Hindawi
Scientific Programming
Volume 2022, Article ID 3330196, 9 pages
https://doi.org/10.1155/2022/3330196

Research Article

A Sentiment Classification Model of E-Commerce User Comments Based on Improved Particle Swarm Optimization Algorithm and Support Vector Machines

Xuehui Jiang

College of Information Engineering, Zhengzhou University of Industrial Technology, Xinzheng 451150, China

Correspondence should be addressed to Xuehui Jiang; 002664@zzgyxy.edu.cn

Received 17 November 2021; Revised 21 January 2022; Accepted 25 January 2022; Published 1 April 2022

Academic Editor: Hangjun Che

With the rapid increase of the number of Internet users and the amount of online comment data, a large number of referable information samples are provided for data mining technology. As a technical application of data mining, text sentiment classification can be widely used in public opinion management, marketing, and other fields. In this study, a combination approach to SVM (support vector machine) and IPSO (improved particle swarm optimization) is proposed to classify sentiment by using text data. First, the text data of 30,000 goods reviews and corresponding ratings are collected through the web crawler. Then, TFIDF (term frequency-inverse document frequency) and Word2vec are used to vectorize the goods review text data. Next, the proposed classification model is trained by the SVM, and the initial parameters of the SVM are optimized by the IPSO. Finally, we applied the trained SVM-IPSO model to the test set and evaluated the performance by several measures. Our experiment results indicate that the proposed model performed the best for text data sentiment classification. Additionally, the traditional machine learning model SVM becomes very effective after parameter optimization, which demonstrates that the parameters' optimization by IPSO has successfully improved the classification accuracy. Furthermore, our proposed model SVM-IPSO significantly outperforms other benchmark models, indicating that it could be applied to improve the accuracy and efficiency for text data sentiment classification.

1. Introduction

Natural language processing (NLP) refers to the technology of processing human's unique natural language with computer as a tool, which is an important content in the field of artificial intelligence and computer science [1]. Emotional analysis is one of the directions of natural language processing, also known as emotional extraction or opinion mining, and it is a hot topic in text mining [2]. Text information mining can be applied to many aspects, such as user feedback, comment information, advertising recommendation after intelligent analysis, public opinion detection of government departments, and processing of uncivilized and untrue information.

Dave et al. designed the earliest emotion analysis tool [3]. Go et al. used the training set data designed by Hashtag to classify multiple topic-based clusters [4]. Joshi et al. designed a set of emotional analysis systems, which divided microblog comments into positive and negative emotions based on certain characteristics of the microblog, such as emoticons [5]. Gamon et al. used the function of clustering to obtain users' opinions and analyzed the tendency and intensity of emotional evaluation on cars [6]. Li et al. completed the emotional analysis by feature selection and extraction using SVM, Bayesian classification, and N-element language methods [7]. Li Guwei et al. constructed an unsupervised thematic emotion model to achieve emotion classification [8]. Jasson et al. applied CNN to text and made accurate predictions with the help of one-dimensional structure (word order) of text data [9]. Yoom Kim uses convolution to achieve a sentence-level classification model for English text. Xue et al. proposed a classification model based on convolutional neural network and gating mechanism [10]. Parupalli et al. constructed a corpus with systematic annotation [11], which supports the use of word-level annotation to enhance emotion analysis tasks. Angelidis et al. proposed an attention-based polarity scoring method for positive and negative text fragments [12]. Gui et al. proposed to extract text emotional reasons through further modeling of context information on the basis of emotion classification [13]. Yuan et al. used a global decoder feedforward network to realize the recognition of multilanguage text, laying a foundation for complex text analysis [14]. Bordoloi et al. designed an effective emotional analysis model, which carried out an advanced analysis on mobile phone comments collected by e-commerce sites based on the graph keyword extraction method [15]. Convolutional neural network (CNN) has been widely applied in many fields such as image recognition and NLP (for example, convolutional neural network [16] has been used to classify texts in social networks emotionally based on graph convolutional neural network [17]. Some scholars have used deep neural network for emotion classification and natural language processing [18]. Other studies have classified text emotions through attentional neural network [19]), and other fields [20]. However, CNNs contain multiple convolutional layers and pooling layers, which require more parameters and require a large cost of parameter optimization and adjustment. At the same time, the problem of gradient disappearance or gradient explosion exists in the CNNs, which limits the accuracy of text classification.

The classification of positive and negative emotions using support vector machine has a good execution efficiency, but the parameters of the support vector machine model are the key factor to determine the prediction accuracy. Therefore, the researchers introduce a particle swarm optimization algorithm (PSO) to optimize the parameters to obtain high accuracy. PSO is an important branch of intelligent optimization algorithm [21] and is proposed in 1995 [22]. The standard PSO algorithm has good performance in solving a variety of nonlinear optimization problems. It controls the entire iterative process with the help of individual optimization and group optimization and has fast convergence speed and high execution efficiency. However, in the late iteration, the diversity of individual particles in the population is small. If the position of the global optimal and local optimal is equal to that of particle, after a certain number of iterations, the algorithm may fall into local optimal, resulting in poor global performance. To improve the performance of PSO, the inertial weight introduced by Shi et al. in 1988 played a key role in improving the performance of PSO [23]. In this study, the adaptive decreasing inertia strategy and crossover operator are combined to improve the searching quality of particle swarm optimization algorithm, and the parameters of the SVM model are optimized with the improved algorithm to further improve the prediction accuracy. Compared with other emotion analysis models, the training speed of this model is faster and the prediction accuracy is better.

The main contribution about this study is (1) the text data of 30,000 goods reviews and corresponding ratings are collected through the web crawler and (2) TFIDF and Word2vec are used to vectorize the goods review text data. The research results demonstrate that the SVM-IPSO makes up for the deficiency of traditional emotional dictionary which is affected by word order and different contexts and solves the problem of local optimization. Our research avoids the problem of gradient disappearance or dispersion when using convolution, reduces the cost of parameter optimization and adjustment, has higher operation efficiency and accuracy, and can predict text emotion well. The main implication of our study is that the text sentiment classification model proposed by us can be widely used in public opinion management, marketing, and other fields. Help practitioners to provide detailed analysis and portraits of customer reviews.

2. Data Preparation

Before classification, data preparation including data acquisition and preprocessing is necessary.

2.1. Comment on Data Acquisition. Data acquisition can be divided into URL queue acquisition, related web page parsing, data crawling, data cleaning, and data storage.

First, the goods comments' data are extracted from Taobao e-commerce platform. Taobao e-commerce goods comments are a five-star rating mechanism, with one star to five stars: very poor, poor, ok, recommendation, and strong recommendation; in this experiment, one-star and two-star comments were regarded as negative comments, while four-star and five-star comments were regarded as positive comments. Then, based on Scrapy framework to achieve data capture, we generate the corresponding positive and negative emotional comments' csv file. Finally, through further screening, 21000 positive comments and negative comments were obtained 9000, of which 2/3 were used as the training set and 1/3 as the test set. The distribution of experimental data is listed in Table 1.

To obtain the datasets of sufficient scale, it is necessary to realize the simulated login in the crawling process and break through the limitation of web crawler. At the same time, a comment may be incomplete in the current page. Scrapy has a default de-duplication mechanism, which will determine that the second time is repeated crawling, so the problem of URL duplication needs to be solved.

2.2. Text preprocessing. Text preprocessing is to further process the original data to make the data into the next operable object. Word preprocessing consists of word repetition, noise, word segmentation, and word stopping.

(1) Text weight: there are not only repeated words in the preprocessed text, resulting in storage redundancy, but also an increased amount of calculation, so it is necessary to traverse in duplicate.
(2) Text denoising: some disturbing words or garbled characters appear in the text, which need to be denoised and improve the accuracy of analysis.
(3) Word segmentation: the words in the text are segmented and given corresponding parts of speech in

Table 1: Data distribution of experimental corpus.

	Number of comments	Positive comments	Negative comments
Training data	20000	14000	6000
Test data	10000	7000	3000

combination with the dictionary. In English, each word is separated by a space, so it is easy to process, while Chinese word segmentation has high requirements on the comprehensive accuracy of the dictionary, so this paper adopts jieba word segmentation in Python language environment.

(4) Remove stop words: stop words refer to modal particles, adverbs, prepositions, conjunctions, etc., which are not helpful to the experimental results and can be summarized into the stop words list and removed.

2.3. Word Vectorization. Text is unstructured or semi-structured data, which cannot be recognized by SVM classifier. Therefore, the text needs to be transformed into vector form for further analysis and processing. Lexicographical quantization refers to the expression of words in the vector form; at the same time, it is necessary to ensure the correlation of the processed vectors in semantic similarity and relative similarity. Word vectorization can map words or phrases into real number vectors and reduce the features of higher dimensional vector space to lower dimensional space. There are many models to transform words into a real number vectors, such as implicit Dirichlet location (LDA) and implicit semantic analysis (LSA). However, the calculation amount of the above model will increase sharply with the increase of the total amount of data, and word2vec solves this problem well and improves the efficiency.

Word2vec is a deep learning multilayer neural network structure opened by Google in 2013, which can be trained to simplify the processing of text content with K-dimensional vector operations. Its main structure consists of the following component: input layer, several hidden layers, and an output layer. It transforms all feature words into vector values after matching, to give a deeper feature representation of text data. Based on this, this study uses word2vec to realize word vectorization. Assume that the preprocessed text comment are composed of N words, as shown in Table 1, $d_N = w_1, w_2, \ldots, w_N$. First, Sogou news corpus is trained by the Skip_gram model in Word2vec; next, the word2vec vector of each word in d_N is calculated by the trained model; because word2vec ignored the importance of words in text comments, the TFIDF model was trained according to the goods review data to obtain the TFIDF weight of each word in the goods review and multiply it by the corresponding word2vec vector. Finally, the support vector machine can recognize the input, and the final vector of each goods review is D:

$$D = \frac{\left(\sum_{i=1}^{N} \text{word 2 vec}\left(w_i\right) \times tfidf_{w_i}\right)}{N}. \tag{1}$$

3. Methods

3.1. The SVM Classification. Supporting vector machine can construct a hyperplane to maximize the distance between positive and negative poles in the decision surface. The main idea is shown in Figure 1.

In Figure 1, four-pointed stars and circles, respectively, represent two types of samples, and SVM maximizes the straight-line distance of these two types of samples. Supposing $x_1 = \{x_i,\ i \in I_1\}$ and $x_2 = \{x_i,\ i \in I_2\}$ two training sets, the label of X_1 is +1 and the label of X_2 is −1:

$$\begin{aligned} m &= \left|I_1\right|, \\ k &= \left|I_2\right|, \\ l &= m + k, \\ X &= X_1 \bigcup X_2. \end{aligned} \tag{2}$$

When X_1 and X_2 are linearly separable, SVM separates these two types of sample points without error by constructing a linear classification hyperplane, and the classification hyperplane is

$$\begin{aligned} H&: \langle \omega \cdot x \rangle = \lambda, \\ &\quad X_2 = \{x_i,\ i \in I_2\}. \end{aligned} \tag{3}$$

SVM proposes an optimal plane classification under the condition of linear separability, requiring the model not only to distinguish data without error but also to maximize the classification gap. Linear discriminant functions in multidimensional space are generally expressed as

$$g(x) = wx + b. \tag{4}$$

The equation of the classification surface is ab, and the discriminant function is normalized so that the distance between the two kinds of sample books and the optimal plane is greater than or equal to one.

At this point, the sample closest to the classification plane satisfies $|g(x)| = 1$ so that the sample of equation (5) is the support vector:

$$y_i\left(w \cdot x_i + b\right) \geq 1,\ i = 1, 2, 3, \ldots, n. \tag{5}$$

Classification interval is $2/\ w$, and the minimum of $w^2/2$ is equivalent to the maximum interval, which satisfies $\phi(w) = w^2/2$; at the same time, the classification surface satisfying equation (5) is the optimal classification surface. However, the samples are not linearly separable, that is, SVM algorithm cannot run a solvable scheme. To this end, the set of slack variables is $\xi = (\xi_1, \xi_2, \ldots, \xi_n)$:

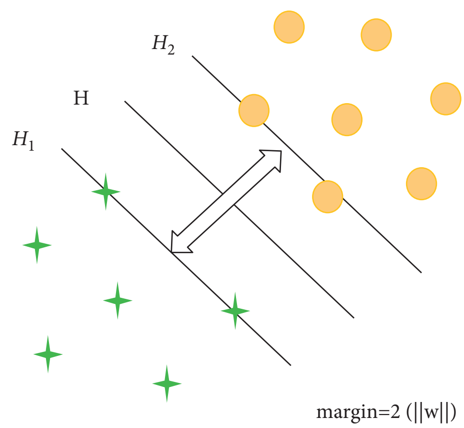

FIGURE 1: SVM classification model.

$$\xi_i^* = \begin{cases} 0, \left|f(x) - y_i\right| < \varepsilon, \\ \left|f(x) - y_i\right| - \varepsilon, \left|f(x) - y_i\right| > \varepsilon. \end{cases} \tag{6}$$

Nonlinear problems can be transformed into linear problems by the use of the loss function ε, penalty parameter C, and slack variable ξ, to minimizing the error rate while realizing the sample separation, and the specific formula is as follows:

$$\min_{\omega, b, \xi} \phi = \frac{1}{2}\omega^T \cdot \omega + C\sum_{i=1}^{n}(\xi_i + \xi_i^*) \text{ s.t. } (w \cdot x_i + b) - y_i \leq \varepsilon + \xi_i,$$
$$y_i - (w \cdot x_i + b) \leq \varepsilon + \xi_i^*. \tag{7}$$

The Lagrange factors a_i and a_i^* are obtained by solving the duality problem. The coefficients of the regression equation are

$$w = \sum_{i=1}^{l}(a_i - a_i^*)x_i. \tag{8}$$

Using the Gaussian RBF,

$$k(x_i, x) = N\left(x_i - x; 0, \sigma^2 I\right), \tag{9}$$

where $N\ (x; \mu; \Sigma)$ is the standard normal distribution. The point product is replaced by kernel estimation, and the expression of the discriminant function is

$$f(x) = \sum_{i=1}^{l}(a_1 - a_i^*)K(x_i, x) + b. \tag{10}$$

Loss function ε, penalty parameter C, and kernel function parameter σ determine the performance of support vector machines. The loss function ε is the error expectation of the estimation function, which affects the number of support vectors to a certain extent; if the penalty parameter C is too small, it will lead to underlearning, and if it is too large, it will lead to overlearning; σ is a parameter of the kernel function, which reflects the characteristics of the training set and determines the complexity of understanding. Therefore, the selection of parameters selection plays a very important role in classification efficiency and accuracy; in this research, the improved PSO algorithm is used to achieve parameter optimization.

3.2. Improved PSO. PSO algorithm is derived from the research on the foraging behavior of birds [24], and this algorithm firstly randomly initializes a group of particles, each particle is a feasible solution to the optimization problem, and the fitness is determined according to the objective function. The particle moves in the direction of the current optimal particle; the optimal solution is obtained through generation by generation search. There are two extreme values in each generation population, the optimal solution p_{best} is found by the particle itself, and the optimal solution of g_{best} is found by the whole population. According to these two extreme values, each particle keeps updating and generates a new generation of population. Once these two extremes are found, the particle's position and velocity are updated as follows:

$$v_{id}' = \omega V_{id} + c_1 \text{rand}()\ (P_{id} - X_{id}) + c_2 \text{rand}()\left(P_{gd} - X_{id}\right)\text{id},$$
$$X_{id}' = X_{id} + V_{id}', \tag{11}$$

where ω is the inertial weight, nonnegative, c_1 and c_2 are nonnegative constants, and V_{id} represents the velocity of the ith particle in the Dth dimension. Studies show that the acceleration coefficient should satisfy $c_1 + c_2 < 4(1 + w)$[25]. Martniez et al. proposed that $c_1 = c_2$ can maximize the second-order stability region [26]. In addition, equal acceleration coefficients can give the same weight to all optimal values (global and local) to avoid the algorithm falling into local optimization at the beginning. P_{id} and P_{gd} are the individual and global optimal values of the corresponding dimensions. rand () generates random numbers between $(0, 1)$.

3.2.1. Improvement of Inertia Weight. The adjustment of inertia weight can be divided into four categories: constant [23], random number [27], time-varying, and adaptive inertia weights. In the initial stage, the step size is large and large inertial weights ω are required. The middle and later stages require strong local development ability, requiring smaller ω. To improve the optimization efficiency and avoid falling into local optimal, this study constructs an inertia decreasing strategy, which gradually reduces the value of ω in the iterative process, introduces the vector h with K elements (K is a constant), and obtains the following algorithm, where k is the modulus of the current iteration number t with respect to K; ω (t), ω_{start}, and ω_{end} are the current, start, and stop inertial weights. When the particles' optimal positions are close to each other, ω (t) increases the reverse trend, preventing premature convergence, and the particles explore more. With each iteration K times, the weight will gradually decrease, thus enhancing the local optimization ability of the particle and finally effectively improving the optimization ability:

$$h(k) = \max_{1\le j\le D}\left\{\mathrm{std}\left(P_{gd}(t) - X_{id}(t)\right)\right\},\ 1 \le k \le K,$$
$$\omega(t) = \omega_{\mathrm{start}} - \omega_{\mathrm{end}}\frac{h(k)}{\max_1 \le k \le K^{\{h(k)\}}}. \tag{12}$$

3.2.2. Introducing Crossover Operator. To avoid the possibility of proposed algorithm falling into local optimum at the end of iteration, a crossover operator is introduced to enhance the information exchange between particles. The search process is controlled by individual optimization, population optimization, and individual genetic operation, so as to make up for the defect that it is easy to fall into local optimization so that the proposed method can jump out of local optimization and get the global optimal solution.

3.2.3. Simulation. To evaluate the effectiveness of PSO-W-GA algorithm, four benchmark functions of Sphere, Schewefel, Rastrigin, and Rosenbrock in CEC2014 were selected to evaluate the algorithm's performance. The comparison between the proposed method and the improved method is as follows: (1) standard PSO, (2) APSO, where $\omega = 1/1 + 1.5\exp(-2.6f) \in [0.4, 0.5]$, f is the evolutionary factor calculated by the distance between particles, and (3) AIWPSO, where $\omega = S(t)/N \in [0, 1]$, N is the population size and S(t) is the best position of the population at t time.

According to experience, the value range of ω is [0.4, 0.9][28]. In this study, $\omega_{\mathrm{start}} = 0.9$, $\omega_{\mathrm{end}} = 0.4$, $c_1 = c_2 = 1.5$, population size is 30, and maximum iteration number is 1000, K is 100, and particle dimension is 30. To evaluate the results of the algorithm, the maximum number of calculation function values is set as FE_S [29]:

$$FE_S = N \times T = N \times 10000 \times \frac{D}{N}. \tag{13}$$

We added the traditional RNN model and CNN model in the simulation experiment for full comparison and experiment; Table 2 shows the experiments results for all listed algorithm.

In Table 2, f_1, f_2, f_3, and f_4 are the test functions Sphere, Schewefel, Rastrigin, and Rosenbrock in Table 2, respectively. Mean and SD, respectively, represent the mean and standard deviation obtained after running the function. According to Table 2 and Figures 2–5, it can be found that the improved algorithm can converges quickly in the above four test functions and outperforms others in the optimal value of convergence.

Figures 2–5 indicates that, under the four evaluation functions, W-GA-PSO algorithm proposed in this study has the fastest convergence speed, the smallest error, and the highest performance, followed by APSO algorithm, which is slightly inferior to the model proposed by us. Then, AIPSO algorithm ranked third, and PSO algorithm had the worst performance, with poor performance in all evaluation indicators. The experiment proved that the model proposed by us could quickly and accurately help the emotion classification model to optimize parameters and achieve good classification results.

4. Proposed Model

This study combines the characteristics of PSO global optimization and SVM fast classification and proposes the SVM model optimized by improved PSO to achieve emotion classification. The specific implementation steps are as follows.

(1) Crawl the comment information from the web page and preprocess the text
(2) The comments' data were divided into training and test sets with corresponding positive and negative labels.
(3) Sogou news corpus is trained by the Skip_gram model in Word2vec; next, the word2vec vector of each word in d_N is calculated by the trained model.
(4) Train the TFIDF model according to the goods review data of this experiment, and obtain the TFIDF weight $\mathrm{tfidf}_{\omega_i}$ of each word in the goods review.
(5) The word vector of each goods review is obtained through equation (1), which is used as the input of the support vector machine.
(6) Initialize the loss function ε, penalty parameter C, and kernel function parameter s of the support vector machine, regard the vector (ε, c, σ) as a particle, define $c_1 = c_2 = 2$ according to the experience, and use formulas (10) and (12) to generate r_1, r_2, and ω.
(7) Define fitness function:

$$F_{\mathrm{fitness}} = \frac{1}{m}\sum_{i=1}^{m}\left(f_i - y_i\right)^2, \tag{14}$$

where f_i, y_i, and m represent the prediction value, actual value, and the number of samples, respectively.
(8) The fitness value of each particle is calculated according to the fitness function, and if the current fitness value is less than the previous fitness value, the original P_{best} is replaced; otherwise, it remains unchanged.
(9) Take the smallest p_{best} and compare it with g_{best}; if $p_{best} < g_{best}$, replace g_{best} with p_{best}; otherwise, retain g_{best}.
(10) The OX crossover operator of GA algorithm is used to generate new particles, and the adaptive values are calculated, and steps (8) and (9) are repeated to update g_{best} and p_{best}.
(11) Judging whether the number of iterations is arrived maximum; if so, proceed to the next step; otherwise, increase the number of iterations by one and skip to step (6).

Table 2: Running results of each algorithm ($D = 30$).

f	Indicators	PSO	AIWPSO	APSO	PSO-W-GA	RNN	CNN
f_1	Mean	1.3×10^{-1}	8.7×10^{-2}	2.4×10^{-2}	1.8×10^{-2}	2.6×10^{-2}	2.8×10^{-2}
	SD	9.1×10^{-2}	1.7×10^{-3}	3.2×10^{-3}	1.5×10^{-3}	3.1×10^{-3}	2.4×10^{-3}
f_2	Mean	9.4×10^{-2}	3.7×10^{-2}	2.8×10^{-2}	2.3×10^{-2}	2.4×10^{-2}	2.6×10^{-2}
	SD	6.3×10^{-2}	4.7×10^{-3}	4.4×10^{-3}	1.4×10^{-3}	4.2×10^{-3}	3.2×10^{-3}
f_3	Mean	7.5	3.8	6.7×10^{-1}	2.4×10^{-1}	3.5×10^{-2}	3.8×10^{-2}
	SD	4.7×10^{-1}	1.5×10^{-1}	4.3×10^{-2}	7.8×10^{-2}	4.5×10^{-3}	4.1×10^{-3}
f_4	Mean	7.3	6.8	4.4	3.1	4.2×10^{-2}	3.9×10^{-2}
	SD	4.9×10^{-1}	3.4×10^{-1}	2.3×10^{-1}	6.5×10^{-2}	5.1×10^{-3}	5.4×10^{-3}

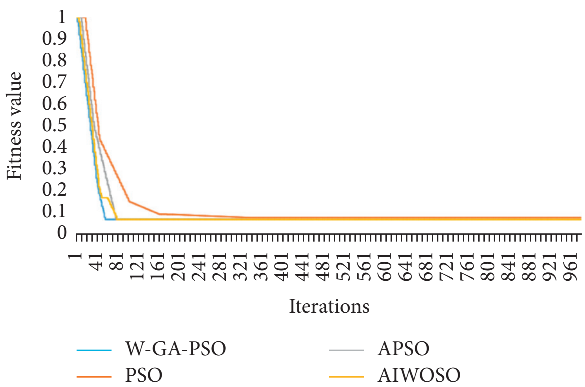

Figure 2: Test results for single peak function: sphere.

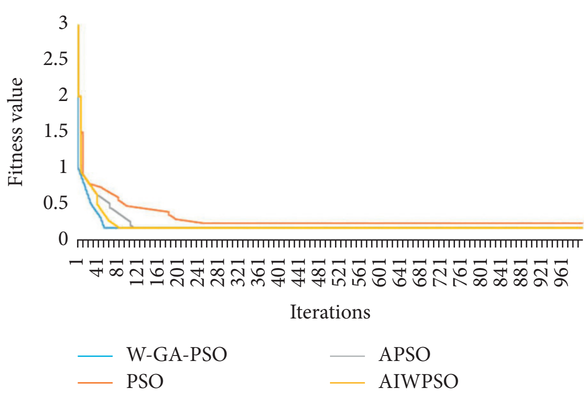

Figure 3: Test results for single peak function: Schewefel.

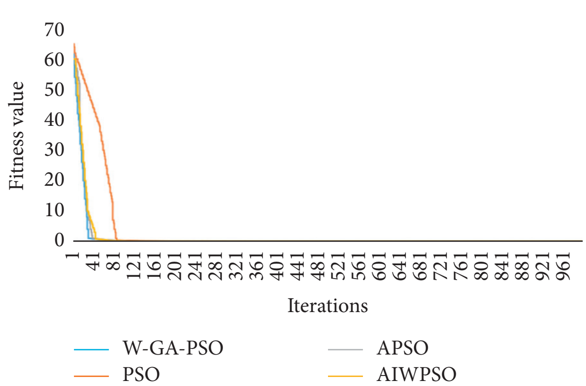

Figure 4: Test results for multimodal function: Rastrigin.

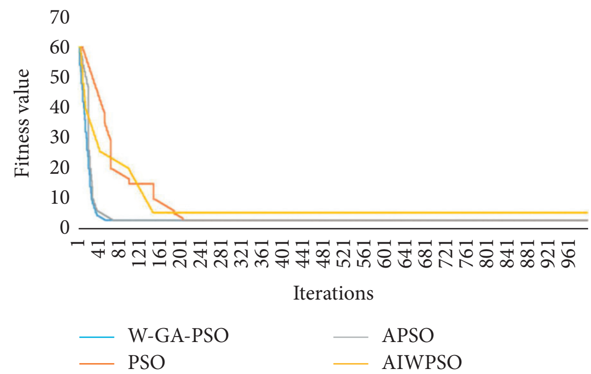

Figure 5: Test results for multimodal function: Rosenborck.

(12) The SVM model was constructed according to the optimized parameters, and the feature vectors processed by weighted word2vec were classified, and the classification results were output.

In this study, the whole experiment process is shown in Figure 6.

5. Experiment Results

Prediction results can be divided into 4 categories: (1) TP, which refers to the number of positive predictions, (2) FN, the positive prediction is the negative quantity, (3) FP, which predicts the negative as the positive quantity, and (4) TN, the negative direction is predicted as the number of negative directions.

The accuracy of positive class is P_pos = TP/(TP + EP); the positive recall rate is R_pos = TP/(TP + EP). The positive class F_1 value is

$$F_1 = \frac{2\times \text{P\_pos}\times \text{R\_pos}}{\text{P\_pos} + \text{R\_pos}}. \tag{15}$$

The accuracy rate and F_1 value were used as evaluation criteria. In our study, a total of 7000 positive comments and 3000 negative comments were selected for ROC curve drawing, and AUC was also used as the standard to evaluate the classification effect. The AUC values of SVM-IPSO are introduced Figures 6 in 7.

Finally, the classification effects under different models are compared in Table 3. The research results show that the

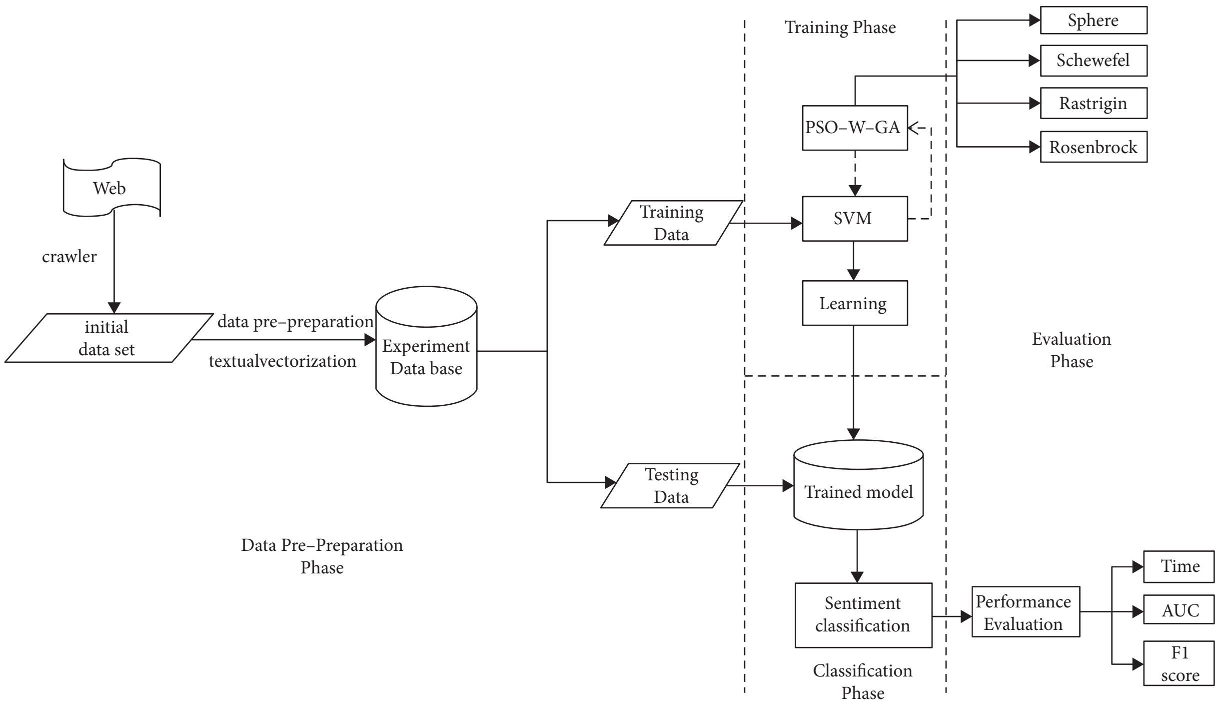

FIGURE 6: The structure of the proposed method.

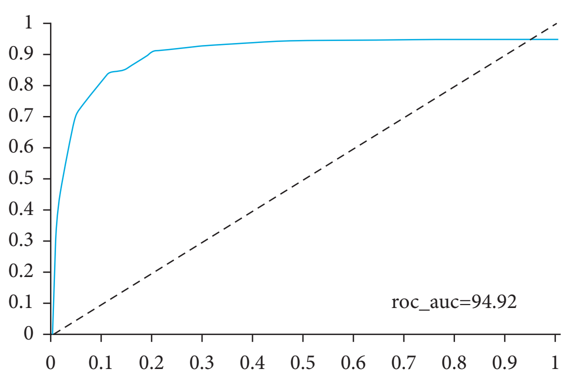

FIGURE 7: The AUC values of SVM-IPSO.

TABLE 3: Classification effect under different models.

Model	T (s)	Auc (%)	F1 (%)
Emotional dictionary	542	81.15	80.13
SVM	225	84.64	84.75
SVM + PSO	267	92.81	92.66
CNN	342	93.84	93.87
SVM + PSO-W-GA	283	94.92	94.82

proposed method has higher prediction accuracy and higher operation efficiency.

In terms of time consumption, emotion dictionary has the worst performance. Both SVM algorithm and our proposed algorithm are less than 300 s. Since we conduct parameter optimization on the basis of the SVM model, all time-consuming SVM algorithm is longer than the single SVM algorithm, and our proposed algorithm is significantly better than other comparison algorithms in terms of accuracy and F1 score. This shows the effectiveness of our algorithm. In addition, the accuracy of CNN algorithm is also high. We will consider adding CNN algorithm into our combined model in future research to further improve the accuracy of our model for emotion classification.

6. Conclusion

In this research, we proposed an improved particle swarm optimization algorithm, and it is used to optimize SVM model parameters for Chinese text emotional classification. The kernel technique is used to learn the nonlinear model in SVM model, and the loss is reduced. The research results demonstrate that the SVM-IPSO makes up for the deficiency of traditional emotional dictionary which is affected by word order and different contexts and solve the problem of local optimization. Meanwhile, it avoids the problem of gradient disappearance or dispersion when using convolution, reduces the cost of parameter optimization and adjustment, has higher operation efficiency and accuracy, and can predict text emotion well. In the future, we will also continue to work on improving the algorithm to enhance the accuracy; at the same time, the experimental data in this study are all extracted from the Internet, so it is necessary to increase the data scale and verify the classification effect of the proposed model in large-scale data. This model has a good effect on binary classification problems, so we will consider extending the model to solve more complex classification problems. The limitations of our study is that the network comment text emotion classification may not be a second classification; we can set up more emotional expression in mood for more accurate classification according to the quantitative, such as the design of the emotional scale to quantify the score values of emotions from low to high, so as to better provide effective decision support for managers.

Data Availability

The experimental data of this research are available upon request from the corresponding author.

Conflicts of Interest

All authors declared that they have no conflicts of interest regarding this study.

Acknowledgments

This work was supported by 2018 the backbone teachers Training Program of Henan Colleges and Universities (no. 2018GGJS183).

References

[1] D. P. Morgan and C. L. Scofield, "natural language processing," *Neural Networks and Speech Processing*, vol. 8, pp. 245–288, Springer US, 1991.
[2] H. Kaur and V. Mangat, "A survey of s\entiment analysis techniques," in *Proceedings of the 2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, IEEE, Palladam, India, February 2017.
[3] K. Dave, S. Lawrence, and D. M. Pennock, "mining the peanut gallery: opinion extraction and semantic classification of product reviews," in *Proceedings of the 12th international conference on World Wide Web*, ACM, NY, USA, May 2003.
[4] A. Go, R. Bhayani, and L. Huang, "Twitter Sentiment Classification Using Distant supervision," pp. 1–6, 2009.
[5] A. Joshi, A. R. Balamurali, and P. Bhattacharyya, "C-Feel-It: A Sentiment Analyzer for Micro-blogs," in *Proceedings of the International Conference on Networked Computing & Advanced Information Management*, IEEE Computer Society, Portland, Oregon, USA, June 2011.
[6] M. Gamon, A. Aue, and S. Corston-Oliver, *Pulse: Mining Customer Opinions from Free Text*, Springer-Verlag, Heidelberg, Germany, 2005.
[7] S. Li, C. R. Huang, and G. Zhou, "Employing personal/impersonal views in supervised and semi-supervised sentiment classification," in *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, DBLP, Uppsala Sweden, July 2010.
[8] Y. G. Li, X. G. Zhou, Y. Sun, and Z. H. Guo, "Research and implementation of chinese microblog sentiment classification," *Journal of Software*, vol. 28, 2017.
[9] R. Johnson and Z. Tong, "Effective use of word order for text categorization with convolutional neural networks," Eprint Arxiv, 2014, https://arxiv.org/abs/1412.1058.
[10] X. Wei and L. Tao, "Aspect Based Sentiment Analysis with Gated Convolutional Networks," in *Proceedings of the Meeting of the Association for Computational Linguistics*, Melbourne, Australia, July 2018.
[11] S. Parupalli, V. A. Rao, and R. Mamidi, "Bcsat : a benchmark corpus for sentiment analysis in Telugu using word-level annotations," *Student Research Workshop in 56th Annual Meeting of the Association for Computational Linguistics*, vol. 1, 2018.
[12] S. Angelidis and M. Lapata, "Multiple instance learning networks for fine-grained sentiment analysis," *Transactions of the Association for Computational Lingus*, vol. 6, pp. 17–31, 2017.
[13] L. Gui, J. Hu, and Y. He, "A Question Answering Approach for Emotion Cause Extraction," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, Copenhagen, Denmark, September 2017.
[14] Y. Zhang, J. Riesa, D. Gillick, A. Bakalov, J. Baldridge, and D. Weiss, "A Fast, Compact, Accurate Model for Language Identification of Codemixed Text," 2018, https://arxiv.org/abs/1810.04142.
[15] M. Bordoloi and S. K. Biswas, "Graph-Based Sentiment Analysis Model for E-Commerce Websites' Data," *Cognitive Informatics and Soft Computing. Advances in Intelligent Systems and Computing*, vol. 768, 2019.
[16] M. Dong, Y. Li, X. Tang, J. Xu, S. Bi, and Y. Cai, "Variable convolution and pooling convolutional neural network for text sentiment classification," *IEEE Access*, vol. 8, no. 99, p. 1, 2020.
[17] X. Liu, T. Tang, and N. Ding, "Social network sentiment classification method combined Chinese text syntax with graph convolutional neural network," *Egyptian Informatics Journal*, vol. 23, 2021.
[18] T. Shoryu, L. Wang, and R. Ma, "A deep neural network approach using convolutional network and long short term memory for text sentiment classification," in *Proceedings of the 2021 IEEE 24th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, IEEE, Dalian, China, May 2021.
[19] Y. Zhu, W. Zheng, and H. Tang, "Interactive dual attention network for text sentiment classification," *Computational Intelligence and Neuroscience*, vol. 2020, no. 3, pp. 1–11, Article ID 8858717, 2020.
[20] Z. Bao and C. Wang, "A multi-agent knowledge integration process for enterprise management innovation from the perspective of neural network," *Information Processing & Management*, vol. 59, no. 2, Article ID 102873, 2022.
[21] L. I. Rong-Yu, W. J. Zhang, and Z. Y. Zhou, "Improved PSO algorithm and its load distribution optimization of hot strip mills," *Computer Science*, vol. 45, 2018.
[22] J. Kennedy, "Particle swarm optimization," in *Proceedings of the 1995 IEEE Int. Conf. Neural Networks*, vol. 4, no. 8, pp. 1942–1948, Perth, Australia, December 2011.
[23] Y. Shi, "A Modified Particle Swarm Optimizer," in *Proceedings of the IEEE Icec Conference*, Anchorage, AK, USA, May 1998.
[24] X.-F. Song, Y. Zhang, D.-W. Gong, and X.-Z. Gao, "A fast hybrid feature selection based on correlation-guided clustering and particle swarm optimization for high-dimensional data," *IEEE Transactions on Cybernetics*, pp. 1–14, 2021.
[25] M. R. Rapaic and Z. Kanovic, "Time-varying PSO – convergence analysis, convergence-related parameterization and new parameter adjustment schemes," *Information Processing Letters*, vol. 109, no. 11, pp. 548–552, 2009.
[26] J. L. F. Martínez and E. G. Gonzalo, "The PSO family: deduction, stochastic analysis and comparison," *Swarm Intelligence*, vol. 3, no. 4, pp. 245–273, 2009.
[27] R. C. Eberhart and Y. Shi, "Tracking and optimizing dynamic systems with particle swarms," in *Proceedings of the 2001 Congress on Evolutionary Computation*, IEEE, Seoul, Korea (South), May 2001.

[28] Y. H. Shi and R. C. Eberhart, “Empirical Study of Particle Swarm optimization,” in *Proceedings of the Congress on Evolutionary Computation*, IEEE, Washington, DC, USA, August 2002.
[29] J. J. Liang, B. Y. Qu, and P. N. Suganthan, “Problem definitions and evaluation criteria for the CEC 2014 special session and competition on single objective real-parameter numerical optimization,” Technical report, Zhengzhou Univ, China, 2013.