An adaptive fuzzy weight algorithm for the class imbalance learning problem

Vo Duc Quang

School of Information and Communication Technology,
Hanoi University of Science and Technology,
Hanoi, Vietnam
and
Faculty of Information Technology,
Vinh University,
Nghean, Vietnam
Email: quangvd@vinhuni.edu.vn

Tran Dinh Khang*

School of Information and Communication Technology,
Hanoi University of Science and Technology,
Hanoi, Vietnam
Email: khangtd@soict.hust.edu.vn
*Corresponding author

Abstract: In this study, we propose an adaptive fuzzy weight algorithm for the problem of two-class imbalanced learning. Initially, our algorithm finds a set of fuzzy weight values for data samples based on the distance from each sample to the centres of both minority and majority classes. Then, our algorithm iteratively adjusts the fuzzy weight values of sensitive samples on either positive or negative margins or class label noises. By doing so, our algorithm increases the influence of minority samples and decreases the influence of majority samples in forming a classifier model. Experimental results on four benchmark real-world imbalanced datasets including *Transfusion*, *Ecoli*, *Yeast*, and *Abalone* show that our algorithm outperforms the fuzzy SVM-CIL algorithm in terms of classification performance.

Keywords: classification algorithm; class imbalance learning; CIL; fuzzy support vector machines; FSVM; weighted support vector machines; WSVM; support vector machine; SVM.

Reference to this paper should be made as follows: Quang, V.D. and Khang, T.D. (2024) 'An adaptive fuzzy weight algorithm for the class imbalance learning problem', *Int. J. Intelligent Information and Database Systems*, Vol. 16, No. 3, pp.221–240.

Biographical notes: Vo Duc Quang received his BS and MS in Computer Science from the Hanoi University of Science and Technology in 2010 and 2014, respectively. He is currently a Lecturer at the School of Engineering and Technology, Vinh University, Vietnam. His current research interests

include artificial intelligence, machine learning, and fuzzy logic. He is currently pursuing a PhD in Computer Science at the Hanoi University of Science and Technology, Vietnam.

Tran Dinh Khang received his Diploma on Mathematics from the Technische Universitaet Dresden, Germany in 1986 and PhD on Computer Science from the Hanoi University of Technology, Vietnam in 1999. He has been working as an Associate Professor at the School of Information and Communication Technology, Hanoi University of Science and Technology since 2000. His main research topics include computational intelligence, hedge algebras, fuzzy information processing, and decision support systems.

1 Introduction

In the fields of data mining and machine learning, the problem of class imbalance learning (CIL) has been interesting and widely studying by many researchers (Jordan and Mitchell, 2015; Kubat et al., 1998). In this paper, we study the problem of two-class imbalanced learning, where the class with much more samples is called the majority class, and the class with a few samples is called the minority class (Fernández et al., 2018; Japkowicz and Stephen, 2002; Liu, 2021). In general, traditional classification algorithms often consider datasets with an equal number of samples in two classes and try to find classifier models with the highest accuracy rate. In the case of two-class imbalanced datasets, samples of the minority class represent rare cases, while these of the majority class represent the normal cases. This means that the number of samples of the majority class appears a large proportion in the datasets. Therefore, the classifier models formed by these algorithms often tend to classify samples of the minority class into the majority class. Obviously, these models are not good, because samples of the minority class are often important and should be prioritised to classify correctly in classifier models. For example, in the problem of diagnosing diseases, patients' information is represented by samples of the minority class in the dataset. Accurate diagnosis of patient cases is very important. Therefore, the trained models need to have the highest accuracy in diagnosing patient cases that play the role of the minority class in the dataset. To solve this problem, researchers often propose approaches to either data level or algorithm level modifications.

At the data level approach, researchers focus on improving imbalanced datasets such that datasets are more balanced than previous ones, such as reducing samples of the majority class (called under-sampling technique) (Liu et al., 2008; Rekha et al., 2020), generating samples of the minority class (called over-sampling technique) (Chawla et al., 2002; Ning et al., 2022), or combining them (Zeng et al., 2016) before using traditional classification algorithms. Tomek links (Tomek, 1976) is an algorithm to determine the pairs of two samples belonging to two different classes with the closest distance from each other. Therefore, the pairs determined by this algorithm often are either on classification boundaries or noises in datasets. Hereafter, we call a Tomek link pair as TLP. In the sampling techniques on imbalanced datasets, algorithms often delete TLPs to eliminate noises and make classification boundaries clearer and more separate on datasets. If so, this may be not good since it modifies the primitive datasets. Alternatively, we can assign some weight to each sample to indicate how important a sample is for constructing a classifier model. Specifically, if a sample in a pair of TLPs is noise, then we decrease its weight, however, if a pair in TLPs is on a boundary, then we increase the weight of the minority sample and decrease the weight of the majority sample.

At the algorithm level approach, researchers focus on improving classification algorithms in terms of learning with costs on samples or using appropriate loops to prioritise the correct classification of samples in the minority class (Quang et al., 2021; Elkan, 2001; Sun et al., 2007). Among these improved algorithms, the support vector machine algorithm (SVM) is a powerful margin-based classification algorithm and is very suitable to improve for datasets with different characteristics (Akbani et al., 2004). So far, several improvements of SVM have been proposed. (Lin and Wang, 2002) proposed a fuzzy SVM algorithm, in which they applied a fuzzy membership to each input sample of SVM and reformulated SVM into fuzzy SVM such that different input samples can make different contributions to the learning of decision surface. (Yang et al., 2005) proposed a weighted SVM algorithm that assigns different weights to different samples such that the algorithm learns the decision surface according to the relative importance of samples in the training dataset. Hao et al. (2022) proposed a rule extraction from biased random forest and fuzzy SVM for early diagnosis of diabetes. Ma et al. (2018) proposed a novel method combining fuzzy SVM and sampling for imbalanced datasets. Batuwita and Palade (2010) proposed a fuzzy SVM-CIL algorithm based on the fuzzy SVM algorithm to improve classification efficiency for imbalanced datasets. Fuzzy SVM-CIL assigns weights to samples based on fuzzy membership functions in terms of prioritising higher weight values for minority samples and lower weight values for majority samples. The fuzzy membership functions of the samples are calculated based on the distance measured by three methods:

1 the distance from samples to their class centre

2 the distance from samples to the estimated hyperplane

3 the distance from samples to an actual hyperplane.

A sample that is further away from its class centre or hyperplanes is considered less important, and therefore it is assigned by a smaller fuzzy weight value. We found that in fuzzy SVM-CIL, the fuzzy membership functions only consider the distance from samples to their class centre without considering the centre of the other class. Therefore, fuzzy SVM-CIL is inefficient for the case where samples have the same distance to their class centre, but they have a different distance to the other class centre. Hereafter, we call fuzzy SVM as FSVM, weighted SVM as WSVM, and fuzzy SVM-CIL as FSVM-CIL.

In this study, we propose a fuzzy membership function to compute initial fuzzy weight values, a method to adjust fuzzy weight values, and an adaptive fuzzy weight algorithm for the problem of two-class imbalanced learning. Our fuzzy membership function is designed based on the distance from each sample to the centres of both minority and majority classes. Meanwhile, our method of adjusting fuzzy weight values is designed based on the positions of the samples in a sensitive region determined by TLPs. From these proposals, our algorithm iteratively adjusts fuzzy weight values to obtain an efficient classifier model for both minority and majority samples. Experimental results on four benchmark real-world imbalanced datasets consisting of *Transfusion*, *Ecoli*, *Yeast*, and *Abalone* show that our algorithm gives a better classification performance than FSVM-CIL (Batuwita and Palade, 2010).

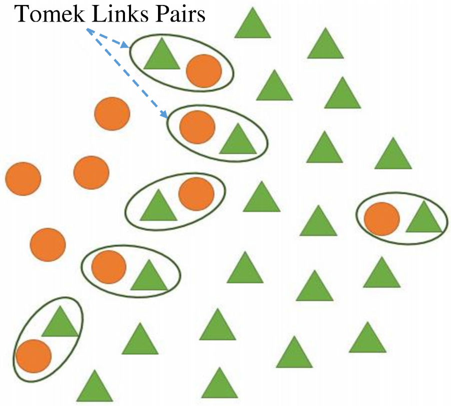
The remainder of our paper is structured as follows. Section 2 provides the preliminaries related to our study. Section 3 presents our proposed algorithm. Section 4 gives experimental results and discussions. Section 5 concludes our work.

2 Preliminaries

2.1 Tomek links

Tomek links pairs, namely TLPs, are defined as pairs of two samples belonging to two different classes with the shortest distance. It is assumed that S_{min} and S_{maj} are sets of minority and majority samples, respectively, and $d(x_i, x_j)$ is the distance between $x_i \in S_{min}$ and $x_j \in S_{maj}$. The pair (x_i, x_j) is called a TLP if there exists no x_k such that $d(x_i, x_k) < d(x_i, x_j)$ or $d(x_j, x_k) < d(x_i, x_j)$. Figure 1 illustrates the locations of TLPs in a dataset. When we locate a TLP, either two samples in TLP create a class boundary or one of two samples in TLP is noisy.

Figure 1 Tomek links pairs (see online version for colours)



In the problem of class imbalance learning, TLPs are often used to clean up the datasets after SMOTE algorithm and its variants to generate more aggregate samples for the minority class (Han et al., 2005; He et al., 2008; Chawla et al., 2003). As a result, the generated datasets are more balanced and clearer and therefore, classification algorithms improve more accurate performance on minority samples. So far, methods of using and improving Tomek links algorithm are very diverse such as OOS (Kubat, 2000), CNN+Tomek links (Batista et al., 2004), NCL (Laurikkala, 2001), SMOTE+ENN (Xu et al., 2020), etc. However, we recognised that in the above methods, removing such TLPs will alter the characteristics of original datasets, and this leads to the fact that some of the minority samples will be discarded, while they need to be kept at most. To overcome this weakness, after identifying the pairs of TLPs and evaluating the position of each pair of TLPs in the sample distribution space, we adjust the weights for the samples to prioritise increasing the importance of positive samples, decreasing the importance of the negative samples, and significantly decreasing the importance of the samples as noise without altering the characteristics of original datasets.

2.2 WSVM and FSVM-CIL algorithms

The SVM is a powerful classifier based on the optimisation of classifier margins. This algorithm is scalable and improved to create an efficient classification model for datasets with different characteristics. For a two-class learning problem, it is assumed that the dataset is $D = \{(x_i, y_i)|i = 1, 2, ..., N\}$, where $x_i \in R^n$ is a n-dimensional input feature vector and $y_i \in \{-1, +1\}$ is the class label of x_i. SVM tries to find the optimal parameter values for the separate hyperplane in the feature space R^n, in which the separate hyperplane is expressed by

$$\langle \omega, x \rangle + b = 0, \tag{1}$$

where ω is a parameter matrix and b is a constant.

For the problem of class imbalanced learning, many improvements of SVM have been proposed (Lin and Wang, 2002; Akbani et al., 2004; Yang et al., 2005; Batuwita and Palade, 2010). One typical of them is WSVM (Yang et al., 2005). WSVM assigns fuzzy weights to the training samples and therefore, it results in the influence of samples in forming a classification model. In WSVM, the objective function is represented as follows:

$$\begin{aligned} &\min \frac{1}{2} \|\omega\|^2 + C \sum_{i=1}^{N} m_i \varepsilon_i, \\ &\text{st. } y_i * (\langle \omega, x_i \rangle + b) \geq 1 - \varepsilon_i, \\ &\quad \varepsilon_i \geq 0, \text{with } i = 1, 2, ..., N, \end{aligned} \tag{2}$$

where slack variables $\varepsilon_i > 0$ represent the *misclassification* of the samples, $\sum_{i=1}^{N} \varepsilon_i$ is the sum of errors on samples, and C is a parameter to control the trade-off between the maximum margin width and the minimum error total on samples. The best C value can be found after running pre-tests on datasets. It should be noted that each m_i is a weight value that reflects the importance for correctly classifying a sample x_i. The larger the weight value of a sample is, the more important the sample is in correctly classifying itself. In contrast, the smaller the weight value of a sample is, the smaller the influence of the sample on the generation of the optimal separate hyperplane is.

An efficient improvement of FSVM for the problem of two-class imbalanced learning is FSVM-CIL (Batuwita and Palade, 2010). In FSVM-CIL, fuzzy membership functions are designed to determine fuzzy weight values to meet the main objectives: reducing the influence of imbalance between data classes; reflecting the importance of samples in a training model; and reducing the influence of outliers and noise samples. FSVM-CIL assigns higher fuzzy weight values, denoted by $m_i^+ (i = 1, 2, ..., N)$, to minority samples x_i^+ (labelled +1, so-called positive samples) and lower fuzzy weight values, denoted by $m_i^- (i = 1, 2, ..., N)$, to majority samples x_i^- (labelled -1, so-called negative samples). The fuzzy weight values are calculated by

$$\begin{aligned} m_i^+ &= f\left(x_i^+\right) * r^+, \\ m_i^- &= f\left(x_i^-\right) * r^-, \end{aligned} \tag{3}$$

where $f(x_i) \in (0,1)$ is a fuzzy membership function that reflects the importance of x_i in its own class, while r^+ and r^- represent the influence level of the imbalanced ratio in datasets. FSVM-CIL assigns $r^+ = 1$ and $r^- = r$, where r is the imbalanced ratio between the minority class and the majority class and thus $r < 1$. Accordingly, the fuzzy weights on the positive samples have values in the range $(0,1)$, while the negative samples have values in the range $(0,r)$.

The fuzzy membership function $f(x_i)$ is determined based on the position of the sample x_i in the feature space R^n. Accordingly, the samples with a distance closer to the class centre, actual or estimated hyperplane are considered to have a higher influence than the other samples, i.e., they have fuzzy weight values that are higher than those of the others. In FSVM-CIL, $f(x_i)$ uses three distance measures from the sample x_i: to its class centre (d_i^{cen}); to the estimated hyperplane (d_i^{shp}) defined as the centre of the entire dataset; to the actual hyperplane (d_i^{hyp}) formed by a basic SVM model. For each distance-based method, FSVM-CIL constructs two fuzzy membership functions, one is a fuzzy linear function (lin) and the other is a fuzzy exponential function (exp). As a result, six fuzzy membership functions of sample x_i are formed as follows:

1 Based on the distance to the own class centre:

$$f_{lin}^{cen}(x_i) = 1 - \frac{d_i^{cen}}{\max\left(d_i^{cen}\right) + \Delta}, \tag{4}$$

$$f_{exp}^{cen}(x_i) = \frac{2}{1 + \exp\left(\beta d_i^{cen}\right)}. \tag{5}$$

2 Based on the distance to the estimated hyperplane:

$$f_{lin}^{shp}(x_i) = 1 - \frac{d_i^{shp}}{\max\left(d_i^{shp}\right) + \Delta}, \tag{6}$$

$$f_{exp}^{shp}(x_i) = \frac{2}{1 + \exp\left(\beta d_i^{shp}\right)}. \tag{7}$$

3 Based on the distance to the actual hyperplane:

$$f_{lin}^{hyp}(x_i) = 1 - \frac{d_i^{hyp}}{\max\left(d_i^{hyp}\right) + \Delta}, \tag{8}$$

$$f_{exp}^{hyp}(x_i) = \frac{2}{1 + \exp\left(\beta d_i^{hyp}\right)}. \tag{9}$$

It should be noted that in equations (4)–(9), Δ is a small positive value to avoid the case where $f_{lin}^{cen}(x_i) = 0$, $f_{lin}^{shp}(x_i) = 0$, $f_{lin}^{hyp}(x_i) = 0$, and $\beta \in [0,1]$ to control the slope of the exponential functions $f_{exp}^{cen}(x_i)$, $f_{exp}^{shp}(x_i)$, and $f_{exp}^{hyp}(x_i)$.

3 Proposed algorithm

In this section, we propose a fuzzy membership function to compute initial fuzzy weight values for samples, a method of adjusting fuzzy weights to fit samples, and an adaptive fuzzy weight algorithm for the problem of two-class imbalanced learning, namely AFW-CIL. The general model of our AFW-CIL is shown in Figure 2. First, AFW-CIL computes initial fuzzy weight values for all training samples. Then, AFW-CIL runs in T loops, in which at each loop $t = 1, 2, ..., T$, it performs the following three steps:

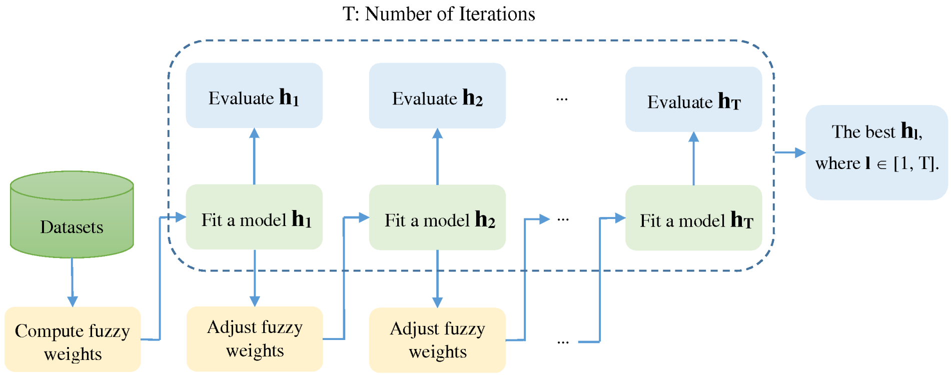
1 fit a classifier model h_t with the adjusted fuzzy weights

2 evaluate the fitted classifier model h_t

3 adjust fuzzy weights for the next loop.

After T loops, AFW-CIL returns the best classifier model h_l $(l \in [1, T])$ in terms of maximum geometric mean. It should be recalled that in the problem of two-class imbalanced learning, the geometric mean is the squared root of the product of the sensitivity and specificity measures. A model h_l with the maximum geometric mean indicates that it gives the maximum accuracy on each of two classes while keeping these accuracies balanced. AFW-CIL is described more details in the following subsections.

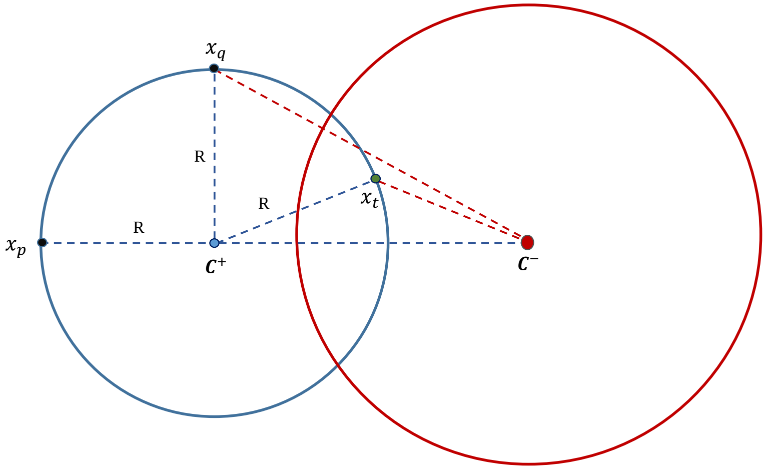
Figure 2 The general model of the proposed algorithm (see online version for colours)



3.1 Fuzzy membership function

In FSVM-CIL, the fuzzy membership function of each sample is defined based on the distance measures given in equations (4)–(9). In cases where the fuzzy membership function is calculated based on the distance from each sample to its class centre, FSVM-CIL uses equations (4) and (5). As a result, the samples with a distance closer to the centre of their class are considered to have a higher influence and therefore, they have higher fuzzy weight values. In contrast, the samples farther from the centre of their class have lower fuzzy weight values.

Figure 3 The relative positions of samples between two classes (see online version for colours)



We recognised that FSVM-CIL only considers and evaluates the importance of samples based on the distance to their class centre and thus, it is quite simple and incomplete. For an example illustrated in Figure 3, it is assumed that C^+ and C^- are the centres of two sets of samples labelled +1 and –1, respectively. Moreover, x_p, x_q, and x_t are samples labelled +1 having the same distance R to centre C^+, meaning that $d(x_p, C^+)$ = $d(x_q, C^+) = d(x_t, C^+) = R$. If we apply FSVM-CIL to find a classifier model, then the fuzzy weight values m_p^+, m_q^+ and m_t^+ of x_p, x_q, and x_t are calculated by the fuzzy membership functions $f(x_p)$, $f(x_q)$, and $f(x_t)$ as given in equation (3), where $f(x_p) = f(x_q) = f(x_t)$ since $r^+ = 1$. This means that three samples x_p, x_q, x_t are equally important in contributing to form a classifier model. However, we see that the positions of these three samples to the centre C^- of the opposite class are clearly different: $d(x_p, C^-) > d(x_q, C^-) > d(x_t, C^-)$. In terms of significance, x_t can be a sensitive sample, because it is closest to the centre C^-. The influenced level on a classifier model of x_p must be greater than x_q and that of x_q must be greater than x_t, i.e., $m_p^+ > m_q^+ > m_t^+$.

To deal with the weakness of FSVM-CIL, we propose a fuzzy membership function for samples based on considering the distance from samples to the centres of two classes. Specifically, if x_i is a sample, then the fuzzy membership function for x_i is defined by

$$f_{lin}^{cen\_2c}(x_i) = \frac{d_{xi}^{cen\_opp}}{d_{x_i}^{cen\_own} + d_{cen\_own}^{cen\_opp} + \Delta}, \tag{10}$$

where $d_{x_i}^{cen\_opp}$ is the distance from x_i to the opposite class centre, $d_{x_i}^{cen\_own}$ is the distance from x_i to its class centre, $d_{cen\_own}^{cen\_opp}$ is the distance between the centres of two classes, and Δ is a small positive value.

Algorithm 1 Calculate fuzzy weights for a dataset

```
Fnfunction CalFW($D, r^+, r^-, \Delta$) for $i$ = 1 to $N$ do
    calculate the fuzzy membership $f_{lin}^{cen\_2c}$ by Eq. (10);
    if $y_i = +1$ then
        $m_i^+ = f_{lin}^{cen\_2c}(x_i) \times r^+$;
    else
        $m_i^- = f_{lin}^{cen\_2c}(x_i) \times r^-$;
    end
end
return $\{m_i^+, m_i^-\}, i = 1, 2, ..., N$;
end function
```

Given a dataset $D = \{(x_1, y_1), ..., (x_N, y_N)\}$ of N samples, where $y_i \in \{-1, +1\}, \forall i = 1, 2, ..., N$, we set $r^+ = 1$ and denote r^- by the imbalanced ratio of the number of minority samples to that of majority samples. Our algorithm to find the fuzzy weight values m_i^+ and m_i^- for $x_i \in D$ is shown in Algorithm 1. At each loop, the algorithm calculates the fuzzy membership function $f_{lin}^{cen\_2c}$ for each sample x_i by equation (10) with $\Delta = 10^{(-6)}$. If the current sample x_i belongs to the minority class, i.e., $y_i = +1$, then m_i^+ is the value of the fuzzy membership function of x_i since $r^+ = 1$. Otherwise, m_i^- is decreased by r^-. After N loops, the algorithm returns a set of fuzzy weight values $\{m_i^+, m_i^-\}$ for samples in $x_i \in D$ $(i = 1, 2, ..., N)$.

3.2 *Adjusting fuzzy weights*

In the problem with highly imbalanced data, researchers often use the methods of reducing negative samples (Liu et al., 2008) and/or generating positive samples (Chawla et al., 2002) to get a more balanced dataset. Then, they use Tomek links algorithm (Tomek, 1976) to remove noisy samples as well as improve classification boundaries in the formed datasets. However, we recognised that if we do so, the primitive datasets are modified. Alternatively, we determine sensitive samples based on the Tomek links algorithm and design rules of assigning and adjusting fuzzy weights of samples in four cases. Depending on each case, fuzzy weights of samples are adjusted to control their influence in forming a classifier model. Figure 4 illustrates four cases for sensitive samples found by the relative positions of TLPs along with their K-nearest neighbours, where samples of the minority class are represented by circles, samples of the majority class are represented by triangles, and $K = 5$. Specifically, a pair of TLPs is either:

1 in the positive margin

2 outside the positive margin, but it is a negative label noise

3 in the negative margin

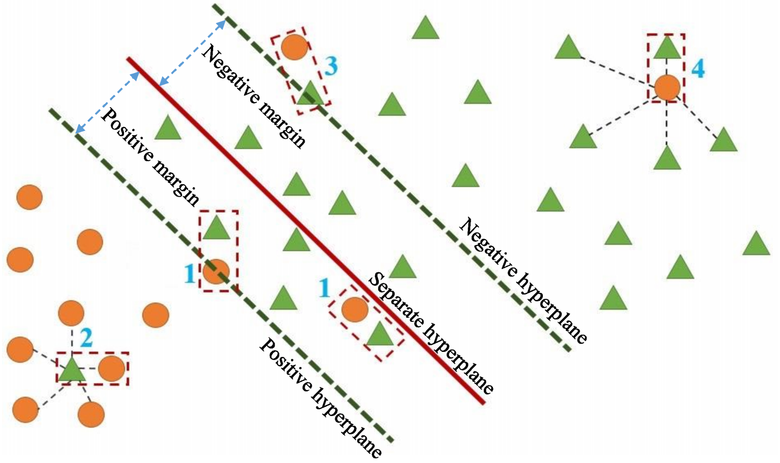
4 outside the negative margin, but it is a positive label noise.

Our algorithm to adjust the fuzzy weight values is shown in Algorithm 2, where h_t is a WSVM classifier, h_{KNN} is a KNN classifier, K is the number of nearest neighbours of x_i, and $\{\sigma_1, \sigma_2, \sigma_3, \sigma_4\}$ is a set of parameters to adjust fuzzy weight values. The algorithm runs as follows. First, it finds a set $\{(x_i, x_j)\}$ of TLPs (lines 2–7). Then, for each $(x_i, x_j) \in$ TLPs such that $y_i = 1$ and $y_j = -1$, it checks and adjusts the fuzzy weight values as follows, where cases 1, 2, 3, and 4 are illustrated in Figure 4:

1 If x_i and x_j are classified by h_t into the minority class, i.e., $h_t(x_i) = 1$ and $h_t(x_j) = 1$ or (x_i, x_j) is in the positive margin (case 1), then m_i^+ is adjusted up by σ_1 to increase the influence of x_i, while m_j^- is adjusted down by σ_1 to decrease the influence of x_j (lines 10–11). However, if the K-nearest neighbours $x_{j_k} (j_k = 1, 2, ..., K)$ of x_j are in the minority class, i.e., x_j is a negative label noise (case 2), then m_j^- is adjusted down significantly by σ_2 to decrease significantly the influence of x_j (lines 12–14). Therefore, σ_1 and σ_2 are chosen such that $0 < \sigma_1 < 0.5$ and $0 < \sigma_2 < 1$.

2 If x_i and x_j are classified by h_t into the majority class, i.e., $h_t(x_i) = -1$ and $h_t(x_j) = -1$ or (x_i, x_j) is in the negative margin (case 3), then m_i^+ is adjusted up by σ_3 to increase the influence of x_i, while m_j^- is adjusted down by σ_3 to decrease the influence of x_j (lines 17–18). However, if the K-nearest neighbours $x_{i_k} (i_k = 1, 2, ..., K)$ of x_i are in the majority class, i.e., x_i is a positive label noise (case 4), then m_i^+ is adjusted down significantly by σ_4 to decrease significantly the influence of x_i (lines 19–21). Therefore, σ_3 and σ_4 are chosen such that $0 < \sigma_3 < 0.5$ and $0 < \sigma_4 < 1$.

Figure 4 An illustration of four cases for sensitive samples found by TLPs (see online version for colours)



By doing so, our algorithm increases m_i^+ and decreases m_j^- to prioritise the correct classification of minority samples x_i. Moreover, if x_i is a positive label noise and x_j is a negative label noise, then our algorithm decreases significantly m_i^+ and m_j^- to reduce the influence of samples x_i and x_j in forming classifier models. Our algorithm returns a set of adjusted fuzzy weights $\{m_i^+, m_i^-\}$ for samples $x_i \in D$ $(i = 1, 2, ..., N)$.

Algorithm 2 Adjust fuzzy weights

```
Fnfunction AdjFW($(D, h_t, K, \sigma_1, \sigma_2, \sigma_3, \sigma_4)$) Initialise $TLPs = \{\}$;
for $i = 1$ to $N$ do
    find a sample $(x_j,\ y_j)$ such that $(x_i,\ y_i)$ and $(x_j,\ y_j)$ are the nearest neighbours of
    each other;
    if $(x_i,\ x_j) \notin TLPs$ and $(y_i \neq y_j)$ then
        $TLPs = TLPs \cup \{(x_i, x_j)\}$;
    end
end
for each $(x_i, x_j) \in TLPs$ such that $y_i = 1$ and $y_j = -1$ do
    if $h_t(x_i) = 1$ and $h_t(x_j) = 1$ then
        $m_i^+ = m_i^+ \times (1 + \sigma_1)$;
        $m_j^- = m_j^- \times (1 - \sigma_1)$;
        if $h_{KNN}(x_{j_k}) = 1$ then
            $m_j^- = m_j^- \times \sigma_2$;
        end
    end
    if $h_t(x_i) = -1$ and $h_t(x_j) = -1$ then
        $m_i^+ = m_i^+ \times (1 + \sigma_3)$;
        $m_j^- = m_j^- \times (1 - \sigma_3)$;
        if $h_{KNN}(x_{i_k}) = -1$ then
            $m_i^+ = m_i^+ \times \sigma_4$;
        end
    end
end
return $\{m_i^+, m_i^-\}, i = 1, 2, ..., N$;
end function
```

3.3 AFW-CIL algorithm

In this section, we present an AFW-CIL algorithm for the problem of two-class imbalanced learning, as shown in Algorithm 3. Our AFW-CIL uses WSVM as a basic classifier model, denoted by h_t, for adjusting fuzzy weights.

Algorithm 3 AFW-CIL algorithm

```
Fnfunction AFW_CIL($(D, h_1, K, \sigma_1, \sigma_2, \sigma_3, \sigma_4, T)$) find a set of fuzzy weights
    $\{m_i^+, m_i^-\}$ for $D$ by CalFW($(D, r^+, r^-, \Delta)$);
for $t := 1$ to $T$ do
    fit a classifier model $h_t$ using WSVM with $\{m_i^+, m_i^-\}$ on $D$;
    adjust fuzzy weights $\{m_i^+, m_i^-\}$ of $D$ by AdjFW($(D, h_t, K, \sigma_1, \sigma_2, \sigma_3, \sigma_4)$);
    evaluate the geometric mean $g(h_t)$ of $h_t$;
end
$l := max(g(h_t)), \forall t = 1, 2..., T$;
return $h_l$;
end function
```

First, AFW-CIL finds a set of fuzzy weights $\{m_i^+, m_i^-\}$ for samples in D by calling the function given in Algorithm 1. Then, AFW-CIL runs in T loops, in which at each loop $t = 1, 2, ..., T$, it performs as follows:

1 fits a classifier model h_t using WSVM with fuzzy weights $\{m_i^+, m_i^-\}$ on samples of D

2 adjusts fuzzy weights $\{m_i^+, m_i^-\}$ of D by calling the function given in Algorithm 2

3 evaluates the geometric mean $g(h_t)$ of the model h_t. After T loops, the algorithm returns a model h_l $(l \in [1, T])$ such that the geometric mean $g(h_l)$ is maximum.

This means that h_l gives the maximum accuracy on each of two classes while keeping these balanced accuracies.

4 Experiments

In this section, we present three experiments to evaluate the efficiency of our proposed:

1 fuzzy membership function

2 method of adjusting fuzzy weights

3 AFW-CIL algorithm.

We used four benchmark real-world imbalanced datasets consisting of *Transfusion*, *Ecoli*, *Yeast*, and *Abalone* published in UCI Machine Learning Repository (Dua and Graff, 2017) for our experiments. Table 1 shows the details of these datasets in the descending order of the imbalanced ratio of positive and negative samples. To consider the performance of our proposals in datasets with different characteristics, we chose two datasets containing a small number of samples and having a low imbalance rate, and two datasets containing a large number of samples and having a very high imbalance rate. Moreover, we used measures of *sensitivity* (SE), *specificity* (SP), *geometric mean* (GM), *area under curve* (AUC), *accuracy* (ACC), and *F1-score* (F1S) to evaluate experimental results. It should be noted that SE, GM and AUC are three most important measures considered to evaluate for the two-class imbalanced learning problem. Besides, we chose FSVM-CIL to compare its results with those of our proposed 1, 2 and 3 since FSVM-CIL has been shown to be more efficient than both WSVM and FSVM for the two-class imbalanced learning problem (Batuwita and Palade, 2010).

For each dataset, we used the 5-fold cross validation. In FSVM-CIL, we set parameters to optimal values inherited from FSVM-CIL as follows: $\beta \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9{,}1\}$, $\Delta = 10^{(-6)}$, and $C = 100$. In our AFW-CIL, we set $K = 5$, $\sigma_2 = \sigma_4 = 0.5$, $\sigma_1 = 0.1$, $\sigma_3 = 0.1$. Besides, we used the Euclidean distance in the fuzzy membership functions. We implemented the AFW-CIL and FSVM-CIL algorithms by Python 3.11 software on a laptop computer with Core i7-8550U CPU 1.8 GHz and 16 GB RAM, running on Windows 10.

Table 1 Description of datasets

ID	*Dataset*	*Number of samples*	*Positive samples*	*Negative samples*	*Number of attributes*	*Ratio (%)*
1	Transfusion	748	178	570	5	31.23
2	Ecoli	336	77	259	8	29.73
3	Yeast	1,484	51	1,433	8	3.56
4	Abalone	4,177	103	4,074	8	2.53

4.1 Experiment 1

In this experiment, we ran experiments to evaluate our proposed fuzzy membership function. To do this, we applied our $f_{lin}^{cen\_2c}(x_i)$ function given in equation (10) to FSVM-CIL and compared the obtained results to those of FSVM-CIL used the fuzzy membership functions given in equations (4)–(9). For simplicity, we name FSVM-CIL using the same superscript and subscript of the fuzzy membership functions as shown in Table 2 and we call these FSVM-CILs the normal FSVM-CIL. For example, if FSVM-CIL uses the fuzzy linear function based on the distance to the own class centre, i.e., equations (4) and (5), then FSVM-CIL is named by FSVM-CIL$_{lin}^{cen}$. Table 3 shows the experimental results of FSVM-CIL using the fuzzy membership functions. The main results can be observed as follows:

1 For both *Transfusion* and *Ecoli* datasets, FSVM-CIL$_{lin}^{cen\_2c}$ gave higher SE, GM, F1S, ACC, and AUC than FSVM-CIL. Especially, FSVM-CIL$_{lin}^{cen\_2c}$ gave much higher SE, GM, and AUC than FSVM-CIL.

2 For the *Yeast* dataset, FSVM-CIL$_{lin}^{cen\_2c}$ gave higher SE, GM, and AUC than FSVM-CIL. Especially, FSVM-CIL$_{lin}^{cen\_2c}$ gave much higher SE (79.77%) than the best FSVM-CIL (41.50%).

3 For the *Abalone* dataset, FSVM-CIL$_{lin}^{cen\_2c}$ gave much higher SE, GM, F1S, and AUC than FSVM-CIL. It should be noted that in six fuzzy membership functions given in equations (4)–(9), only FSVM-CIL$_{lin}^{hyp}$ correctly classified some positive samples (SE = 19.57%), while FSVM-CIL$_{lin}^{cen\_2c}$ demonstrates that it classified positive samples much more correct than FSVM-CIL$_{lin}^{hyp}$ (SE = 61.48%).

Table 2 FSVM-CIL setting with methods of calculating fuzzy weights

ID	*FSVM-CIL setting*	m_i^+	m_i^-
1	FSVM-CIL$_{lin}^{cen}$	$f_{lin}^{cen}(x_i^+)$	$f_{lin}^{cen}(x_i^-) * r$
2	FSVM-CIL$_{exp}^{cen}$	$f_{exp}^{cen}(x_i^+)$	$f_{exp}^{cen}(x_i^-) * r$
3	FSVM-CIL$_{lin}^{shp}$	$f_{lin}^{shp}(x_i^+)$	$f_{lin}^{shp}(x_i^-) * r$
4	FSVM-CIL$_{exp}^{shp}$	$f_{exp}^{shp}(x_i^+)$	$f_{exp}^{shp}(x_i^-) * r$
5	FSVM-CIL$_{lin}^{hyp}$	$f_{lin}^{hyp}(x_i^+)$	$f_{lin}^{hyp}(x_i^-) * r$
6	FSVM-CIL$_{exp}^{hyp}$	$f_{exp}^{hyp}(x_i^+)$	$f_{exp}^{hyp}(x_i^-) * r$
7	FSVM-CIL$_{lin}^{cen\_2c}$	$f_{lin}^{cen\_2c}(x_i^+)$	$f_{lin}^{cen\_2c}(x_i^-) * r$

Table 3 Classification results of FSVM-CIL and FSVM-CIL using our fuzzy membership function

Dataset	*FSVM-CIL method*	*SP (%)*	*SE (%)*	*GM (%)*	*F1S (%)*	*ACC (%)*	*AUC (%)*
Transfusion	FSVM-CIL$^{cen}_{lin}$	90.46	27.33	39.81	29.44	75.46	58.89
	FSVM-CIL$^{cen}_{exp}$	89.26	30.73	45.97	33.85	75.35	60.00
	FSVM-CIL$^{shp}_{lin}$	90.63	26.29	37.74	27.87	75.35	58.46
	FSVM-CIL$^{shp}_{exp}$	*91.93*	20.80	30.10	21.88	75.03	56.36
	FSVM-CIL$^{hyp}_{lin}$	85.33	34.60	39.05	29.08	73.32	59.97
	FSVM-CIL$^{hyp}_{exp}$	89.19	32.99	47.62	35.72	75.83	61.09
	FSVM-CIL$^{cen\_2c}_{lin}$	88.14	*38.96*	*54.30*	*41.53*	*76.45*	*63.55*
Ecoli	FSVM-CIL$^{cen}_{lin}$	92.21	76.88	83.88	75.58	88.69	84.55
	FSVM-CIL$^{cen}_{exp}$	91.89	75.33	82.89	74.19	88.09	83.61
	FSVM-CIL$^{shp}_{lin}$	92.13	76.60	83.74	75.32	88.57	84.36
	FSVM-CIL$^{shp}_{exp}$	92.13	76.08	83.39	75.02	88.45	84.11
	FSVM-CIL$^{hyp}_{lin}$	92.59	77.15	84.23	76.18	89.05	84.87
	FSVM-CIL$^{hyp}_{exp}$	92.13	77.15	84.05	75.67	88.69	84.64
	FSVM-CIL$^{cen\_2c}_{lin}$	*92.67*	*77.38*	*84.42*	*76.49*	*89.17*	*85.03*
Yeast	FSVM-CIL$^{cen}_{lin}$	97.49	37.45	57.86	34.73	95.42	67.47
	FSVM-CIL$^{cen}_{exp}$	97.31	38.27	58.26	34.62	95.28	67.79
	FSVM-CIL$^{shp}_{lin}$	97.63	35.45	54.90	32.91	95.48	66.54
	FSVM-CIL$^{shp}_{exp}$	*98.36*	29.41	42.44	26.98	*95.99*	63.88
	FSVM-CIL$^{hyp}_{lin}$	89.08	39.68	58.11	31.17	87.40	64.38
	FSVM-CIL$^{hyp}_{exp}$	96.68	41.50	62.46	*35.75*	94.79	69.09
	FSVM-CIL$^{cen\_2c}_{lin}$	59.65	*79.77*	*65.09*	14.55	60.35	*69.71*
Abalone	FSVM-CIL$^{cen}_{lin}$	100	0.00	0.00	0.00	97.53	50.00
	FSVM-CIL$^{cen}_{exp}$	100	0.00	0.00	0.00	97.53	50.00
	FSVM-CIL$^{shp}_{lin}$	100	0.00	0.00	0.00	97.53	50.00
	FSVM-CIL$^{shp}_{exp}$	100	0.00	0.00	0.00	97.53	50.00
	FSVM-CIL$^{hyp}_{lin}$	61.10	19.57	33.79	2.34	60.07	40.33
	FSVM-CIL$^{hyp}_{exp}$	100	0.00	0.00	0.00	97.53	50.00
	FSVM-CIL$^{cen\_2c}_{lin}$	56.65	*61.48*	*52.91*	*6.51*	56.76	*59.06*

The experimental results showed that when we applied our fuzzy membership function $f^{cen\_2c}_{lin}(x_i)$ to FSVM-CIL, FSVM-CIL$^{cen\_2c}_{lin}$ is more efficient than FSVM-CIL for the experimental datasets. Moreover, when the imbalanced ratio increases, FSVM-CIL$^{cen\_2c}_{lin}$ exhibits a better performance than FSVM-CIL in correctly classifying the positive samples in the dataset since FSVM-CIL$^{cen\_2c}_{lin}$ found much higher values of SE and GM than FSVM-CIL. This is because our $f^{cen\_2c}_{lin}(x_i)$ function is designed based on the distance from x_i to the centre of two classes, while the fuzzy membership functions in FSVM-CIL only consider the distance from x_i to its class centre.

4.2 Experiment 2

In this experiment, we ran experiments to evaluate our method for adjusting fuzzy weight values. To do this, we applied our method to FSVM-CIL and compared the obtained results to those of the normal FSVM-CIL. We set the maximum number of iterations to adjust the set of fuzzy weights $\{m^+, m^-\}$ to $T = 20$. Table 4 shows our experimental results, where $l \in [1, T]$ is a value when GM is maximum. We see that when we applied our method of adjusting fuzzy weights to FSVM-CIL, FSVM-CIL gave slightly higher SE, GM, F1S, ACC, and AUC than the normal FSVM-CIL for the *Ecoli* dataset. However, when we applied our method of adjusting fuzzy weights to FSVM-CIL, FSVM-CIL gave much higher SE, GM, F1S, ACC, and AUC than the normal FSVM-CIL for *Transfusion*, *Yeast*, and *Abalone* datasets.

This shows that our method of adjusting fuzzy weight values is efficient when it is applied to the normal FSVM-CIL. This is due to the fact that our method iteratively adjusts the fuzzy weights of sensitive samples on boundaries and positive/negative label noises based on TLPs, while the normal FSVM-CIL only computes and uses initial fuzzy weights for all samples in the training dataset.

Table 4 Classification results of FSVM-CIL and FSVM-CIL using our adjusting fuzzy weights

Dataset	*FSVM-CIL*	*Adjusted*	*SP*	*SE*	*GM*	*F1S*	*ACC*	*AUC*
	method	*method*	*(%)*	*(%)*	*(%)*	*(%)*	*(%)*	*(%)*
Transfusion	FSVM-CIL$^{cen}_{lin}$	None	90.46	27.33	39.81	29.44	75.46	58.89
		$l = 19$	83.51	*48.20*	*63.06*	*47.85*	75.11	*65.85*
	FSVM-CIL$^{cen}_{exp}$	None	89.26	30.73	45.97	33.85	75.35	60.00
		$l = 19$	84.21	*47.65*	*62.93*	*47.95*	*75.51*	*65.93*
	FSVM-CIL$^{shp}_{lin}$	None	90.63	26.29	37.74	27.87	75.35	58.46
		$l = 18$	83.33	*47.64*	*62.63*	*47.33*	74.84	*65.49*
	FSVM-CIL$^{shp}_{exp}$	None	91.93	20.80	30.10	21.88	75.03	56.36
		$l = 18$	83.37	*47.87*	*62.81*	*47.56*	74.93	*65.62*
	FSVM-CIL$^{hyp}_{lin}$	None	85.33	34.60	39.05	29.08	73.32	59.97
		$l = 11$	*86.46*	*35.65*	*55.04*	*39.63*	*74.36*	*61.06*
	FSVM-CIL$^{hyp}_{exp}$	None	89.19	32.99	47.62	35.72	75.83	61.09
		$l = 18$	83.68	*48.08*	*63.06*	*47.98*	75.22	*65.88*
Ecoli	FSVM-CIL$^{cen}_{lin}$	None	92.21	76.88	83.88	75.58	88.69	84.55
		$l = 4$	91.97	*77.63*	*84.21*	*75.72*	88.69	*84.80*
	FSVM-CIL$^{cen}_{exp}$	None	91.89	75.33	82.89	74.19	88.09	83.61
		$l = 2$	*92.05*	74.78	82.71	74.13	88.09	83.41
	FSVM-CIL$^{shp}_{lin}$	None	92.13	76.60	83.74	75.32	88.57	84.37
		$l = 7$	92.05	*77.37*	*84.13*	*75.69*	*88.69*	*84.71*
	FSVM-CIL$^{shp}_{exp}$	None	92.13	76.08	83.39	75.02	88.45	84.11
		$l = 4$	92.05	*76.35*	*83.53*	*75.09*	88.45	*84.20*
	FSVM-CIL$^{hyp}_{lin}$	None	92.59	77.15	84.23	76.18	89.05	84.87
		$l = 7$	*93.21*	*76.83*	*84.34*	*76.65*	*89.46*	*85.02*
	FSVM-CIL$^{hyp}_{exp}$	None	92.13	77.15	84.05	75.67	88.69	84.64
		$l = 4$	91.97	*78.18*	*84.51*	*75.99*	*88.81*	*85.08*

Table 4 Classification results of FSVM-CIL and FSVM-CIL using our adjusting fuzzy weights (continued)

Dataset	*FSVM-CIL method*	*Adjusted method*	*SP (%)*	*SE (%)*	*GM (%)*	*F1S (%)*	*ACC (%)*	*AUC (%)*
Yeast	FSVM-CIL$_{lin}^{cen}$	None	97.49	37.45	57.86	34.73	95.42	67.47
		$l = 6$	97.09	*41.64*	*62.56*	*37.12*	95.18	*69.36*
	FSVM-CIL$_{exp}^{cen}$	None	97.31	38.27	58.26	34.62	95.28	67.79
		$l = 2$	97.19	*42.45*	*62.98*	*38.01*	*95.32*	*69.82*
	FSVM-CIL$_{lin}^{shp}$	None	97.63	35.45	54.9	32.91	95.48	66.54
		$l = 9$	97.37	*42.14*	*63.32*	*39.07*	95.47	*69.75*
	FSVM-CIL$_{exp}^{shp}$	None	98.36	29.41	42.44	26.98	95.99	63.88
		$l = 7$	97.87	*38.18*	*58.22*	*37.18*	95.82	*68.03*
	FSVM-CIL$_{lin}^{hyp}$	None	89.08	39.68	58.11	31.17	87.40	64.38
		$l = 1$	*89.62*	*42.41*	*60.00*	*31.54*	*88.02*	*66.02*
	FSVM-CIL$_{exp}^{hyp}$	None	96.68	41.5	62.46	35.75	94.79	69.09
		$l = 4$	*96.75*	*42.55*	*63.25*	*36.74*	*94.9*	*69.65*
Abalone	FSVM-CIL$_{lin}^{cen}$	None	100	0.00	0.00	0.00	97.53	50.00
		$l = 5$	100	0.00	0.00	0.00	97.53	50.00
	FSVM-CIL$_{exp}^{cen}$	None	100	0.00	0.00	0.00	97.53	50.00
		$l = 5$	100	0.00	0.00	0.00	97.53	50.00
	FSVM-CIL$_{lin}^{shp}$	None	100	0.00	0.00	0.00	97.53	50.00
		$l = 5$	100	0.00	0.00	0.00	97.53	50.00
	FSVM-CIL$_{exp}^{shp}$	None	100	0.00	0.00	0.00	97.53	50.00
		$l = 10$	100	0.00	0.00	0.00	97.53	50.00
	FSVM-CIL$_{lin}^{hyp}$	None	61.1	19.57	33.79	2.34	60.07	40.33
		$l = 5$	59.82	*49.43*	*53.89*	*6.03*	59.56	*54.62*
	FSVM-CIL$_{exp}^{hyp}$	None	100	0.00	0.00	0.00	97.53	50.00
		$l = 4$	100	0.00	0.00	0.00	97.53	50.00

4.3 Experiment 3

In this experiment, we evaluated the efficiency of our AFW-CIL. To do this, we compared the experimental results found by AFW-CIL with those found by FSVM-CIL as shown in Table 5. The main results observed on the experimental results are as follows. For *Transfusion* dataset, AFW-CIL gave much higher SE, GM, F1S, and AUC than FSVM-CIL. For *Ecoli* dataset, AFW-CIL gave higher all the SP, SE, GM, F1S, ACC, and AUC than FSVM-CIL. For *Yeast* dataset, AFW-CIL gave much higher SE, GM, and AUC than FSVM-CIL. For *Abalone* dataset, AFW-CIL gave much higher SE, GM, F1S, and AUC than FSVM-CIL. As we mentioned before, for the two-class imbalanced learning problem, SE, GM, and AUC are the three most important measures to evaluate the efficiency of a classification algorithm. Therefore, we can see that:

1 For the *Transfusion* dataset, the best FSVM-CIL gave the highest SE, GM, and AUC at 34.60%, 47.62%, and 61.09%, respectively, while AFW-CIL gave SE, GM, and AUC at 47.64%, 62.62%, and 65.50%, respectively. This shows that AFW-CIL gave 13.04%, 15.00%, and 4.41% of SE, GM, and AUC, respectively, higher than the best FSVM-CIL, i.e., AFW-CIL is much more efficient than FSVM-CIL.

2 For the *Ecoli* dataset, the best FSVM-CIL gave the highest SE, GM, and AUC at 77.15%, 84.23%, and 84.87%, respectively, while AFW-CIL gave SE, GM, and AUC at 78.42%, 84.93%, and 85.51%, respectively. This means that AFW-CIL classified positive samples more correct than FSVM-CIL.

3 For the *Yeast* dataset, the best FSVM-CIL gave the highest SE, GM, and AUC at 41.50%, 62.46%, and 69.09%, respectively, while AFW-CIL gave SE, GM, and AUC at 85.14%, 74.15%, and 75.93%, respectively. This means that AFW-CIL outperforms FSVM-CIL in correctly classifying positive samples.

4 For the *Abalone* dataset, only FSVM-CIL$^{hyp}_{lin}$ found SE and GM at 19.57% and 33.79%, respectively, while AFW-CIL gave SE, GM, and AUC at 65.10%, 59.88%, and 60.17%, respectively. This shows that AFW-CIL gave 44.53%, 26.09%, and 19.84% of SE, GM, and AUC, respectively, much higher than the best FSVM-CIL, i.e., AFW-CIL is better than FSVM-CIL in terms of correctly classifying positive samples.

Table 5 Classification results of FSVM-CIL and AFW-CIL

Dataset	*Learning method*	*SP* *(%)*	*SE* *(%)*	*GM* *(%)*	*F1S* *(%)*	*ACC* *(%)*	*AUC* *(%)*
Transfusion	FSVM-CIL$^{cen}_{lin}$	90.46	27.33	39.81	29.44	75.46	58.89
	FSVM-CIL$^{cen}_{exp}$	89.26	30.73	45.97	33.85	75.35	60.00
	FSVM-CIL$^{shp}_{lin}$	90.63	26.29	37.74	27.87	75.35	58.46
	FSVM-CIL$^{shp}_{exp}$	*91.93*	20.80	30.10	21.88	75.03	56.36
	FSVM-CIL$^{hyp}_{lin}$	85.33	34.60	39.05	29.08	73.32	59.97
	FSVM-CIL$^{hyp}_{exp}$	89.19	32.99	47.62	35.72	*75.83*	61.09
	AFW-CIL ($l = 16$)	83.37	*47.64*	*62.62*	*47.25*	74.87	*65.50*
Ecoli	FSVM-CIL$^{cen}_{lin}$	92.21	76.88	83.88	75.58	88.69	84.55
	FSVM-CIL$^{cen}_{exp}$	91.89	75.33	82.89	74.19	88.09	83.61
	FSVM-CIL$^{shp}_{lin}$	92.13	76.60	83.74	75.32	88.57	84.37
	FSVM-CIL$^{shp}_{exp}$	92.13	76.08	83.39	75.02	88.45	84.11
	FSVM-CIL$^{hyp}_{lin}$	92.59	77.15	84.23	76.18	89.05	84.87
	FSVM-CIL$^{hyp}_{exp}$	92.13	77.15	84.05	75.67	88.69	84.64
	AFW-CIL ($l = 7$)	*92.59*	*78.42*	*84.93*	*76.95*	*89.35*	*85.51*
Yeast	FSVM-CIL$^{cen}_{lin}$	97.49	37.45	57.86	34.73	95.42	67.47
	FSVM-CIL$^{cen}_{exp}$	97.31	38.27	58.26	34.62	95.28	67.79
	FSVM-CIL$^{shp}_{lin}$	97.63	35.45	54.90	32.91	95.48	66.54
	FSVM-CIL$^{shp}_{exp}$	98.36	29.41	42.44	26.98	95.99	63.88
	FSVM-CIL$^{hyp}_{lin}$	89.08	39.68	58.11	31.17	87.40	64.38
	FSVM-CIL$^{hyp}_{exp}$	96.68	41.50	62.46	35.75	94.79	69.09
	AFW-CIL ($l = 2$)	66.73	*85.14*	*74.15*	17.14	67.37	*75.93*
Abalone	FSVM-CIL$^{cen}_{lin}$	100	0.00	0.00	0.00	97.53	50.00
	FSVM-CIL$^{cen}_{exp}$	100	0.00	0.00	0.00	97.53	50.00
	FSVM-CIL$^{shp}_{lin}$	100	0.00	0.00	0.00	97.53	50.00
	FSVM-CIL$^{shp}_{exp}$	100	0.00	0.00	0.00	97.53	50.00
	FSVM-CIL$^{hyp}_{lin}$	61.10	19.57	33.79	2.34	60.07	40.33
	FSVM-CIL$^{hyp}_{exp}$	100	0.00	0.00	0.00	97.53	50.00
	AFW-CIL ($l = 4$)	55.25	*65.10*	*59.88*	*6.73*	55.49	*60.17*

In summary, the results of the three above experiments show that:

1 when we apply our fuzzy membership function $f_{lin}^{cen\_2c}(x_i)$ to the normal FSVM-CIL, FSVM-CIL$_{lin}^{cen\_2c}$ classifies positive samples more correct than FSVM-CIL for all the experimental datasets, since FSVM-CIL$_{lin}^{cen\_2c}$ gave higher SE, GM, and AUC than the normal FSVM-CIL

2 when we apply our method of adjusting fuzzy weights to the normal FSVM-CIL, FSVM-CIL is efficient than the normal FSVM-CIL in terms of classifying positive samples, since FSVM-CIL gave much higher SE, GM, and AUC than the normal FSVM-CIL

3 when we combine our fuzzy membership function $f_{lin}^{cen\_2c}(x_i)$ and our method of adjusting fuzzy weights together in AFW-CIL, AFW-CIL outperforms FSVM-CIL for all the experimental datasets, since AFW-CIL gave much higher SE, GM, and AUC than the normal FSVM-CIL.

5 Conclusions

In this study, we proposed an efficient adaptive fuzzy weight algorithm for the two-class imbalanced learning problem, namely AFW-CIL. To do this, we first proposed a fuzzy membership function based on the distance from each sample to the centres of both minority and majority classes. We then proposed a method for adjusting fuzzy weight values based on the positions of the sensitive samples determined by Tomek links pairs in the feature space of samples. Initially, our algorithm finds a set of fuzzy weight values for samples based on the proposed fuzzy membership function. Then, our algorithm iteratively adjusts fuzzy weight values of the sensitive samples based on the proposed method of adjusting fuzzy weight values such that it increases the influence of minority samples and decreases the influence of majority samples in constructing a classifier model. Our experimental results on the datasets consisting of *Transfusion*, *Ecoli*, *Yeast*, and *Abalone* published in UCI Machine Learning Repository (Dua and Graff, 2017) show that AFW-CIL does not only outperform FSVM-CIL but also outperforms FSVM-CIL combined with either our fuzzy membership function or our method of adjusting fuzzy weights.

References

Akbani, R., Kwek, S. and Japkowicz, N. (2004) 'Applying support vector machines to imbalanced datasets', *Proceedings of the European Conference on Machine Learning (ECML2004)*, Springer, pp.39–50.

Batista, G.E.A.P.A., Prati, R.C. and Monard, M.C. (2004) 'A study of the behavior of several methods for balancing machine learning training data', *SIGKDD Explor. Newsl.*, Vol. 6, No. 1, pp.20–29.

Batuwita, R. and Palade, V. (2010) 'FSVM-CIL: fuzzy support vector machines for class imbalance learning', *IEEE Transactions on Fuzzy Systems*, Vol. 18, No. 3, pp.558–571.

Chawla, N., Lazarevic, A., Hall, L. and Bowyer, K. (2003) 'SMOTEBoost: improving prediction of the minority class in boosting', *Proceedings of the 7th European Conference on Principles and Practice of Knowledge Discovery in Database*, Vol. 2838, No. 1, pp.107–119.

Chawla, N.V., Bowyer, K.W., Hall, L.O. and Kegelmeyer, W.P. (2002) 'SMOTE: synthetic minority over-sampling technique', *Journal of Artificial Intelligence Research*, Vol. 16, pp.321–357.

Dua, D. and Graff, C. (2017) *UCI Machine Learning Repository* [online] http://archive.ics.uci.edu/ml (accessed 15 October 2022).

Elkan, C. (2001) 'The foundations of cost-sensitive learning', *Proceedings of the Seventeenth International Conference on Artificial Intelligence*, 4–10 August, Seattle, Vol. 1, pp.973–978.

Fernández, A., García, S., Galar, M., Prati, R.C., Krawczyk, B. and Herrera, F. (2018) *Learning from Imbalanced Data Sets*, Vol. 10, Springer, Berlin, Germany.

Han, H., Wang, W-Y. and Mao, B-H. (2005) 'Borderline-smote: a new over-sampling method in imbalanced data sets learning', in Huang, D-S., Zhang, X-P. and Huang, G-B. (Eds.): *Advances in Intelligent Computing*, pp.878–887, Springer Berlin Heidelberg, Berlin, Heidelberg.

Hao, J., Luo, S. and Pan, L. (2022) 'Rule extraction from biased random forest and fuzzy support vector machine for early diagnosis of diabetes', *Scientific Reports*, Vol. 12, No. 9858, pp.1–11.

He, H., Bai, Y., Garcia, E.A. and Li, S. (2008) 'ADASYN: adaptive synthetic sampling approach for imbalanced learning', *Proceedings of the 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, pp.1322–1328.

Japkowicz, N. and Stephen, S. (2002) 'The class imbalance problem: a systematic study', *Intelligent Data Analysis*, Vol. 6, No. 5, pp.429–449.

Jordan, M.I. and Mitchell, T.M. (2015) 'Machine learning: trends, perspectives, and prospects', *Science*, Vol. 349, No. 6245, pp.255–260.

Kubat, M. (2000) 'Addressing the curse of imbalanced training sets: one-sided selection', *Fourteenth International Conference on Machine Learning*.

Kubat, M., Holte, R.C. and Matwin, S. (1998) 'Machine learning for the detection of oil spills in satellite radar images', *Machine Learning*, Vol. 30, No. 2, pp.195–215.

Laurikkala, J. (2001) 'Improving identification of difficult small classes by balancing class distribution', in Quaglini, S., Barahona, P. and Andreassen, S. (Eds.): *Artificial Intelligence in Medicine*, pp.63–66, Springer Berlin Heidelberg, Berlin, Heidelberg.

Lin, C-F. and Wang, S-D. (2002) 'Fuzzy support vector machines', *IEEE Transactions on Neural Networks*, Vol. 13, No. 2, pp.464–471.

Liu, J. (2021) 'Fuzzy support vector machine for imbalanced data with borderline noise', *Fuzzy Sets and Systems*, Vol. 413, No. 1, pp.64–73.

Liu, X-Y., Wu, J. and Zhou, Z-H. (2008) 'Exploratory undersampling for class-imbalance learning', *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, Vol. 39, No. 2, pp.539–550.

Ma, T., Hou, Y., Chen, X. and Cheng, J. (2018) 'A novel method combining fuzzy SVM and sampling for imbalanced classification', *International Journal of Applied Systemic Studies*, Vol. 8, No. 1, p.1.

Ning, Q., Zhao, X. and Ma, Z. (2022) 'A novel method for identification of glutarylation sites combining borderline-SMOTE with Tomek links technique in imbalanced data', *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, Vol. 19, No. 5, pp.2632–2641.

Quang, V.D., Khang, T.D. and Huy, N.M. (2021) 'Improving AdaBoost algorithm with weighted SVM for imbalanced data classification', *Proceedings of the International Conference on Future Data and Security Engineering*, Springer, pp.125–136.

Rekha, G., Reddy, V. and Tyagi, A. (2020) 'An earth mover's distance-based undersampling approach for handling class-imbalanced data', *International Journal of Intelligent Information and Database Systems*, Vol. 13, Nos. 2–4, p.376.

Sun, Y., Kamel, M.S., Wong, A.K. and Wang, Y. (2007) 'Cost-sensitive boosting for classification of imbalanced data', *Pattern Recognition*, Vol. 40, No. 12, pp.3358–3378.

Tomek, I. (1976) 'Two modifications of CNN', *IEEE Transactions on Systems Man and Communications*, Vol. 6, No. 1, pp.769–772.

Xu, Z., Shen, D., Nie, T. and Kou, Y. (2020) 'A hybrid sampling algorithm combining M-SMOTE and ENN based on random forest for medical imbalanced data', *Journal of Biomedical Informatics*, Vol. 107, No. 103465, pp.1–11.

Yang, X., Song, Q. and Cao, A. (2005) 'Weighted support vector machine for data classification', *Proceedings of the 2005 IEEE International Joint Conference on Neural Networks*, IEEE, Vol. 2, pp.859–864.

Zeng, M., Zou, B., Wei, F., Liu, X. and Wang, L. (2016) 'Effective prediction of three common diseases by combining SMOTE with Tomek links technique for imbalanced medical data', *Proceedings of the 2016 IEEE International Conference of Online Analysis and Computing Science (ICOACS)*, IEEE, pp.225–228.