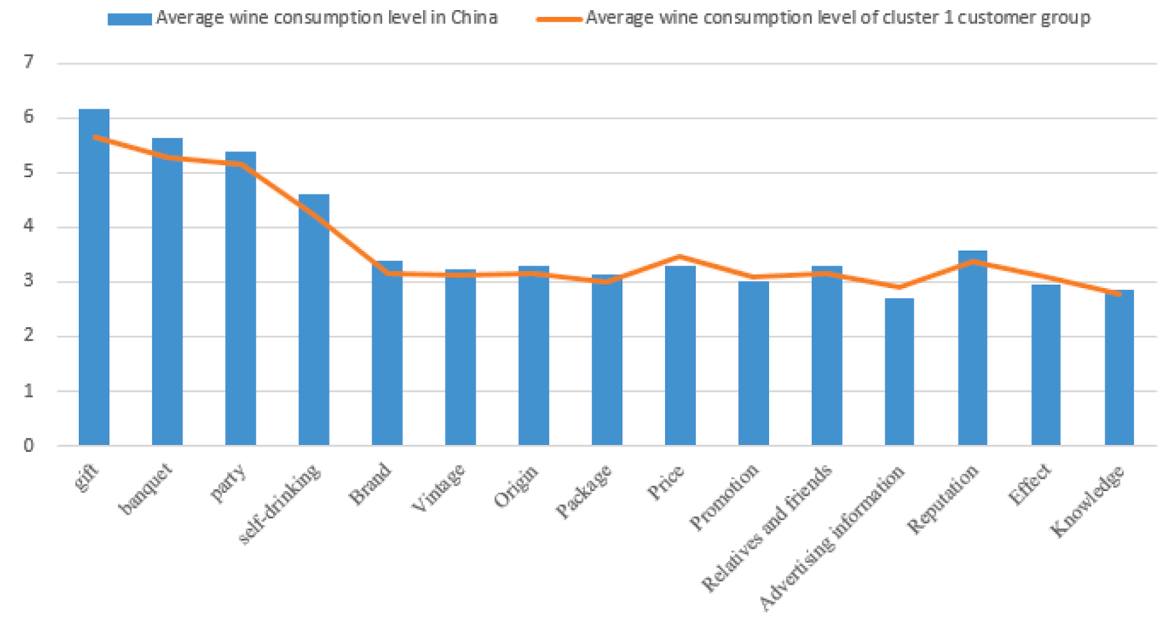


(b) LS-RBFNN with the robust kernel.

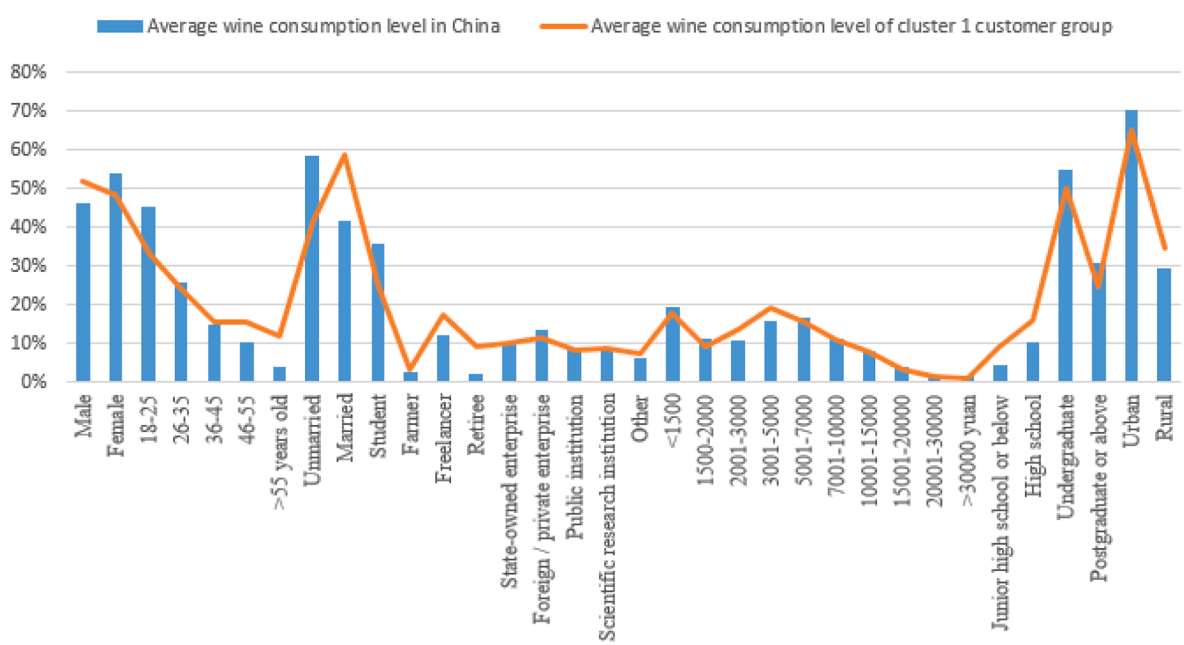


(c) ILS-RBFNN with the proposed kernel.

Fig. 20. Running times of the three classification methods under different oversampling techniques.



(a) Purchasing motivations and factors affecting consumption behavior.



(b) Basic customer characteristics.

Fig. 21. Comparison of consumption level between cluster 1 customer group and wine customers in China.

implementation of accurate customer consumption behavior prediction is of great significance to adjust the production structure of products.

We illustrate some insights that can be gain from prediction and segmentation, and that can be used at the management level. In this section, we further study the consumption characteristics of segmented customer groups based on the real-world wine-customer behavior data from June to October 2020. To facilitate the display and description of consumption characteristics, we take the purchasing motivations and the factors affecting consumption behavior of the cluster 1 customer group as an example. Fig. 21 shows the segmentation results between cluster 1 customer group and wine customers in China.

The prediction results can distinguish customer groups with similar consumption behavior habits and preferences. Fig. 21 illustrates the following main conclusions: (1) For the different purchasing motivations, customers' choices of wine prices show obvious differences. The prices for gift, banquet, party, and self-drinking for the cluster 1 customer group are significantly lower than the average consumption level of the wine market in China. (2) Different customer groups are affected by various factors affecting consumption behavior, and the influence degrees of consumption attributes are different. The affecting

factors containing wine price, promotion, advertising information, and effect have a higher level of consumption than the wine market in China. This is because the cluster 1 customer group is a rural middle-aged and elderly customer group with relatively low income, and most of them are freelancers and retirees. This paper offers a new perspective on the prediction of customer consumption behavior in China. The results significantly assist enterprises to carry out marketing activities more effectively, so this study has certain empirical significance.

8. Conclusions and further research

In the current business environment, predicting customer consumption behavior is of great importance for enterprises to provide personalized services. In this paper, we propose a new classifier using the proposed Tomek-CASUWO and ILS-RBFNN for customer consumption behavior prediction. This paper has two key innovations: (1) a new Tomek-CASUWO oversampling algorithm is proposed to solve the problems of noise samples and fuzzy class boundaries. This method can significantly improve the prediction performance of a classifier; (2) a new ILS-RBFNN classifier with the hybrid kernel is proposed to further improve the accuracy of prediction of customer consumption behavior. By designing the kernel function and introducing the IA to optimize the parameters of RBFNN, the prediction accuracy of the RBFNN model is improved, and premature convergence is effectively restricted.

We use wine-customer behavior data to test the feasibility and effectiveness of our Tomek-CASUWO algorithm and the ILS-RBFNN prediction model. The experimental results verify that the prediction performance of Tomek-CASUWO is superior to ASUWO based on LR, DT, RF, NB, KNN, SVM, BPNN, and RBFNN classifiers. To estimate the prediction accuracy of the ILS-RBFNN with the hybrid kernel, we compare it with several well-known kernel functions and parameter update methods. The experimental results illustrate that our proposed ILS-RBFNN method has a smaller prediction error compared with other methods. In addition, the sample dimensions and sample size of the original wine-consumer behavior data are supplemented. Experiments demonstrate the strong robustness and application capabilities of the proposed methods in predicting wine-customer consumption behavior. We also apply our methods to eleven typical UCI datasets and compare our Tomek-CASUWO and ILS-RBFNN with three target methods. The simulation results show that our proposed methods have better prediction performances than other target methods. They also suggest the potential applications of our methods in predicting the consumption behavior of other products with similar data characteristics.

Due to conditional constraints, there are still some limitations in the present research. The proposed Tomek-CASUWO is an optimization based on the ASUWO algorithm for problems existing in practical applications. The research on oversampling methods is still a bottleneck. Moreover, the large sample size and high dimensionality are the main challenges for predicting customer consumption behavior. In future research, we will further optimize, compare, and discuss the running time of the proposed ILS-RBFNN with the proposed kernel.

9. Data statement

Due to the sensitive nature of the questions asked in this study, survey respondents were assured raw data would remain confidential and would not be shared.

CRediT authorship contribution statement

Yue Li: Conceptualization, Methodology, Formal analysis, Writing – original draft. **Xiaoyun Jia:** Supervision, Writing – review & editing. **Ruili Wang:** Supervision, Writing – review & editing. **Jianfang Qi:** Software. **Haibin Jin:** Resources. **Xiaoquan Chu:** Visualization. **Weisong Mu:** Funding acquisition, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This work was supported by the Chinese Agricultural Research System [Grant No. CARS-29]; and the open funds of the Key Laboratory of Viticulture and Enology, Ministry of Agriculture, PR China.

References

Aftab, W., Moinuddin, M., & Shaikh, M. S. (2014). A novel kernel for RBF based neural networks. *Abstract and Applied Analysis, 2014*, 1–10. https://doi.org/10.1155/2014/176253

Ahmadi, M., Ghoushchi, S. J., Taghizadeh, R., & Sharifi, A. (2019). Presentation of a new hybrid approach for forecasting economic growth using artificial intelligence approaches. *Neural Computing and Applications, 31*(12), 8661–8680. https://doi.org/10.1007/s00521-019-04417-0

Anyfantis, D., Karagiannopoulos, M., Kotsiantis, S., & Pintelas, P. (2007). In *IFIP The International Federation for Information ProcessingArtificial Intelligence and Innovations 2007: from Theory to Applications* (pp. 21–28). Boston, MA: Springer US.

Atif, S. M., Khan, S., Naseem, I., Togneri, R., & Bennamoun, M. (2020). Multi-kernel fusion for RBF neural networks. *Machine Learning.* In press. http://arxiv.org/abs/2007.02592.

Beyan, C., & Fisher, R. (2015). Classifying imbalanced data sets using similarity based hierarchical decomposition. *Pattern Recognition, 48*(5), 1653–1672. https://doi.org/10.1016/j.patcog.2014.10.032

Bucak, S. S., Jin, R., & Jain, A. K. (2014). Multiple kernel learning for visual object recognition: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 36*(7), 1354–1369. https://doi.org/10.1109/TPAMI.2013.212

Bunkhumpornpat, C., Sinapiromsaran, K., & Lursinsap, C. (2009). Safe-level-SMOTE: safe-level-synthetic minority over-sampling technique for handling the class imbalanced problem. In T. Theeramunkong, B. Kijsirikul, N. Cercone, & T. B. Ho (Eds.), *Advances in Knowledge Discovery and Data Mining - Pacific-Asia conference, (PAKDD 2009), Bangkok, Thailand, April 27-30. Lecture Notes in Computer Science* (Vol. 5476, pp. 475-482). Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-01307-2_43.

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research, 16*(1), 321–357. https://doi.org/10.1613/jair.953

Chiang, H.-S., Chen, M.-Y., & Huang, Y.-J. (2019). Wavelet-based EEG processing for epilepsy detection using fuzzy entropy and associative Petri net. *IEEE Access, 7*, 103255–103262. https://doi.org/10.1109/Access.628763910.1109/ACCESS.2019.2929266

Chiu, C. Y., Kuo, I. T., & Lin, C. H. (2009). Applying artificial immune system and ant algorithm in air-conditioner market segmentation. *Expert Systems with Applications, 36*(3), 4437–4442. https://doi.org/10.1016/j.eswa.2008.05.005

Chu, X., Li, Y., Xie, Y., Tian, D., & Mu, W. (2019). Regional difference analyzing and prediction model building for Chinese wine consumers' sensory preference. *British Food Journal, 122*(8), 2587–2602. https://doi.org/10.1108/BFJ-06-2019-0465

Dey, P., Gopal, M., Pradhan, P., & Pal, T. (2019). On robustness of radial basis function network with input perturbation. *Neural Computing and Applications, 31*(2), 523–537. https://doi.org/10.1007/s00521-017-3086-5

Douzas, G., Bacao, F., & Last, F. (2018). Improving imbalanced learning through a heuristic oversampling method based on k-means and SMOTE. *Information Sciences, 465*, 1–20. https://doi.org/10.1016/j.ins.2018.06.056

Duma, M., & Twala, B. (2018). Optimising latent features using artificial immune system in collaborative filtering for recommender systems. *Applied Soft Computing, 71*(2), 183–198. https://doi.org/10.1016/j.asoc.2018.07.001

Duma, M., & Twala, B. (2019). Sparseness reduction in collaborative filtering using a nearest neighbour artificial immune system with genetic algorithms. *Expert Systems with Applications, 132*, 110–125. https://doi.org/10.1016/j.eswa.2019.04.034

Duvvuri, S. P., & Anmala, J. (2019). Fecal coliform predictive model using genetic algorithm-based radial basis function neural networks (GA-RBFNNs). *Neural Computing and Applications, 31*(12), 8393–8409. https://doi.org/10.1007/s00521-019-04520-2

Feng, H. M. (2006). Self-generation RBFNs using evolutional PSO learning. *Neurocomputing, 70*(1–3), 241–251. https://doi.org/10.1016/j.neucom.2006.03.007

Galar, M., Fernández, A., Barrenechea, E., Sola, H. B., & Herrera, F. (2012). A review on ensembles for the class imbalance problem: Bagging-, boosting-, and hybrid-based approaches. *IEEE Transactions on Systems Man and Cybernetics Part C (Applications and Reviews), 42*(4), 463–484. https://doi.org/10.1109/tsmcc.2011.2161285

Gao, X., Ren, B., Zhang, H., Sun, B., Li, J., Xu, J., … Li, K. (2020). An ensemble imbalanced classification method based on model dynamic selection driven by data partition hybrid sampling. *Expert Systems with Applications, 160*, 113660. https://doi.org/10.1016/j.eswa.2020.113660

García, S., Cano, J. R., Fernández, A., & Herrera, F. (2006). A proposal of evolutionary prototype selection for class imbalance problems. In E. Corchado, H. Yin, V. Botti, & C. Fyfe (Eds.), *Intelligent Data Engineering and Automated Learning - International*

conference, (IDEAL 2006), Burgos, Spain, September 20-23. Lecture Notes in Computer Science (Vol. 4224, pp. 1415-1423). Springer, Berlin, Heidelberg. https://doi.org/10.1007/11875581_168.

García, V., Sánchez, J. S., & Mollineda, R. A. (2012). On the effectiveness of preprocessing methods when dealing with different levels of class imbalance. *Knowledge-Based Systems, 25*(1), 13–21. https://doi.org/10.1016/j.knosys.2011.06.013

Gu, Y., Liu, T., Jia, X., Benediktsson, J. A., & Chanussot, J. (2016). Nonlinear multiple kernel learning with multiple-structure-element extended morphological profiles for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing, 54*(6), 3235–3247. https://doi.org/10.1109/TGRS.3610.1109/TGRS.2015.2514161

Gupta, D., Richhariya, B., & Borah, P. (2019). A fuzzy twin support vector machine based on information entropy for class imbalance learning. *Neural Computing and Applications, 31*(11), 7153–7164. https://doi.org/10.1007/s00521-018-3551-9

Gyamfi, K. S., Brusey, J., Hunt, A., & Gaura, E. (2018). Linear dimensionality reduction for classification via a sequential Bayes error minimisation with an application to flow meter diagnostics. *Expert Systems with Applications, 91*, 252–262. https://doi.org/10.1016/j.eswa.2017.09.010

Han, H., Wang, W. Y., & Mao, B. H. (2005). Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning. In D. S. Huang, X. P. Zhang, G. B. Huang (Eds.), *Advances in Intelligent Computing - International conference, (ICIC 2005), Hefei, China, August 23-26. Lecture Notes in Computer Science* (Vol. 3644, pp. 878-887). Springer, Berlin, Heidelberg. https://doi.org/10.1007/11538059_91.

Han, H. G., Lu, W., Hou, Y., & Qiao, J. F. (2018). An adaptive-PSO-based self-organizing RBF neural network. *IEEE Transactions on Neural Networks and Learning Systems, 29*(1), 104–117. https://doi.org/10.1109/TNNLS.2016.2616413

He, H., Bai, Y., Garcia, E. A., & Li, S. (2008). ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In *IEEE International Joint Conference on Neural Networks - IEEE World Congress on Computational Intelligence, Hong Kong*. https://doi.org/10.1109/IJCNN.2008.4633969

Huang, Y., & Kechadi, T. (2013). An effective hybrid learning system for telecommunication churn prediction. *Expert Systems with Applications, 40*(14), 5635–5647. https://doi.org/10.1016/j.eswa.2013.04.020

Hussain, S. F. (2019). A novel robust kernel for classifying high-dimensional data using Support Vector Machines. *Expert Systems with Applications, 131*, 116–131. https://doi.org/10.1016/j.eswa.2019.04.037

Jafarzadegan, M., Safi-Esfahani, F., & Beheshti, Z. (2019). Combining hierarchical clustering approaches using the PCA method. *Expert Systems with Applications, 137*, 1–10. https://doi.org/10.1016/j.eswa.2019.06.064

Jian, C., Gao, J., & Ao, Y. (2016). A new sampling method for classifying imbalanced data based on support vector machine ensemble. *Neurocomputing, 193*, 115–122. https://doi.org/10.1016/j.neucom.2016.02.006

Khan, F., Kanwal, S., Alamri, S., & Mumtaz, B. (2020). Hyper-parameter optimization of classifiers, using an artificial immune network and its application to software bug prediction. *IEEE Access, 8*, 20954–20964. https://doi.org/10.1109/Access.628763910.1109/ACCESS.2020.2968362

Khan, S., Naseem, I., Togneri, R., & Bennamoun, M. (2017). A novel adaptive kernel for the RBF neural networks. *Circuits, Systems, and Signal Processing, 36*(4), 1639–1653. https://doi.org/10.1007/s00034-016-0375-7

Kitayama, S., & Yamazaki, K. (2011). Simple estimate of the width in Gaussian kernel with adaptive scaling technique. *Applied Soft Computing, 11*(8), 4726–4737. https://doi.org/10.1016/j.asoc.2011.07.011

Koehn, D., Lessmann, S., & Schaal, M. (2020). Predicting online shopping behaviour from clickstream data using deep learning. *Expert Systems with Applications, 150*, 113342. https://doi.org/10.1016/j.eswa.2020.113342

Kruzlicova, D., Mocak, J., Balla, B., Petka, J., Farkova, M., & Haveld, J. (2009). Classification of Slovak white wines using artificial neural networks and discriminant techniques. *Food Chemistry, 112*(4), 1046–1052. https://doi.org/10.1016/j.foodchem.2008.06.047

Kytö, E., Virtanen, M., & Mustonen, S. (2019). From intention to action: Predicting purchase behavior with consumers' product expectations and perceptions, and their individual properties. *Food Quality and Preference, 75*, 1–9. https://doi.org/10.1016/j.foodqual.2019.02.002

Lee, W. I., Shih, B. Y., & Chung, Y. S. (2008). The exploration of consumers' behavior in choosing hospital by the application of neural network. *Expert Systems with Applications, 34*(2), 806–816. https://doi.org/10.1016/j.eswa.2006.10.020

Li, Y., Chu, X., Fu, Z., Feng, J., & Mu, W. (2019). Shelf life prediction model of postharvest table grape using optimized radial basis function (RBF) neural network. *British Food Journal, 121*(11), 2919–2936. https://doi.org/10.1108/BFJ-03-2019-0183

Li, Y., Chu, X., Tian, D., Feng, J., & Mu, W. (2021). Customer segmentation using K-means clustering and the adaptive particle swarm optimization algorithm. *Applied Soft Computing, 113*, 107924. https://doi.org/10.1016/j.asoc.2021.107924

Li, Q., & Mao, Y. (2014). A review of boosting methods for imbalanced data classification. *Pattern Analysis and Applications, 17*(4), 679–693. https://doi.org/10.1007/s10044-014-0392-8

Lin, W. C., Tsai, C. F., Hu, Y. H., & Jhang, J. S. (2017). Clustering-based undersampling in class-imbalanced data. *Information Sciences, 409-410*, 17–26. https://doi.org/10.1016/j.ins.2017.05.008

Lin, C. L., Wang, J. F., Chen, C. Y., Chen, C. W., & Yen, C. W. (2009). Improving the generalization performance of RBF neural networks using a linear regression technique. *Expert Systems with Applications, 36*(10), 12049–12053. https://doi.org/10.1016/j.eswa.2009.03.012

Liu, Y., Zhao, J., & Xiao, Y. (2018). C-RBFNN: A user retweet behavior prediction method for hotspot topics based on improved RBF neural network. *Neurocomputing, 275*, 733–746. https://doi.org/10.1016/j.neucom.2017.09.015

López, V., Fernández, A., Moreno-Torres, J. G., & Herrera, F. (2012). Analysis of preprocessing vs. cost-sensitive learning for imbalanced classification. Open problems on intrinsic data characteristics. *Expert Systems with Application, 39*(7), 6585–6608. https://doi.org/10.1016/j.eswa.2011.12.043

López-González, A., Meda Campaña, J. A., Hernández Martínez, E. G., & Contro, P. P. (2020). Multi robot distance based formation using parallel genetic algorithm. *Applied Soft Computing, 86*, 105929. https://doi.org/10.1016/j.asoc.2019.105929

Lu, H., Ma, X., Huang, K., & Azimi, M. (2020). Carbon trading volume and price forecasting in China using multiple machine learning models. *Journal of Cleaner Production, 249*, 119386. https://doi.org/10.1016/j.jclepro.2019.119386

Moinuddin, M., Naseem, I., Aftab, W., Bencherif, S. A., & Memich, A. (2017). A weighted cosine RBF neural networks. *Journal of Molecular Biology and Biotechnology, 2*(2), 1–8. http://www.imedpub.com/articles/a-weighted-cosine-rbf-neural-networks.pdf.

Monney, Augustine, Zhan, Yongzhao, Jiang, Zhen, & Benuwa, Ben-Bright (2020). A multi-kernel method of measuring adaptive similarity for spectral clustering. *Expert Systems with Applications, 159*, 113570. https://doi.org/10.1016/j.eswa.2020.113570

Muling, Tian, Muqin, Tian, Jieming, Yang, Li, Julie, Sundhararajan, Mahalingam, Gao, Xiao-Zhi, & Vahdat Nejad, Hamed (2018). Optimization of RBF neural network used in state recognition of coal flotation. *Journal of Intelligent and Fuzzy Systems, 34*(2), 1193–1204. https://doi.org/10.3233/JIFS-169414

Nadkarni, J., & Neves, R. F. (2018). Combining neuro evolution and principal component analysis to trade in the financial markets. *Expert Systems with Applications, 103*, 184–195. https://doi.org/10.1016/j.eswa.2018.03.012

Nekooeimehr, I., & Lai-Yuen, S. K. (2016). Adaptive semi-unsupervised weighted oversampling (A-SUWO) for imbalanced datasets. *Expert Systems with Applications, 46*, 405–416. https://doi.org/10.1016/j.eswa.2015.10.031

Oh, S. K., Kim, W. D., Pedrycz, W., & Joo, S. C. (2012). Design of K-means clustering-based polynomial radial basis function neural networks (pRBF NNs) realized with the aid of particle swarm optimization and differential evolution. *Neurocomputing, 78*(1), 121–132. https://doi.org/10.1016/j.neucom.2011.06.031

Peng, Y., Wang, Y., Niu, Y., & Hu, Q. (2014). Application study on intrusion detection system using IRBF. *Journal of Software, 9*(1), 177–183. https://doi.org/10.4304/jsw.9.1.177-183

Qi, J., Mou, X., Li, Y., Chu, X., & Mu, W. (2021). A novel consumer preference mining method based on improved weclat algorithm. *Journal of Enterprising Communities: People and Places in the Global Economy, 16*(1), 74–92. https://doi.org/10.1108/JEC-08-2021-0113

Ren, Siqi, Zhu, Wen, Liao, Bo, Li, Zeng, Wang, Peng, Li, Keqin, … Li, Zejun (2019). Selection-based resampling ensemble algorithm for nonstationary imbalanced stream data learning. *Knowledge-Based Systems, 163*, 705–722. https://doi.org/10.1016/j.knosys.2018.09.032

Rivera, W. A. (2017). Noise reduction a priori synthetic over-sampling for class imbalanced data sets. *Information Sciences, 408*, 146–161. https://doi.org/10.1016/j.ins.2017.04.046

Rubio, J. J. (2021). Stability analysis of the modified Levenberg-Marquardt algorithm for the artificial neural network training. *IEEE Transactions on Neural Networks and Learning Systems, 32*(8), 3510–3524. https://doi.org/10.1109/TNNLS.2020.3015200

Rubio, J. J., Islas, M. A., Ochoa, G., Cruz, D. R., Garcia, E., & Pacheco, J. (2022). Convergent newton method and neural network for the electric energy usage prediction. *Information Sciences, 585*, 89–112. https://doi.org/10.1016/j.ins.2021.11.038

Rubio, José de Jesús, Lughofer, Edwin, Pieper, Jeff, Cruz, Panuncio, Martinez, Dany Ivan, Ochoa, Genaro, … Garcia, Enrique (2021). Adapting H-infinity controller for the desired reference tracking of the sphere position in the maglev process. *Information Sciences, 569*, 669–686. https://doi.org/10.1016/j.ins.2021.05.018

Sáez, J. A., Krawczyk, B., & Woźniak, M. (2016). Analyzing the oversampling of different classes and types of examples in multi-class imbalanced datasets. *Pattern Recognition, 57*, 164–178. https://doi.org/10.1016/j.patcog.2016.03.012

Samigulina, G. A., & Samigulina, Z. I. (2019). Modified immune network algorithm based on the Random Forest approach for the complex objects control. *Artificial Intelligence Review, 52*(4), 2457–2473. https://doi.org/10.1007/s10462-018-9621-7

Shastry, K. A., Sanjay, H. A., & Deexith, G. (2017). Quadratic-radial-basis-function-kernel for classifying multi-class agricultural datasets with continuous attributes. *Applied Soft Computing, 58*, 65–74. https://doi.org/10.1016/j.asoc.2017.04.049

Shen, Feng, Zhao, Xingchao, Li, Zhiyong, Li, Ke, & Meng, Zhiyi (2019). A novel ensemble classification model based on neural networks and a classifier optimisation technique for imbalanced credit risk evaluation. *Physica A: Statistical Mechanics and its Applications, 526*, 121073. https://doi.org/10.1016/j.physa:2019.121073

Shilaskar, S., & Ghatol, A. (2019). Diagnosis system for imbalanced multi-minority medical dataset. *Soft Computing, 23*(13), 4789–4799. https://doi.org/10.1007/s00500-018-3133-x

Sivasankar, E., & Vijaya, J. (2019). Hybrid PPFCM-ANN model: An efficient system for customer churn prediction through probabilistic possibilistic fuzzy clustering and artificial neural network. *Neural Computing and Applications, 31*(11), 7181–7200. https://doi.org/10.1007/s00521-018-3548-4

Suh, E., Lim, S., Hwang, H., & Kim, S. (2004). A prediction model for the purchase probability of anonymous customers to support real time web marketing: A case study. *Expert Systems with Applications, 27*(2), 245–255. https://doi.org/10.1016/j.eswa.2004.01.008

Vargas, D. M. (2021). Superpixels extraction by an Intuitionistic fuzzy clustering algorithm. *Journal of Applied Research and Technology, 19*(2), 140–152. https://doi.org/10.22201/icat.24486736e.2021.19.2.1581.

Vijayaragavan, P., Ponnusamy, R., & Arramuthan, M. (2018). A study on consumer behavior predict in e-Commerce based on rough set. *Journal of Engineering and Applied Sciences, 13*(6), 1520–1522. https://doi.org/10.3923/jeasci.2018.1520.1522

Wang, H., Zhou, Z., Feng, J., Tian, D., & Mu, W. (2018). Influencing factors on Chinese wine consumers behavior under different purchasing motivations based on a multi-classification method. *Italian Journal of Food Science, 30*(4), 775–791. https://www.researchgate.net/publication/329246646.

Wei, Wenhong, Chen, Shuo, Lin, Qiuzhen, Ji, Junkai, & Chen, Jianyong (2020). A multi-objective immune algorithm for intrusion feature selection. *Applied Soft Computing, 95*, 106522. https://doi.org/10.1016/j.asoc.2020.106522

Wu, Q., Wang, X., & Shen, Q. (2016). Research on dynamic modeling and simulation of axial-flow pumping system based on RBF neural network. *Neurocomputing, 186*, 200–206. https://doi.org/10.1016/j.neucom.2015.12.064

Xu, Zhaozhao, Shen, Derong, Nie, Tiezheng, & Kou, Yue (2020). A hybrid sampling algorithm combining M-SMOTE and ENN based on Random Forest for medical imbalanced data. *Journal of Biomedical Informatics, 107*, 103465. https://doi.org/10.1016/j.jbi.2020.103465

Yang, Z., Mourshed, M., Liu, K., Xu, X., & Feng, S. (2020). A novel competitive swarm optimized RBF neural network model for short-term solar power generation forecasting. *Neurocomputing, 397*, 415–421. https://doi.org/10.1016/j.neucom.2019.09.110

Yu, Jing, Zhan, Jicheng, & Huang, Weidong (2017). Identification of wine according to grape variety using near-infrared spectroscopy based on radial basis function neural networks and least-squares support vector machines. *Food Analytical Methods, 10*(10), 3306–3311. https://doi.org/10.1007/s12161-017-0887-1

Yuan, J., Bo, L., Wang, K., & Yu, T. (2009). Adaptive spherical Gaussian kernel in sparse Bayesian learning framework for nonlinear regression. *Expert Systems with Applications, 36*(2), 3982–3989. https://doi.org/10.1016/j.eswa.2008.02.055

Zhang, Y., Dai, M., & Ju, Z. (2016). Preliminary discussion regarding SVM kernel function selection in the twofold rock slope prediction model. *Journal of Computing in Civil Engineering, 30*(3), 04015031. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000499

Zhang, J., Lu, Q., Li, G., & Wang, X. (2021). Analysis of factors influencing energy efficiency based on spatial quantile autoregression: evidence from the panel data in China. *Energies, 14*(2), 504. https://doi.org/10.3390/en14020504

Zhao, H., & Li, X. (2017). A cost sensitive decision tree algorithm based on weighted class distribution with batch deleting attribute mechanism. *Information Sciences, 378*, 303–316. https://doi.org/10.1016/j.ins.2016.09.054

Zounemat-Kermani, M., Stephan, D., Barjenbruch, M., & Hinkelmann, R. (2020). Ensemble data mining modeling in corrosion of concrete sewer: A comparative study of network-based (MLPNN & RBFNN) and tree-based (RF, CHAID, & CART) models. *Advanced Engineering Informatics, 43*, Article 101030. https://doi.org/10.1016/j.aei.2019.101030