



Imbalance Classification Based on Deep Learning and Fuzzy Support Vector Machine

Kefan Wang¹, Jing An¹, Xianghua Ma¹(✉), Chao Ma¹, and Hanqiu Bao²

¹ School of Electrical and Electronic Engineering, Shanghai Institute of Technology, Shanghai 201418, China

xhuam@sit.edu.cn

² Department of Control Science and Engineering, Tongji University, Shanghai 201804, China

Abstract. Imbalanced data is widespread in the fields of medical diagnosis, information security and industrial production. Traditional classification methods can handle balanced data very well. However, when dealing with imbalanced classification, it will favor majority classes, which results in low classification performance. This paper proposes an imbalanced classification method based on deep feature representation, named DL-FSVM. DL-FSVM extracts feature information in the input space using a deep neural network (DNN) to ensure similarity within class and improve the separation between different classes. After obtaining the feature representation, oversampling is performed in this embedding space based on the center distance to enhance the balance of the data distribution. Fuzzy Support Vector Machine (FSVM) is used as the final classifier. Assigning higher misclassification costs to minority class samples through cost-sensitive learning. Experiments were performed on six real-world datasets. The experimental results show that DL-FSVM achieves promising classification performance in three evaluation metrics: G-means, F1-score and AUC.

Keywords: Imbalance classification · Deep neural network · Fuzzy support vector machine

1 Introduction

In many fields, the data are imbalance. There are significant quantitative differences between the samples of different classes. For example, in disease diagnosis [1], most of the data is healthy, and it is difficult to obtain data on diseases. The class that is easily available and more numerous is referred to as majority class, and the class with less data due to the natural frequency of occurrence or data collection is called minority class. The imbalanced data distribution is also exist in the fields of fraud detection [2], computer security [3] and image recognition [4]. In machine learning, there are many well-established classification methods, but they are based on the setting of uniform data distribution and have overall accuracy as the optimization goal. When traditional classification methods are used to deal with imbalanced classification, the result are more in favor of the majority class. Although the overall accuracy is relatively high, the minority class data with important information cannot be accurately identified.

Many imbalance classification algorithms have been proposed in recent decades. These algorithms in general can be grouped into two types: data-level and algorithm-level [5]. The data-level approaches first bring the original imbalanced dataset to balanced distribution by some sampling processing, and then classify it using a traditional classifier. The algorithm-level approaches attempt to improve traditional models by reducing their favoring for the majority class data, and thus adapt to imbalanced data distribution.

In this paper, a novel imbalance classification method based on deep feature representation is proposed, named DL-FSVM. First, from the perspective of data features, embedding space features are obtained by deep neural networks. Appropriate feature representation can lead to better classification quality, and it also enhances the differentiation of features of different classes and the similarity of feature areas of the same class. In addition, it will provide a basis for effective recognition of samples. The deep neural network has a complex nonlinear network structure, which can effectively extract the deep features of samples. When training the network, a triplet loss function [6] is used to enable the network to separate minority class and majority class features. Additionally, Gumbel distribution function [7] is applied as an activation function in the activation layer. This function is continuously differentiable, and it can be easily used as an activation function in stochastic gradient descent optimization neural networks. The original input samples are mapped to the same embedding space after feature extraction. In the embedding space, a new minority class sample is randomly generated based on the distance between the sample and the center of the class, which makes the data distribution balanced. After obtaining the embedding features of samples, FSVM [8] classifies the samples. FSVM introduces membership values (MVs) in the objective function of traditional support vector machine, and it sets different misclassification costs for different classes samples. Misclassification costs are higher for minority class than for majority class. FSVM is a cost-sensitive learning strategy that is effective in improving the recognition of the minority class samples. Traditional classification methods use accuracy as classifier evaluation metrics, but classifiers with accuracy as evaluation metrics tend to ignore the importance of minority class samples. Moreover, accuracy limits the effect of minority class samples on classification performance. Therefore, this paper uses G-means, F1-score and AUC values to evaluate the results more comprehensively.

2 Related Work

Research on imbalanced classification can be grouped into two levels: data-level, and algorithm-level.

2.1 Data-Level

Data resampling is the most important method of data-level, which reduces the imbalance rate (IR) by changing the data distribution. The under-sampling algorithm reduces the bias of model to the majority class samples by decreasing the number of them. Random under-sampling randomly selects and deletes parts of the sample. Some heuristic algorithms are proposed to compensate the limitations of the above non-heuristic method, such as neighborhood cleaning rule (NCL) [9]. Kang et al. added a noise filter to the

under-sampling process [10]. In addition, Kang et al. also proposed a weighted under-sampling algorithm (WU-SVM) [11] based on the geometric distance of the data input space. Oversampling increases the number of minority class samples to make the data balanced. The most representative method, the Small Sample Synthetic Oversampling Technique (SMOTE), was proposed by Chawla et al. [12]. SMOTE randomly selects the k nearest neighbors in the same class of the minority class sample and generates new minority class samples between them using linear interpolation. In addition, Borderline-SMOTE (BSMOTE) [13] and adaptive synthetic sampling approach (ADASYN) [14] are also popular oversampling methods. Matthew et al. proposed the kernel-SMOTE algorithm [15] and weighted K-SMOTE [16] for sampling in the embedding space obtained by SVM.

2.2 Algorithm-Level

Algorithm-level methods use some methods to make appropriate improvements to existing algorithms, such as cost-sensitive learning and ensemble learning methods. As a cost-sensitive algorithm, the fuzzy membership values (MVs) in fuzzy support vector machine (FSVM) [8] reflects the importance of the sample. FSVM differs from the traditional support vector machine in that FSVM introduces the MVs of the sample in the objective function. Batuwita et al. [17] proposed the FSVM-CIL algorithm. FSVM-CIL works in the original data space and calculates the membership values based on the distance between samples. Yu et al. [18] design the membership functions (MFs) based on the relative density within and between classes. This approach makes up for the shortcomings of the distance-based membership values calculation method. ACFSVM [19] based on affinity and class probabilities was proposed by Tao et al. Dealing with imbalance problems using ensemble learning is generally a combination of the standard ensemble methods with the existing methods for classifying imbalanced data, such as SMOTEBagging [20] and SMOTEBoost [21].

The DL-FSVM method proposed in this paper uses FSVM as the base classifier and use data sampling method to obtain balanced data distribution. The new samples generated after oversampling still belong to the minority class, and the use of FSVM can further improve the model's focus on the minority class. In addition, deep neural networks are used to obtain more discriminative feature information, which convenience subsequent classification.

3 Proposed Method

3.1 Feature Extraction with Deep Learning

With the growth of data and the improvement of computing power, the powerful feature extraction capability of deep learning has attracted widespread attention in academia and industry. Deep neural networks (DNNs) have succeeded in significantly improving the best recognition rate of each previous problem by increasing the network depth or changing the structure of the model [22, 23]. Feature representation has a key role in classification quality, so this paper applies the classification method to the embedding space after feature extraction.

For this paper, a DNN is used for the feature extractor because it can learn advanced feature representations from samples [24]. Once training is complete, the hidden feature representations can be used as embedding features to reveal interesting structures in the data. To enhance the differentiation of features from different classes and reduce the differentiation of features from samples in the same class, a triplet loss [6] is used to train the network model, and bring samples in the same class closer and to further separate samples in different classes. Each sample can be converted into a differentiated feature space based on the trained model. The triple loss is based on anchor points, making the features in the embedding space more differentiated. It is defined as:

$$L_{triplet} = (D_{a,min} - D_{a,maj} + r)_+ \quad (1)$$

where r is the margin and set to 0.2 in experiments. D is the function used to calculate the Euclidean distance of samples in the embedding space. a is the anchor point belonging to the minority class, min is the minority class samples, and maj is the majority class samples. $(\cdot)_+$ indicates the value is taken as loss if it is greater than 0. If it is less than 0, the loss is 0.

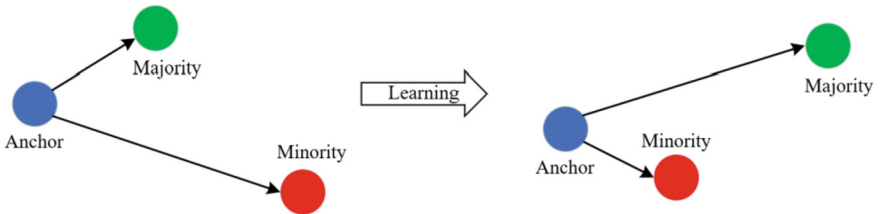


Fig. 1. Optimization result using triplet loss function

Figure 1 shows the results and geometric significance of optimization using triplet loss. Triplet loss tries to learn an embedding space in which anchor is nearer to the minority class samples, and the anchor is further away from the majority class samples. The advantage of triplet loss is detail differentiation, i.e., triplet loss is able to better model the details when the two inputs are similar. This allows better feature representation to be learned.

Gumbel distribution [7] is used as the activation function in DNN. The Gumbel distribution is widely used to design the distribution of extreme value samples of various distributions. The cumulative distribution function (CDF) is defined as:

$$\sigma(x) = e^{-e^{-x}} \quad (2)$$

As shown in Fig. 2, the Gumbel distribution function is continuously differentiable, so it can be easily used as an activation function with optimization in a neural network. Finally, the whole DNN framework used for feature extraction is shown in Fig. 3.

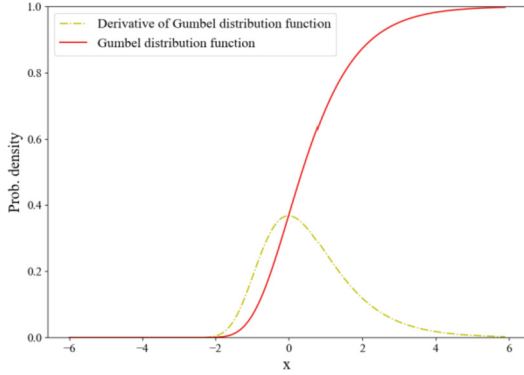


Fig. 2. The curves of CDF and derivatives of Gumbel functions

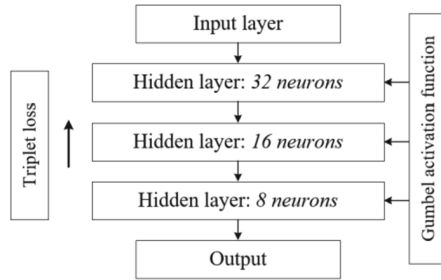


Fig. 3. Deep neural network framework for feature extraction

3.2 Random Feature Oversampling Based on Center Distance

After obtaining the embedding space representation of samples, the data distribution is still imbalanced. The dataset in the embedding space is $X = \{x_1, x_2, \dots, x_n\}$, n is the total number of samples, $x_i = [f_i^1, f_i^2, \dots, f_i^p] \in \mathbb{R}^p$, $i \in 1, 2, \dots, n$. f_i^j is the feature of sample x_i on the j -th dimension, $j \in 1, 2, \dots, p$. For the minority class samples, the set of features in each dimension is denoted as $F = \{F^1, F^2, \dots, F^p\}$, where $F^j = \{f_1^j, f_2^j, \dots, f_{n_min}^j\}$, $j \in 1, 2, \dots, p$. n_min is the number of the minority class samples. F^j is the set of values of all minority class samples on the j -th dimension feature. The feature of each dimension of the new synthetic sample is randomly selected from the corresponding feature set, $x_{syn} = [f_{syn}^1 \in F^1, f_{syn}^2 \in F^2, \dots, f_{syn}^p \in F^p]$.

This method of randomly generated features can increase the diversity of the minority class samples and avoid overfitting. However, the method generates some outliers and noise, so a constraint based on class center distance is used to filter the synthetic samples. As shown in Fig. 4, in the embedding space, the majority class is centered on C_{maj} , the minority class is centered on C_{min} , and the whole data is centered on C_{all} . By calculating the distance between each center and the synthetic sample to determine whether the following equation is satisfied:

$$d(x_{syn}, C_{maj}) > d(x_{syn}, C_{all}) > d(x_{syn}, C_{min}) \tag{3}$$

where $d(\cdot)$ is the distance function. If the synthesized sample fits this condition, it will be kept, otherwise, it will be deleted. In this paper, the influence of irregular data distribution is avoided by calculating the class centers in the embedding space. The number of synthesized samples is set to achieve balanced data distribution.

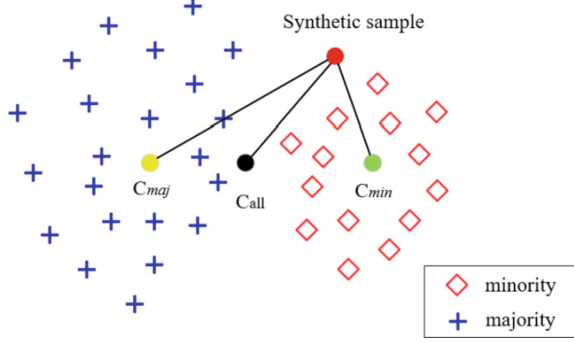


Fig. 4. Validation of the new synthetic feature vector

3.3 Fuzzy Support Vector Machine

In many practical situations, each sample has a different level of importance. For imbalanced data problems, the minority class tends to contain more important information. To improve the quality of classification, each sample needs to be assigned a corresponding weight according to its importance. In this paper, a fuzzy support vector machine (FSVM) [8] is used as the classifier to achieve the assignment of different weights.

The data after sampling as $X = \{x_1, x_2, \dots, x_n\}$, n is the number of samples including all synthetic samples, $x_i \in \mathbb{R}^p$, $i = 1, 2, \dots, n$. p is the feature dimension. Assuming that the dataset is $D = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$. $y_i \in [1, -1]$ is the label of the corresponding sample. FSVM adds an attribute to each sample to expand the original data set to $D = \{(x_1, y_1, s_1), (x_2, y_2, s_2), \dots, (x_n, y_n, s_n)\}$, s_i represents the fuzzy membership value (MV) corresponding to different samples. The value of s reflects the importance level and the misclassification cost of the sample. In this way, the optimization function of FSVM can be written as:

$$\begin{aligned}
 \min : & \frac{1}{2} \|w\|^2 + C \sum_{i=1}^n s_i \varepsilon_i \\
 \text{s.t.} & y_i (w * \phi(x_i) + b) \geq 1 - \varepsilon_i \\
 & \varepsilon_i \geq 0
 \end{aligned} \tag{4}$$

where $\|w\|^2$ represents the margin ratio of the generalization ability of the learning model. The slack variable ε_i represents the acceptable training error degree of the corresponding instance x_i . $C > 0$ is called the penalty parameter, it is a parameter that weighs the size of the separation interval and the number of misclassified points, as well as a trade-off

between learning model accuracy and generalization ability. $\phi(\cdot)$ is the mapping of high-dimensional feature space. The fuzzy membership value s_i can adjust the punishment degree of the corresponding sample. In order to solve this optimization problem, firstly, formula (4) is transformed into an unconstrained problem using the Lagrangian function:

$$L(w, b, \alpha, \beta) = \frac{1}{2}w^2 + C \sum_{i=1}^n s_i \varepsilon_i - \sum_{i=1}^n \alpha_i (y_i (w * x_i + b) - 1 + \varepsilon_i) - \sum_{i=1}^n \beta_i \varepsilon_i \quad (5)$$

The above formula satisfies the following conditions:

$$\begin{aligned} \frac{\partial L(w, b, \alpha, \beta)}{\partial w} &= w - \sum_{i=1}^n \alpha_i y_i x_i = 0 \\ \frac{\partial L(w, b, \alpha, \beta)}{\partial b} &= - \sum_{i=1}^n \alpha_i y_i = 0 \\ \frac{\partial L(w, b, \alpha, \beta)}{\partial \varepsilon_i} &= \varepsilon_i C - \alpha_i - \beta_i = 0 \end{aligned} \quad (6)$$

Introduce formula (6) into formula (5). Transforming optimization problem into the following formula:

$$\begin{aligned} \min : & - \sum_{i=1}^n \alpha_i + \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n y_i y_j \alpha_i \alpha_j \phi(x_i) \phi(x_j) \\ \text{s.t.} & \sum_{i=1}^n y_i \alpha_i = 0, \forall i : 0 \leq \alpha_i \leq s_i C \end{aligned} \quad (7)$$

where α_i is the Lagrangian multiplier corresponding to x_i , and it must also meet the KKT condition:

$$\begin{aligned} \forall i : \alpha_i (y_i (w * \phi(x_i) + b) - 1 + \varepsilon_i) &= 0 \\ \forall i : (s_i C - \alpha_i) \varepsilon_i &= 0 \end{aligned} \quad (8)$$

In this way, the value of α_i can be calculated. Then, according to formula (9), we can calculate w :

$$w = \sum_{j=1}^n \alpha_j \beta_j \phi(x_j) \quad (9)$$

After that, the value of b can be calculated by formula (8). The sample of $\alpha_i > 0$ is called a support vector. When $0 < \alpha_i < s_i C$, the support vector is located on the boundary of the interval. When $\alpha_i = s_i C$, the sample is located between the boundary of the interval and the separation hyperplane or on the side of the separation hyperplane that is misclassified. The biggest difference between traditional SVM and FSVM is that even though two samples have the same value of α_i , the different values of fuzzy membership values s_i can also lead to two samples belonging to different types of support vectors. Under normal circumstances, a smaller s_i is assigned to the majority class to reduce the impact of the numerical advantage of the majority class on the classification results. Finally, the decision function of the optimal separating hyperplane can be expressed as:

$$f(x) = \text{sign}(w * \phi(x) + b) = \text{sign} \left(\sum_{j=1}^n \alpha_j y_j \phi(x_j) \phi(x) + b \right) \quad (10)$$

4 Experiments and Results

4.1 Evaluation Metrics and Datasets

Evaluating the imbalanced classification effect of the model with overall accuracy can cause the model to be biased in favor of the majority class. The overall classification accuracy tends to guarantee the classification effect of only the majority class and ignore the effect of the minority class, which makes the classification effectiveness of the minority class become poor. In this paper, G-mean, F-score and AUC values are used to comprehensively evaluate the classification quality.

Sen is the sensitivity of minority class sample: $TP/(TP + FN)$. *Pre* is the precision of the minority class sample: $TP/(TP + FP)$. *Spe* is the specificity of the majority class sample: $TN/(TN + FP)$. Based on the above definition, G-mean and F-score can be further defined:

$$G - mean = \sqrt{Sen * Spe} \quad (11)$$

$$F - score = 2 * Sen * Pre / (Sen + Pre) \quad (12)$$

AUC is area under curve. It is defined based on the receiver operating characteristic curve (ROC) and its value is less than 1. The algorithm was tested on several datasets from the Keel database, as shown in Table 1.

Table 1. Description of the datasets

Name	Attributes	Data size	Imbalance ratio
ecoli3	7	336	8.6
haberman	3	306	2.78
pima	8	786	1.87
poker-8_vs_6	10	1477	85.88
yeast3	8	1484	8.1
yeast4	8	1484	28.1

4.2 Experiment Settings

In data feature processing, a deep neural network with four fully connected layers is be used. When using fuzzy support vector machine for classification operation, the Gaussian kernel is the kernel function. For FSVM classifier, penalty constant C and the width of Gaussian kernel σ are selected by gird search method from the set $\{10^{-3}, 10^{-2}, 10^{-1}, 1, 10^1, 10^2, 10^3, 10^4\}$ and $\{2^{-5}, 2^{-4}, 2^{-3}, 2^{-2}, 2^{-1}, 1, 2^1, 2^2, 2^3, 2^4\}$. The fuzzy membership value of the minority samples is set to the imbalanced ratio (IR):

$$IR = num_{maj} / num_{min} \quad (13)$$

where num_{min} is the number of the minority class samples, and the minority class is also the positive class. num_{maj} is the number of data of the majority class samples, corresponding to the negative class. For the fuzzy membership value of the majority class, set it to 1. In order to eliminate the randomness, five cross validation is applied, and the algorithms are executed for 5 independent runs.

4.3 Results and Analysis

To compare the classification quality of the proposed algorithm, four baseline methods are used. SMOTE [12] method uses linear interpolation to generate synthetic samples, and finally uses SVM as a classifier. ADASYN [14] assigns the sampling weights of different minority samples based on the number of majority classes in the nearest neighbors. DSVM sets different penalty coefficients C for different classes, the minority class is set to imbalance ratio (IR), and the majority class is set to 1. ACFSVM [19] is a FSVM algorithm combined with sample affinity. The experimental results are shown in Table 2.

In order to observe the table more intuitively, bold the best classification result. It can be seen that DL-FSVM has achieved better classification quality on all three evaluation indicators. On the *ecoli3* dataset, DL-FSVM has an increase of 0.1041 in G-mean compared to SMOTE, and the F1-score also reached an increase of 0.1086. In addition, on other datasets, the classification results of DL-FSVM are better than the baseline SMOTE method. However, on the *poker-8_vs_6* dataset, the baseline SMOTE and ADASYN achieved the best results on the AUC, but its classification performance on G-mean and F1-score was poor.

Compared with the two methods using cost-sensitive learning, the method proposed in this paper has better classification performance. On the *pima* dataset, the fuzzy support vector machine based on sample affinity achieved the best result on F1-score. The result of DL-FSVM is worse than ACFSVM, which is 0.6502. On the *ecoli3* dataset, the G-mean and F1 of DL-FSVM are increased by 0.0201 and 0.0575 respectively compared with the ACFSVM method. The average ranking of algorithm under different evaluation metrics is shown in Fig. 5. It can be seen that the classification performance of DL-FSVM is the best. The imbalanced classification method based on DNN and FSVM proposed in this paper has good robustness and can be used for different types of imbalanced data.

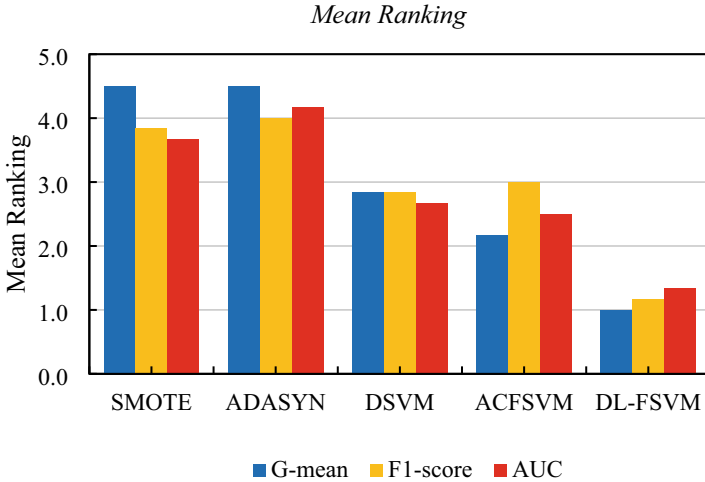
Table 2. Results of different imbalanced classification methods on datasets

Dataset	ecoli3		
Algorithm	G-mean	F-score	AUC
SMOTE	0.7750 ± 0.0545	0.5186 ± 0.0878	0.9100 ± 0.0085
ADASYN	0.7619 ± 0.0134	0.5460 ± 0.0022	0.9042 ± 0.0112
DSVM	0.8317 ± 0.0069	0.5579 ± 0.0269	0.9233 ± 0.0061
ACFSVM	0.8590 ± 0.0671	0.5697 ± 0.0843	0.9368 ± 0.0371
DL-FSVM	0.8791 ± 0.0712	0.6272 ± 0.1263	0.9552 ± 0.0587
Dataset	haberman		
Algorithm	G-mean	F-score	AUC
SMOTE	0.5466 ± 0.0152	0.3342 ± 0.0111	0.5536 ± 0.0385
ADASYN	0.5580 ± 0.0124	0.4267 ± 0.0138	0.6116 ± 0.0080
DSVM	0.5666 ± 0.0052	0.4458 ± 0.0066	0.6303 ± 0.0120
ACFSVM	0.6176 ± 0.0535	0.5439 ± 0.0502	0.6794 ± 0.0542
DL-FSVM	0.6354 ± 0.1442	0.5898 ± 0.0579	0.6945 ± 0.0595
Dataset	pima		
Algorithm	G-mean	F-score	AUC
SMOTE	0.6880 ± 0.0059	0.6023 ± 0.0072	0.7586 ± 0.0130
ADASYN	0.6672 ± 0.0207	0.5770 ± 0.0247	0.7343 ± 0.0188
DSVM	0.7183 ± 0.0036	0.6541 ± 0.0049	0.7634 ± 0.0014
ACFSVM	0.7305 ± 0.0388	0.6614 ± 0.0597	0.8017 ± 0.0467
DL-FSVM	0.7374 ± 0.0259	0.6502 ± 0.0351	0.8152 ± 0.0248
Dataset	poker-8_vs_6		
Algorithm	G-mean	F-score	AUC
SMOTE	0.8487 ± 0.0260	0.8276 ± 0.0281	1.0000 ± 0.0000
ADASYN	0.8630 ± 0.0171	0.8133 ± 0.0189	1.0000 ± 0.0000
DSVM	0.8932 ± 0.0015	0.7905 ± 0.0642	0.9998 ± 0.0002
ACFSVM	0.8953 ± 0.0970	0.7830 ± 0.2871	0.9861 ± 0.0147
DL-FSVM	0.9045 ± 0.0737	0.8313 ± 0.1115	0.9976 ± 0.0057
Dataset	yeast3		
Algorithm	G-mean	F-score	AUC
SMOTE	0.8143 ± 0.0024	0.5544 ± 0.0063	0.9269 ± 0.0028
ADASYN	0.8161 ± 0.0028	0.6193 ± 0.0012	0.9199 ± 0.0011
DSVM	0.9081 ± 0.0057	0.6396 ± 0.0084	0.9687 ± 0.0032

(continued)

Table 2. (continued)

Dataset	ecoli3		
Algorithm	G-mean	F-score	AUC
ACFSVM	0.8987 ± 0.0223	0.6316 ± 0.0421	0.9660 ± 0.0072
DL-FSVM	0.9106 ± 0.0236	0.6875 ± 0.0309	0.9718 ± 0.0187
Dataset	yeast4		
Algorithm	G-mean	F-score	AUC
SMOTE	0.5742 ± 0.0047	0.3069 ± 0.0091	0.8586 ± 0.0010
ADASYN	0.5697 ± 0.0082	0.2971 ± 0.0073	0.8587 ± 0.0007
DSVM	0.8259 ± 0.0125	0.2976 ± 0.0178	0.8914 ± 0.0003
ACFSVM	0.8326 ± 0.0386	0.2391 ± 0.0415	0.9017 ± 0.0312
DL-FSVM	0.8412 ± 0.0927	0.3391 ± 0.0746	0.9158 ± 0.0549

**Fig. 5.** Mean ranking of all compared algorithms on datasets

5 Conclusion

This paper proposes an imbalanced classification method combined with DNN, DL-FSVM. In order to obtain features with intra-class similarity and inter-class discrimination, DNN is trained using triplet loss function and Gumbel activation function to obtain the deep feature representation. To balance the data distribution, a random feature sampling algorithm based on the center of class is used in the minority samples to maintain the diversity of the minority class samples. Fuzzy support vector machine (FSVM) has provided a higher misclassification loss for the minority class, and it enhanced the classification quality of the algorithm for the minority class. Through the experimental

results, it can be found that the proposed DL-FSVM has good classification results on evaluation metrics: G-means, F1-score, and AUC. In future work, more robust feature extractors can be used to provide effective measures for imbalanced classification.

Acknowledgements. This work was supported in part by the National Natural Science Foundation of China (61703279), in part by the Shanghai Municipal Science and Technology Major Project (2021SHZDZX0100) and the Fundamental Research Funds for the Central Universities.

References

1. Bhattacharya, S., Rajan, V., Shrivastava, H.: ICU mortality prediction: a classification algorithm for imbalanced datasets. In: The AAAI Conference on Artificial Intelligence, pp.1288–1294 (2017)
2. Li, H., Wong M.: Financial fraud detection by using grammar-based multi-objective genetic programming with ensemble learning. In: IEEE Congress on Evolutionary Computation (CEC), pp. 1113–1120 (2015)
3. Wang, S., Yao, X.: Using class imbalance learning for software defect prediction. *IEEE Trans. Reliab.* **62**(2), 434–443 (2013)
4. Romani, M., et al.: Face memory and face recognition in children and adolescents with attention deficit hyperactivity disorder: a systematic review. *Neurosci. Biobehav. Rev.* **89**, 1–12 (2018)
5. Tao, X., et al.: Affinity and class probability-based fuzzy support vector machine for imbalanced data sets. *Neural Netw.* **122**, 289–307 (2020)
6. Schroff, F., Kalenichenko, D., Philbin, J.: Facenet: a unified embedding for face recognition and clustering. In: The IEEE Conference on Computer Vision and Pattern Recognition, pp. 815–823 (2015)
7. Cooray, K.: Generalized gumbel distribution. *J. Appl. Stat.* **37**(1), 171–179 (2010)
8. Lin, C.F., Wang, S.D.: Fuzzy support vector machines. *IEEE Trans. Neural Netw.* **13**(2), 464–471 (2002)
9. Laurikkala, J.: Improving identification of difficult small classes by balancing class distribution. In: Quaglini, S., Barahona, P., Andreassen, S. (eds.) AIME 2001. LNCS (LNAI), vol. 2101, pp. 63–66. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-48229-6_9
10. Kang, Q., Chen, X., Li, S., Zhou, M.: A noise-filtered under-sampling scheme for imbalanced classification. *IEEE Trans. Cybern.* **47**(12), 4263–4274 (2016)
11. Kang, Q., Shi, L., Zhou, M., Wang, X., Wu, Q., Wei, Z.: A distance-based weighted under-sampling scheme for support vector machines and its application to imbalanced classification. *IEEE Trans. Neural Netw. Learn. Syst.* **29**(9), 4152–4165 (2017)
12. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: SMOTE: synthetic minority over-sampling technique. *J. Artif. Intell. Res.* **16**, 321–357 (2002)
13. Han, H., Wang, W.-Y., Mao, B.-H.: Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning. In: Huang, D.-S., Zhang, X.-P., Huang, G.-B. (eds.) ICIC 2005. LNCS, vol. 3644, pp. 878–887. Springer, Heidelberg (2005). https://doi.org/10.1007/11538059_91
14. He, H., Bai, Y., Garcia, E.A., Li, S.: ADASYN: adaptive synthetic sampling approach for imbalanced learning. In: IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence), pp. 1322–1328. IEEE (2008)
15. Mathew, J., Luo, M., Pang, C.K., Chan, H.L.: Kernel-based SMOTE for SVM classification of imbalanced datasets. In: 41st Annual Conference of the IEEE Industrial Electronics Society (IECON 2015), pp. 1127–1132. IEEE (2015)

16. Mathew, J., Pang, C.K., Luo, M., Leong, W.H.: Classification of imbalanced data by over-sampling in kernel space of support vector machines. *IEEE Trans. Neural Netw. Learn. Syst.* **29**(9), 4065–4076 (2017)
17. Batuwita, R., Palade, V.: FSVM-CIL: fuzzy support vector machines for class imbalance learning. *IEEE Trans. Fuzzy Syst.* **18**(3), 558–571 (2010)
18. Yu, H., Sun, C., Yang, X., Zheng, S., Zou, H.: Fuzzy support vector machine with relative density information for classifying imbalanced data. *IEEE Trans. Fuzzy Syst.* **27**(12), 2353–2367 (2019)
19. Tao, X., et al.: Affinity and class probability-based fuzzy support vector machine for imbalanced data sets. *Neural Netw.* **122**, 289–307 (2020)
20. Wang, S., Yao, X.: Diversity analysis on imbalanced data sets by using ensemble models. In: *IEEE Symposium on Computational Intelligence and Data Mining*, pp. 324–331. IEEE (2009)
21. Chawla, N.V., Lazarevic, A., Hall, L.O., Bowyer, K.W.: SMOTEBoost: improving prediction of the minority class in boosting. In: Lavrač, N., Gamberger, D., Todorovski, L., Blockeel, H. (eds.) *PKDD 2003. LNCS (LNAI)*, vol. 2838, pp. 107–119. Springer, Heidelberg (2003). https://doi.org/10.1007/978-3-540-39804-2_12
22. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. *Adv. Neural. Inf. Process. Syst.* **25**, 1097–1105 (2012)
23. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778 (2016)
24. Ng, W.W., Zeng, G., Zhang, J., Yeung, D.S., Pedrycz, W.: Dual autoencoders features for imbalance classification problem. *Pattern Recogn.* **60**, 875–889 (2016)