FSVM-CIL: Fuzzy Support Vector Machines for Class Imbalance Learning

Rukshan Batuwita and Vasile Palade

Abstract—**Support vector machines (SVMs) is a popular machine learning technique, which works effectively with balanced datasets. However, when it comes to imbalanced datasets, SVMs produce suboptimal classification models. On the other hand, the SVM algorithm is sensitive to outliers and noise present in the datasets. Therefore, although the existing class imbalance learning (CIL) methods can make SVMs less sensitive to class imbalance, they can still suffer from the problem of outliers and noise. Fuzzy SVMs (FSVMs) is a variant of the SVM algorithm, which has been proposed to handle the problem of outliers and noise. In FSVMs, training examples are assigned different fuzzy-membership values based on their importance, and these membership values are incorporated into the SVM learning algorithm to make it less sensitive to outliers and noise. However, like the normal SVM algorithm, FSVMs can also suffer from the problem of class imbalance. In this paper, we present a method to improve FSVMs for CIL (called FSVM-CIL), which can be used to handle the class imbalance problem in the presence of outliers and noise. We thoroughly evaluated the proposed FSVM-CIL method on ten real-world imbalanced datasets and compared its performance with five existing CIL methods, which are available for normal SVM training. Based on the overall results, we can conclude that the proposed FSVM-CIL method is a very effective method for CIL, especially in the presence of outliers and noise in datasets.**

Index Terms—**Class imbalance learning (CIL), fuzzy support vector machines (FSVMs), outliers, support vector machines (SVMs).**

I. Introduction

SUPPORT Vector machines (SVMs) [1]–[3] is a widely used machine learning technique, which has been successfully applied to many real-world classification problems in various domains. Due to its solid mathematical background, high generalization capability and ability to find global classification solutions, SVMs are usually preferred by many researchers over other available classification paradigms.

Although the SVM algorithm works effectively with balanced datasets, when it comes to imbalanced datasets, it could often produce suboptimal results [4]–[14], i.e., an SVM classifier, which is trained with an imbalanced dataset, can produce a model that is biased toward the majority class and has a low performance on the minority class. In this paper, we consider the binary classification problem, and the positive class is treated as the minority class, while the negative class is treated as the

majority class, as in most of the real-world imbalanced datasets. Generally, a classifier should perform well with respect to both positive and negative classes for it to be useful in real-world application. There exist techniques to develop better performing classifiers with imbalanced datasets, which are generally called class imbalance learning (CIL) methods. These methods can be broadly divided into two categories, namely, external methods and internal methods. External methods involve preprocessing of training datasets in order to make them balanced, while internal methods deal with modifications of the learning algorithms in order to reduce their sensitiveness to class imbalance. The available methods for training SVMs with imbalanced datasets are discussed later in the paper.

On the other hand, since the SVM learning algorithm considers all the training examples uniformly, it is sensitive to outliers and noise present in datasets [15], [16]. Most of the real-world datasets usually contain some outliers and noisy examples. Therefore, although the existing imbalance-learning methods applied for normal SVMs can solve the problem of class imbalance, they can still suffer from the problem of outliers and noise. Fuzzy support vector machines (FSVMs) is a variant of the SVM learning algorithm, which was originally proposed in [17] in order to handle the problem of outliers and noise. The FSVM technique first assigns different fuzzy-membership values (weights) for different training examples to reflect their importance and then incorporates these membership values in the SVM learning algorithm to reduce the effect of outliers/noise when finding the separating hyperplane. However, the FSVM technique, like the normal SVM method, can still be sensitive to the class imbalance problem. We will show that this, indeed, is the case with the empirical results obtained in this study.

In this paper, we propose a method to improve FSVMs for CIL, which can be used to address both the problems of class imbalance and outliers/noise. In the proposed method, we assign fuzzy-membership values for training examples in order to reduce the effect of both of these problems under the principle of cost-sensitive learning. We name this method FSVMs for class imbalance learning (FSVM-CIL). We systematically evaluated the FSVM-CIL method on ten real-world imbalanced datasets and compared its performance with five different existing internal- and external-imbalance-learning methods applied for normal SVMs. We found that the proposed FSVM-CIL method is a very effective method for CIL in the presence of outliers/noise in the datasets.

This paper is organized as follows. Section II briefly reviews the SVM learning theory, and Section III discusses the existing class imbalance solutions that are available for SVM training. In Section IV, we present the FSVM algorithm, and

in Section V, we discuss the proposed method of using the FSVM technique for CIL. Section VI presents the imbalanced datasets used to validate the proposed method, while Section VII discusses the SVM/FSVM training methods used in this study, such as the parameter-selection and the performance-evaluation methods. In Section VIII, we present and discuss, in detail, the classification results obtained by the proposed method and compare them with the results obtained by different existing imbalance-learning methods. In Section IX, we apply a statistical test to compare the performance of the proposed method with the existing methods considered, and finally, in Section X, we conclude the paper.

II. SUPPORT VECTOR MACHINES LEARNING THEORY

In this section, we briefly review the learning algorithm of SVMs, which has been initially proposed in [1], [2]. Consider we have a binary classification problem, which is represented by a dataset $\{(x_1, y_1), (x_2, y_2), \ldots, (x_l, y_l)\}$, where $x_i \in \Re^n$ represents an n-dimensional data point, and $y_i \in \{-1, 1\}$ represents the class of that data point, for $i = 1, \ldots, l$. The goal of the SVM learning algorithm is to find a separating hyperplane that separates these data points into two classes. In order to find a better separation of classes, the data are first transformed into a higher dimensional feature space by a mapping function Φ. Then, a possible separating hyperplane, which resides in the higher dimensional feature space, can be represented by

$$w \cdot \Phi(x) + b = 0. \qquad (1)$$

If the dataset is completely linearly separable, the separating hyperplane with the maximum margin can be found by solving the following maximal-margin optimization problem:

$$\mathrm{Min}\left(\frac{1}{2}w \cdot w\right)$$
$$\mathrm{s.t.}\, y_i(w \cdot \Phi(x_i) + b) \geq 1$$
$$i = 1, \ldots, l. \qquad (2)$$

However, in most real-world problems, the datasets are not completely linearly separable, although they are mapped into a higher dimensional feature space. Therefore, the constrains in the aforementioned optimization problem in (2) are relaxed by introducing a slack variable $\varepsilon_i \geq 0$, and then, the soft-margin optimization problem is formulated as follows:

$$\mathrm{Min}\left(\frac{1}{2}w \cdot w + C\sum_{i=1}^{l}\varepsilon_i\right)$$
$$\mathrm{s.t.}\, y_i(w \cdot \Phi(x_i) + b) \geq 1 - \varepsilon_i$$
$$\varepsilon_i \geq 0, \qquad i = 1, \ldots, l. \qquad (3)$$

The slack variables $\varepsilon_i > 0$ hold for misclassified examples, and therefore, $\sum_{i=1}^{l}\varepsilon_i$ can be thought of as a measure of the amount of misclassifications. This new objective function in (3) has two goals. One is to maximize the margin, and the other one is to minimize the number of misclassifications. The parameter C controls the tradeoff between these two goals, and it can also be treated as the misclassification cost of a training

example. This quadratic-optimization problem can be solved by constructing a Lagrangian representation and transforming it into the following dual problem:

$$\mathrm{Max}\, W(\alpha) = \sum_{i=1}^{l}\alpha_i - \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}\alpha_i\alpha_j y_i y_j \Phi(x_i) \cdot \Phi(x_j)$$
$$\mathrm{s.t.}\, \sum_{i=1}^{l} y_i\alpha_i = 0, \qquad 0 \leq \alpha_i \leq C, \qquad i = 1, \ldots, l \qquad (4)$$

where α_is are Lagrange multipliers, which should satisfy the following Karush–Kuhn–Tucker (KKT) conditions:

$$\alpha_i(y_i(w \cdot \Phi(x_i) + b) - 1 + \varepsilon_i) = 0, \qquad i = 1, \ldots, l \quad (5)$$
$$(C - \alpha_i)\xi_i = 0, \qquad i = 1, \ldots, l. \qquad (6)$$

An important property of SVMs is that it is not necessary to know the mapping function $\Phi(x)$ explicitly. By applying a kernel function, such that $K(x_i, x_j) = \Phi(x_i) \cdot \Phi(x_j)$, we would be able to transform the dual-optimization problem in (4) into

$$\mathrm{Max}\, W(\alpha) = \sum_{i=1}^{l}\alpha_i - \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}\alpha_i\alpha_j y_i y_j K(x_i, x_j)$$
$$\mathrm{s.t.}\, \sum_{i=1}^{l} y_i\alpha_i = 0, \qquad 0 \leq \alpha_i \leq C, \qquad i = 1, \ldots, l. \quad (7)$$

By solving (7) and finding the optimal values for α_i, w can be recovered as follows:

$$w = \sum_{i=1}^{l}\alpha_i y_i \Phi(x_i) \qquad (8)$$

and b can be determined from the KKT conditions in (5). The data points having nonzero α_i values are called support vectors. Finally, the SVM decision function is given by

$$f(x) = \mathrm{sign}(w \cdot \Phi(x) + b) = \mathrm{sign}\left(\sum_{i=1}^{l}\alpha_i y_i K(x_i, x) + b\right). \qquad (9)$$

III. CLASS IMBALANCE LEARNING METHODS AVAILABLE FOR SUPPORT VECTOR MACHINES

It has been well-studied that the SVM algorithm can be sensitive to class imbalance [4]–[14], i.e., the separating hyperplane of an SVM model, which was developed with an imbalanced dataset, can be skewed toward the minority (positive) class [5], [6]. This skewness generally causes the generation of a high number of false-negative predictions, which lower the model's performance on the positive class compared with the performance on the negative (majority) class.

As mentioned earlier, there exist CIL methods that can be applied to reduce the effect of data imbalance, when developing SVM classifiers with imbalanced datasets. Generally, these methods can be divided into two categories: external methods and internal methods [14], [18], [19]. A comprehensive review of different CIL methods can be found in [14]. The following two

sections briefly discuss the external- and internal-imbalance-learning methods that can be applied for SVM training.

A. External Methods

The external methods are independent from the learning algorithm being used, and they involve preprocessing of the training datasets to balance them before training the classifiers. Different resampling methods, such as random and focused oversampling and undersampling, fall into to this category. In random undersampling, the majority-class examples are removed randomly, until a particular class ratio is met [18]. In random oversampling, the minority-class examples are randomly duplicated, until a particular class ratio is met [14]. Synthetic minority oversampling technique (SMOTE) [20] is an oversampling method, where new synthetic examples are generated in the neighborhood of the existing minority-class examples rather than directly duplicating them. In addition, several informed sampling methods have been introduced in [21]. A clustering-based sampling method has been proposed in [22], while a genetic algorithm-based sampling method has been proposed in [10].

Ensemble learning has also been applied to handle the problem of class imbalance [23]. SVM-based ensemble systems for imbalanced dataset learning have been developed in [11]–[13]. Moreover, different boosting methods have also been adopted for CIL in [24] and [25].

B. Internal Methods

Internal methods deal with modifications of a learning algorithm to make it less sensitive for the class imbalance. Algorithmic techniques have been developed for the different classification algorithms, such as neural networks [26], decision trees [27], fuzzy systems [28], [29] etc., for imbalanced dataset learning. For SVMs, several algorithmic solutions have been proposed in the literature. Veropoulos *et al.* [4] has proposed a method called different error costs (DEC), where the SVM objective function has been modified to assign two misclassification cost values C^+ and C^- as follows:

$$\text{Min}\left(\frac{1}{2}w \cdot w + C^+ \sum_{[i|y_i=+1]}^{l} \varepsilon_i + C^- \sum_{[i|y_i=-1]}^{l} \varepsilon_i\right)$$
$$\text{s.t. } y_i(w \cdot \Phi(x_i) + b) \geq 1 - \varepsilon_i$$
$$\varepsilon_i \geq 0, \qquad i = 1, \ldots, l \tag{10}$$

where C^+ is the misclassification cost for the positive-class examples, while C^- is the misclassification cost for the negative-class examples. By assigning a higher misclassification cost for the positive (minority) class examples than the negative (majority) class examples (i.e., $C^+ > C^-$), the effect of class imbalance could be reduced.

zSVM is another algorithmic modification, which is proposed for SVMs in [9] for learning-imbalance datasets. In this method, first, an SVM model is developed by using the available imbalanced training dataset. Then, the decision boundary of the resulted model is modified to remove the classifier's bias towards the majority (negative) class. Consider the SVM decision function given in (9), which can be rewritten as follows:

$$f(x) = \text{sign}\left(\sum_{i=1}^{l} \alpha_i y_i K(x_i, x) + b\right)$$
$$= \text{sign}\left\{\sum_{i=1}^{l_1} \alpha_i^+ y_i K(x_i, x) + \sum_{j=1}^{l_2} \alpha_j^- y_j K(x_j, x) + b\right\}$$
$$\tag{11}$$

where α_i^+ are the coefficients of the positive-support vectors, α_i^- are the coefficients of the negative-support vectors, and l_1 and l_2 represent the number of positive- and negative-training examples, respectively. In the zSVM method, the magnitude of the α_i^+ values of the positive-support vectors are increased by multiplying all of them by a particular small positive value z. Then, the modified SVM decision function can be represented as follows:

$$f(x) = \text{sign}\left\{z\sum_{i=1}^{l_1} \alpha_i^+ y_i K(x_i, x) + \sum_{j=1}^{l_2} \alpha_i^- y_j K(x_j, x) + b\right\}.$$
$$\tag{12}$$

This modification of α_i^+ would increase the weight of the positive-support vectors in the decision function, and therefore, it would decrease its bias toward the majority negative class. In [9], the value of z, which gives the best prediction for the training dataset, was selected as the optimal value of z to be used in (12).

In addition to these methods, several kernel-modification methods for training SVMs with imbalanced datasets have been proposed in [6] and [7].

IV. FUZZY SUPPORT VECTOR MACHINES

There has been some significant research carried out towards the combination of fuzzy logic and SVMs in different ways, and the term "Fuzzy SVMs" has been used to refer to most of these different approaches. Lin and Wang [17] applies a fuzzy-membership value for each training example, which is based on the importance of that example of its class, and reformulates the SVM learning algorithm such that different input points can make different contributions when finding the separating hyperplane. Wang *et al.* [30] extended this approach by introducing two membership values for each training example, which define the memberships of both positive and negative classes. This approach has been further extended, which is based on the concept of "vague sets" in [31]. Another type of fuzzy SVM method, which uses a special type of kernel function constructed from fuzzy-basis functions, has been proposed in [32]. A method for fuzzy support vector regression has been proposed in [33].

In contrast, several other approaches have been developed to assign fuzzy-membership values for the outputs generated by SVMs together with the output class. Xie *et al.* [34] use the decision value, which is generated by the SVMs, to define a membership for the output class, while [35] uses the fuzzy-output decision values for multiclass classification. Mill and Inoue [36] propose a method that uses the strengths (α_i) of

positive- and negative-support vectors to generate the fuzzy-membership values for the output classes. On the other hand, several methods have been proposed in [37]–[40] to extract fuzzy rules from the trained SVM models.

In this paper, we consider the Fuzzy-SVM method, which was initially proposed in [17], as a solution for the problem of outliers and noise. Although, fuzzy-membership functions are used, this method is different from general fuzzy-classification methods that usually involve fuzzification, defuzzification, and fuzzy reasoning. Generally, fuzzy-classification methods are based on fuzzy-rule-based systems [41], neurofuzzy systems [42], [43], and fuzzy measures [44]. Other fuzzy-classification methods handle fuzzy attribute values and/or fuzzy targets [45]. On the other hand, the FSVM method considered in this paper uses fuzzy-membership functions to define a different misclassification cost for each training pattern, as discussed below.

The normal SVM-learning algorithm considers all the training points uniformly, and therefore, it can be sensitive to outliers and noise [15], [16]. As mentioned earlier, the FSVM method proposed in [17] assigns different fuzzy-membership values m_i (or weights) for different examples to reflect their importance for their own class, where more important examples are assigned higher membership values, while the less-important ones, such as outliers and noise, are assigned lower membership values. Then, the SVM soft-margin optimization problem is reformulated as follows:

$$\text{Min}\left(\frac{1}{2}w \cdot w + C\sum_{i=1}^{l} m_i \varepsilon_i\right)$$

$$\text{s.t. } y_i(w \cdot \Phi(x_i) + b) \geq 1 - \varepsilon_i$$

$$\varepsilon_i \geq 0, \qquad i = 1, \dots l. \tag{13}$$

In this formulation, the membership m_i of a data point x_i is incorporated into the objective function in (13), and therefore, a smaller m_i could reduce the effect of the parameter ξ_i in the objective function, where the corresponding x_i is treated as less important. In another view, if we consider C as the cost assigned for a misclassification, then each example is assigned with a different misclassification cost value $m_i C$, such that more important examples are assigned higher costs, while less important ones are assigned lower costs. Therefore, the SVM algorithm can find a more robust hyperplane by maximizing the margin by letting some misclassification of less-important examples, such as outliers and noise.

In order to solve the FSVM optimization problem, (13) is transformed into the following dual Lagrangian [17]:

$$\text{Max}W(\alpha) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l} \alpha_i \alpha_j y_i y_j K(x_i, x_j)$$

$$\text{s.t. } \sum_{i=1}^{l} y_i \alpha_i = 0, 0 \leq \alpha_i \leq m_i C, \qquad i = 1, \dots, l. \tag{14}$$

The only difference between the original SVM dual-optimization problem in (7) and the FSVM dual-optimization problem in (14) is the upper bound of the values of α_i. By solving this dual problem in (14) for optimal α_i, w and b can be recovered in the same way, as in the normal SVM learning algorithm. The same SVM decision function in (9) applies for FSVMs as well.

This FSVM algorithm has been successfully applied to reduce the effect of outliers and noise in different domains with different ways of assigning fuzzy-membership values (see [46]–[50]).

V. FSVM-CIL: FUZZY SUPPORT VECTOR MACHINES FOR CLASS IMBALANCE LEARNING

As stated earlier, the SVM algorithm is sensitive to both the problems of class imbalance and outliers/noise. Although the existing CIL methods available for SVMs, which were discussed in Section III, can overcome the problem of class imbalance, they can still be sensitive to the problem of outliers and noise. On the other hand, although the FSVM method considered in this paper can overcome the problem of outliers/noise, it can suffer from the problem of class imbalance, since there is no modification in the FSVM algorithm compared with the original SVM algorithm to make it less sensitive to class imbalance. We show that this is in fact the case with empirical evidence presented in Section VIII. In this section, we present an improvement for the FSVM learning method, which is called FSVM-CIL, to overcome the problem of class imbalance.

As we discussed in the previous section, in FSVMs, we can assign different membership values (or weights) for training examples to reflect their different importance. We also showed that this is similar to assign different misclassification costs Cm_i for different training examples. Therefore, in order to reduce the effect of class imbalance, we can assign higher membership values, or higher misclassification costs, for the minority-class examples, while we assign lower membership values, or lower misclassification costs, for the majority-class examples, as done in the DEC method discussed in Section III, i.e., if we assign a single membership value, say m^+, for all the minority-class examples and another single membership value, say m^-, for all the majority-class examples such that $m^+ > m^-$, this is exactly equivalent to the DEC method applied for normal SVMs, where $C^+ = m^+ C$, and $C^- = m^- C$. However, the DEC method applied for normal SVMs can suffer the problem of outliers/noise, because all the examples in a separate class are given equal importance.

In the proposed FSVM-CIL method, we assign the membership values for training examples in such a way to satisfy the following two goals:
1) to suppress the effect of between class imbalance
2) to reflect the within-class importance of different training examples in order to suppress the effect of outliers and noise.

Let m_i^+ represent the membership value of a positive-class example x_i^+, while m_i^- represents the membership of a negative-class example x_i^- in their own classes. In the proposed FSVM-CIL method, we define these membership functions as follows:

$$m_i^+ = f(x_i^+)r^+ \tag{15}$$

$$m_i^- = f(x_i^-)r^- \tag{16}$$

where $f(x_i)$ generates a value between 0 and 1, which reflects the importance of x_i in its own class. Moreover, we assign the values for r^+ and r^- in order to reflect the class imbalance, such that $r^+ > r^-$. Therefore, a positive-class example can take a membership value in the $[0, r^+]$ interval, while a negative-class example can take a membership value in the $[0, r^-]$ interval. This way of assigning membership values serves both of the aforementioned goals. Therefore, the proposed FSVM-CIL method can be used to handle both the problems of class imbalance and outliers/noise together.

A. Assigning Fuzzy-Membership Values

When assigning fuzzy memberships for training examples in the FSVM-CIL method, in order to reflect the class imbalance, we assign $r^+ = 1$, and $r^- = r$, where r is the minority-to-majority class ratio. This was following the findings reported in [5], where the optimal results for the DEC method could be obtained when C^-/C^+ equals to the minority-to-majority class ratio. According to this assignment of values, a positive-class example can take a membership value in the $[0, 1]$ interval, while a negative-class example can take a membership value in the $[0, r]$ interval, where $r < 1$.

In order to define the function $f(x_i)$ introduced in (15) and (16), which gives the within-class importance of a training example, we consider three different methods discussed in the following sections.

1) $f(x_i)$ is Based on the Distance from the Own Class Center: In this method, $f(x_i)$ is defined with respect to d_i^{cen}, which is the distance between x_i and its own class center, as previously proposed for FSVMs in [17]. The examples closer to the class center are treated as more informative and assigned higher $f(x_i)$ values, while the examples far away from the center are treated as outliers or noise and assigned lower $f(x_i)$ values. Here, we use two separate decaying functions of d_i^{cen} to define $f(x_i)$, which are represented by $f_{\text{lin}}^{\text{cen}}(x_i)$ and $f_{\text{exp}}^{\text{cen}}(x_i)$ as follows:

$$f_{\text{lin}}^{\text{cen}}(x_i) = 1 - \frac{d_i^{\text{cen}}}{\max(d_i^{\text{cen}}) + \Delta} \qquad (17)$$

is a linearly decaying function of d_i^{cen}. Δ is a small positive value, which is used to avoid the case, where $f_{\text{lin}}^{\text{cen}}(x_i)$ becomes zero.

$$f_{\text{exp}}^{\text{cen}}(x_i) = \frac{2}{1 + \exp(\beta d_i^{\text{cen}})}$$
$$\beta \in [0, 1] \qquad (18)$$

is an exponentially decaying function of d_i^{cen}, where β determines the steepness of the decay. $d_i^{\text{cen}} = \|x_i - \bar{x}\|^{1/2}$ is the Euclidean distance to x_i from its own class center $\bar{x}$.

2) $f(x_i)$ is Based on the Distance from the Estimated Hyperplane: In this method, $f(x_i)$ is defined based on d_i^{sph}, which is the distance to x_i from the estimated separating hyperplane, as introduced in [46]. Here, d_i^{sph} is estimated by the distance to x_i from the center of the "spherical region," which can be defined as a hypersphere covering the overlapping region of the two classes, where the separation hyperplane is more likely to pass through. Both linear- and exponential-decaying functions

are used to define the function $f(x_i)$, which are represented by $f_{\text{lin}}^{\text{sph}}(x_i)$ and $f_{\text{exp}}^{\text{sph}}(x_i)$ as follows:

$$f_{\text{lin}}^{\text{sph}}(x_i) = 1 - \frac{d_i^{\text{sph}}}{\max(d_i^{\text{sph}}) + \Delta} \qquad (19)$$

$$f_{\text{exp}}^{\text{sph}}(x_i) = \frac{2}{1 + \exp(\beta d_i^{\text{sph}})}$$
$$\beta \in [0, 1] \qquad (20)$$

where $d_i^{\text{sph}} = \|x_i - \bar{x}\|^{1/2}$, and $\bar{x}$ is the center of the spherical region, which is estimated by the center of the entire positive and negative dataset.

3) $f(x_i)$ is Based on the Distance from the Actual Hyperplane: In this method, we define $f(x_i)$ based on the distance from the actual separating hyperplane to x_i, which is found by training a conventional SVM model on the imbalanced dataset. The examples closer to the actual separating hyperplane are treated as more informative and assigned higher membership values, while the examples far away from the separating hyperplane are treated as less informative and assigned lower membership values. We do the followings in this method.

1) Train a normal SVM model with the original imbalanced dataset.
2) Find the functional margin d_i^{hyp} of each example x_i [given in (21)] (this is equivalent to the absolute value of the SVM decision value) with respect to the separating hyperplane found. The functional margin is proportional to the geometric margin of a training example with respect to the hyperplane.

$$d_i^{\text{hyp}} = y_i(w \cdot \Phi(x_i) + b). \qquad (21)$$

3) Consider both linear- and exponential-decaying functions to define $f(x_i)$ as follows:

$$f_{\text{lin}}^{\text{hyp}}(x_i) = 1 - \frac{d_i^{\text{hyp}}}{\max(d_i^{\text{hyp}}) + \Delta} \qquad (22)$$

$$f_{\text{exp}}^{\text{hyp}}(x_i) = \frac{2}{1 + \exp(\beta d_i^{\text{hyp}})}$$
$$\beta \in [0, 1]. \qquad (23)$$

B. Different FSVM-CIL Settings

Table I defines the different FSVM-CIL experimental settings, which are considered in this paper, according to the different ways of assigning membership values for positive- and negative-training examples, by following the methods earlier presented.

VI. DATASETS

We considered nine benchmark real-world imbalanced dataset from the UCI machine learning repository [51] to validate the proposed FSVM-CIL method. In addition to these UCI datasets, we considered an imbalanced bioinformatics dataset ("miRNA" dataset) from our previous research on human miRNA gene recognition [52], [53]. It is highly likely that

TABLE I
DIFFERENT FSVM-CIL SETTINGS CONSIDERED IN THIS RESEARCH
ACCORDING TO DIFFERENT WAYS OF ASSIGNING MEMBERSHIP VALUES FOR
POSITIVE (m_i^+) AND NEGATIVE (m_i^-) CLASS EXAMPLES

FSVM-CIL setting	m_i^+	m_i^-
$FSVM-CIL_{lin}^{cen}$	$f_{lin}^{cen}(x_i^+)$	$f_{lin}^{cen}(x_i^-)r$
$FSVM-CIL_{exp}^{cen}$	$f_{exp}^{cen}(x_i^+)$	$f_{exp}^{cen}(x_i^-)r$
$FSVM-CIL_{lin}^{sph}$	$f_{lin}^{sph}(x_i^+)$	$f_{lin}^{sph}(x_i^-)r$
$FSVM-CIL_{exp}^{sph}$	$f_{exp}^{sph}(x_i^+)$	$f_{exp}^{sph}(x_i^-)r$
$FSVM-CIL_{lin}^{hyp}$	$f_{lin}^{hyp}(x_i^+)$	$f_{lin}^{hyp}(x_i^-)r$
$FSVM-CIL_{exp}^{hyp}$	$f_{exp}^{hyp}(x_i^+)$	$f_{exp}^{hyp}(x_i^-)r$

TABLE II
DETAILS OF THE IMBALANCED DATASETS SELECTED FOR THIS STUDY

Dataset	Pos.	Neg.	Total	Imb. Ratio	Total classes	Positive class
Page-blocks	115	5358	5473	2 : 98	5	5
Abanole	103	4074	4177	2 : 98	29	15
Yeast	51	1433	1484	3 : 97	10	5
miRNA	691	9248	9939	7 : 93	2	miRNA
Satimage	626	5809	6435	10:90	7	4
Ecoli	77	259	336	23:77	8	2
Transfusion	178	570	748	24:76	2	1
Haberman	81	225	306	26:74	2	2
Waveform	1657	3343	5000	33:67	3	0
Pima-Indians	268	500	768	35:65	2	1

these real-world datasets contain some outliers and noisy examples in different amounts. Table II summarizes the details of these datasets in the ascending order of the positive-to-negative dataset ratio. This contains the number of positive-class examples (Pos.), the number of negative-class examples (Neg.), the total number of examples (Total), the positive-to-negative imbalance ratio (Imb. Ratio), total number of classes, and the class selected as the positive class for each dataset. For multiclass datasets, the examples belonging to all the other classes, except the class selected as the positive class, were selected as the negative dataset. These datasets represent a whole variety of domains, complexities, and imbalance ratios.

VII. SUPPORT VECTOR MACHINE/FUZZY SUPPORT VECTOR MACHINE TRAINING AND PERFORMANCE EVALUATION

Before training any classifier, each dataset was scaled into [−1,+1] interval. As the SVM training environment, we selected the widely used libsvm software package [54]. For the FSVM program, we used a variant of the libsvm program found in the libsvm tools (http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/#14), which can be used to assign different weights for different training examples. It has been well-studied that when the training dataset is imbalanced, the commonly used performance measure accuracy, which is the proportion of correctly classified instances, could lead to suboptimal models [5], [18]. Therefore, we used the geometric mean of sensitivity (SE = proportion of the positives correctly recognized) and specificity (SP = proportion of the negatives correctly rec-

ognized), given by $Gm = \sqrt{SE \times SP}$, for the classifier performance evaluation in this study, as commonly used in class imbalanced-learning research [5], [14], [55].

In order to evaluate the performance of all the SVM/FSVM/FSVM-CIL classifiers, which were developed on each dataset, we carried out an extensive five-fold cross-validation method. In this method, a complete dataset was first randomly divided into five partitions. We used stratified random sampling such that each partition contained the same imbalanced ratio. We used four partitions as the training dataset, and the remaining one as the testing dataset. Using the training dataset partition, we develop an SVM/FSVM/FSVM-CIL classifier either by applying a CIL method or by not applying one, in the following way. We considered the radial-basis function (RBF) kernel ($K(x_i, x_j) = e^{-\gamma(\|x_i - x_j\|^2)}$) for SVM training and selected the following ranges for parameters: $\log_2 C = \{1, 2, \ldots, 15\}$, and $\log_2 \gamma = \{-15, -14, \ldots, -1\}$. First, we conducted a coarse grid-parameter-search to find the optimal values for $\log_2 C$ and $\log_2 \gamma$ over the aforementioned ranges of values. We evaluated the performance of a model trained on each parameter pair ($\log_2 C, \log_2 \gamma$) by using five-fold cross-validation results on the training dataset. After finding the optimal values for the parameters, say ($\bar{C}, \bar{\gamma}$), again, a narrow grid-parameter-search was conducted over the ranges $\log_2 C = \{\bar{c} - 0.75, \bar{c} - 0.5, \ldots, \bar{c} + 0.75\}$, and $\log_2 \gamma = \{\bar{\gamma} - 0.75, \bar{\gamma} - 0.5, \ldots, \bar{\gamma} + 0.75\}$. Finally, after finding the optimal values for ($\log_2 C, \log_2 \gamma$), a new SVM model was trained by using the complete training dataset on these parameter values. Then, the performance of that model was tested on the remaining testing partition. This whole training and testing procedure was repeated five times with different training and testing partitions in an outer five-fold cross-validation loop, and finally, the results on the testing partitions were averaged and reported. Hereafter, we refer to this whole cross-validation procedure as the *outer–inner–cv* method.

VIII. EXPERIMENTAL RESULTS

In this section, we present and discuss, in detail, the results obtained by the experiments carried out in this research.

A. SVM and Normal FSVM Results

First, we developed a set of normal SVM classifiers on the original imbalanced datasets and evaluated their performance by using the *outer–inner–cv* method. These results are given in column 3 of Table III. From these results, we can clearly observe that for all the datasets, the resulted SVM classifiers favored the majority negative class over the minority positive class by giving suboptimal classification results (higher SP and lower SE). These results demonstrate the fact that SVMs are sensitive to imbalance in datasets.

Next, we applied normal FSVM method to develop classifiers for each dataset and evaluated their results using the *outer–inner–cv* method. When assigning membership values for the training examples in normal FSVMs, we considered the membership of any training example to be equal to $f(x_i)$ so that any positive- or negative-training example was assigned

TABLE III
CLASSIFICATION RESULTS OBTAINED BY NORMAL SVM AND NORMAL FSVM SETTINGS FOR EACH DATASET

Dataset	Results	Normal SVM (%)	$FSVM_{lin}^{cen}$ (%)	$FSVM_{exp}^{cen}$ (%)	$FSVM_{lin}^{sph}$ (%)	$FSVM_{exp}^{sph}$ (%)	$FSVM_{lin}^{hyp}$ (%)	$FSVM_{exp}^{hyp}$ (%)
Page-blocks	Gm	76.14	78.37	**81.63** (1.0)	77.28	76.16 (0.9)	76.15	77.85 (0.8)
	SE	58.26	61.74	66.96	60.00	58.26	58.26	60.87
	SP	99.52	99.48	99.51	99.53	99.55	99.53	99.57
Abanole	Gm	21.93	0.00	19.58 (0.2)	9.99	19.58 (0.2)	13.78	**21.94** (0.5)
	SE	4.86	0.00	3.86	1.00	3.86	1.90	4.86
	SP	99.02	99.83	99.41	99.78	99.41	99.63	99.09
Yeast	Gm	58.68	56.56	**65.11** (1.0)	60.09	63.43 (1.0)	44.73	60.37 (0.4)
	SE	35.09	35.09	43.09	39.09	41.09	27.27	37.09
	SP	98.12	98.40	98.40	97.84	97.91	97.98	98.26
miRNA	Gm	90.73	90.26	90.66 (1.0)	90.04	90.05 (0.7)	90.17	**90.81** (0.3)
	SE	82.78	81.91	82.63	81.48	81.48	81.77	82.92
	SP	99.45	99.48	99.47	99.54	99.51	99.48	99.44
Satimage	Gm	81.52	81.16	**82.99** (0.4)	82.25	82.05 (0.7)	80.97	82.52 (0.1)
	SE	68.53	67.89	71.08	69.81	69.33	67.57	70.30
	SP	96.97	97.02	96.88	96.90	97.11	97.02	96.86
Ecoli	Gm	84.99	87.31	87.35 (0.4)	84.68	86.96 (0.1)	**87.36**	86.89 (0.7)
	SE	76.67	80.60	80.33	75.50	80.60	81.01	80.45
	SP	94.21	94.60	94.99	94.98	93.83	94.22	93.84
Transfusion	Gm	54.47	55.10	**57.36** (0.4)	55.05	52.39 (1.0)	54.15	55.95 (0.2)
	SE	32.03	32.59	35.38	32.59	29.24	31.48	33.71
	SP	92.63	93.16	92.98	92.98	93.86	93.16	92.81
Haberman	Gm	42.04	49.12	**49.42** (0.8)	45.63	46.45 (0.1)	43.91	48.15 (0.8)
	SE	19.78	26.10	27.21	22.21	23.46	20.96	25.96
	SP	89.33	92.44	89.78	93.78	92.00	92.00	89.33
Waveform	Gm	87.67	87.53	**88.32** (0.8)	87.64	88.13 (0.1)	87.65	88.05 (0.1)
	SE	82.86	81.65	84.07	81.17	83.40	82.38	83.22
	SP	92.76	93.84	92.79	94.62	93.12	93.27	93.15
Pima-Indians	Gm	68.94	69.12	**70.06** (0.5)	69.04	69.75 (0.8)	66.88	69.66 (0.2)
	SE	53.04	54.54	55.65	53.79	54.91	51.54	54.89
	SP	89.60	87.60	82.20	88.60	88.60	86.80	88.40

The optimal value of β under which the highest Gm was obtained, when the exponentially decaying function was considered as $f(x_i)$ in FSVMs, is also given for each dataset next to the corresponding Gm value.

a membership value in [0,1] interval, which was based on its within-class importance. In order to define the function $f(x_i)$, we considered the three methods described in Section V-A1–A3, by using both linear- and exponential-decaying functions. Thus, this way of assigning memberships also resulted in six normal FSVM settings, which are represented as $FSVM_{lin}^{cen}$, $FSVM_{exp}^{cen}$, $FSVM_{lin}^{sph}$, $FSVM_{exp}^{sph}$, $FSVM_{lin}^{hyp}$, and $FSVM_{exp}^{hyp}$. When an exponentially decaying function was used as $f(x_i)$, the optimal value of β was chosen from the range $\beta = \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1\}$, based on the results of *outer–inner–cv* method for each dataset separately. When the linearly decaying function was used, we set $\Delta = 10^{-6}$.

The results obtained through these six FSVM settings for each dataset are given in columns 4–9 of Table III. From these results, we can see that the highest normal FSVM results, which are depicted in bold type, are better than the results given by the normal SVM method for all datasets, due to the better handling of outliers and noise present in the datasets by the FSVM method. For *Page-blocks*, *Yeast*, *Satimage*, *Transfusion, Haberman*, *Waveform,* and *Pima-Indians* datasets, $FSVM_{exp}^{cen}$ method resulted in the highest Gm value. For *Escherichia coli* (*E. coli*) dataset, $FSVM_{lin}^{hyp}$ method yielded the highest results, while for *Abanole* and *miRNA* datasets, $FSVM_{exp}^{hyp}$ yielded the highest results. Therefore, we can observe that the best FSVM setting is dataset-dependent.

More importantly, we can see that some FSVM settings performed worse than the normal SVM algorithm for some datasets. For example, $FSVM_{lin}^{cen}$ method gave lower results for *Abanole, Yeast, miRNA, Satimage,* and *Waveform* datasets than the results given by the normal SVMs. Therefore, we can say that

TABLE IV
CLASSIFICATION RESULTS OBTAINED FOR EACH DATASET BY THE PROPOSED FSVM-CIL SETTINGS COMPARED WITH THE BEST NORMAL FSVM RESULTS

Dataset	Results	Best FSVM (%)	$FSVM-CIL_{lin}^{cen}$ (%)	$FSVM-CIL_{exp}^{cen}$ (%)	$FSVM-CIL_{lin}^{sph}$ (%)	$FSVM-CIL_{exp}^{sph}$ (%)	$FSVM-CIL_{lin}^{hyp}$ (%)	$FSVM-CIL_{exp}^{hyp}$ (%)
Page-blocks	Gm	81.63	94.51	94.14 (0.9)	**95.19**	94.85 (0.7)	93.55	94.64 (0.5)
	SE	66.96	93.91	93.04	95.65	94.78	92.17	94.78
	SP	99.51	95.11	95.26	94.74	94.92	94.94	94.49
Abanole	Gm	21.94	71.90	70.10 (0.4)	68.87	71.13 (0.5)	67.34	**73.98** (0.4)
	SE	4.86	66.95	67.95	55.24	68.95	59.43	82.52
	SP	99.09	77.22	72.31	85.86	73.37	76.32	66.32
Yeast	Gm	65.11	**85.03**	84.93 (0.9)	83.40	83.93 (0.2)	83.86	84.46 (0.3)
	SE	43.09	82.36	82.36	78.36	80.36	78.36	82.36
	SP	98.40	87.79	87.58	88.76	87.65	88.76	86.60
miRNA	Gm	90.81	93.17	93.41 (0.3)	93.30	93.40 (1.0)	**93.43**	93.42 (0.3)
	SE	82.92	89.87	90.45	90.60	90.45	90.74	90.60
	SP	99.44	96.59	96.46	96.08	96.44	96.19	96.33
Satimage	Gm	82.99	90.01	**94.14** (0.9)	88.35	89.27 (0.8)	90.59	90.76 (0.2)
	SE	71.08	87.69	93.04	83.86	85.62	89.14	89.93
	SP	96.88	92.39	95.26	93.08	93.08	92.06	91.60
Ecoli	Gm	87.36	88.47	**90.11** (0.3)	87.45	89.72 (0.6)	89.52	89.72 (0.7)
	SE	81.01	90.92	92.25	88.42	92.25	90.92	92.25
	SP	94.22	86.09	88.02	86.49	87.25	86.09	87.25
Transfusion	Gm	57.36	67.82	66.59 (0.5)	**68.53**	65.89 (0.8)	67.68	68.42 (0.1)
	SE	35.38	69.00	63.98	69.00	60.65	69.63	67.90
	SP	92.98	66.67	69.30	68.07	71.58	65.97	68.95
Haberman	Gm	49.42	61.41	63.96 (1.0)	63.44	64.13 (1.0)	64.85	**65.86** (0.6)
	SE	27.21	49.34	49.49	50.59	49.49	53.16	58.09
	SP	89.78	76.44	82.67	79.55	83.11	79.11	74.68
Waveform	Gm	88.32	88.76	88.78 (0.3)	88.74	88.75 (0.2)	88.90	**89.07** (0.5)
	SE	84.07	88.60	89.86	87.21	89.56	89.56	90.65
	SP	92.79	88.93	87.71	90.31	87.95	88.24	87.53
Pima-Indians	Gm	69.12	71.36	72.63 (0.5)	**73.52**	73.22 (1.0)	71.17	73.13 (0.5)
	SE	54.54	66.83	69.05	70.56	70.54	66.83	70.55
	SP	87.60	76.20	76.40	76.60	76.00	75.80	75.80

The optimal value of β under which the highest Gm was obtained, when the exponentially decaying function was considered, is also given for each dataset next to the corresponding Gm value.

the method of assigning fuzzy-membership values for training examples would play a major role when applying the FSVM method for datasets. In other words, the wrong way of assigning membership values can result in worse-performing models than the normal SVM model. Therefore, in order to find the best FSVM setting, ideally, all the available settings should be explored, which is very computationally demanding, especially with large datasets. However, interestingly, we can observe that for all the datasets considered, the results given by the $FSVM_{exp}^{hyp}$ setting were higher than the normal SVM results. Therefore, we can state that $FSVM_{exp}^{hyp}$ setting with the appropriate selection of β value would be an effective choice for normal FSVMs applied to any dataset.

Nevertheless, we can clearly observe that the results given by all the normal FSVM settings for all the datasets were biased toward the majority of negative class (i.e., higher SP and lower SE). Therefore, these results justify the fact that, like the normal SVMs, normal FSVM algorithm is also sensitive to the class imbalance problem.

B. FSVM-CIL Results

We next focused on the proposed FSVM-CIL method and considered the six FSVM-CIL settings, which are presented in Table I. As mentioned earlier, the value of r was set to the positive-to-negative dataset ratio of the corresponding dataset. When the exponentially decaying function was used as $f(x_i)$, the optimal value of β was chosen from the range $\beta = \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1\}$, based on the results of the *outer–inner–cv* method for each dataset. For linearly decaying function, we set $\Delta = 10^{-6}$. The classification results, which are obtained by the six FSVM-CIL settings for each dataset via the *outer–inner–cv* method, are given in columns 4–9 of Table IV. The best normal FSVM results for each dataset are also given in columns 3 for comparative reasons. From these results, we can see that all FSVM-CIL settings performed well on all datasets, which gives both higher SE and SP and, hence, higher Gm compared with the original classification results, which were obtained by the normal SVM

and normal FSVM methods. Moreover, we can observe that the best classification results, which are depicted in bold type, for different datasets were given by different FSVM-CIL settings: $\text{FSVM} - \text{CIL}_{\text{lin}}^{\text{cen}}$ method gave the highest Gm for *Yeast* dataset, while the $\text{FSVM} - \text{CIL}_{\text{exp}}^{\text{cen}}$ method gave the highest Gm for *Satimage* and *E. coli* datasets; $\text{FSVM} - \text{CIL}_{\text{lin}}^{\text{sph}}$ yielded the highest results for *Pageblocks, Transfusion,* and *Pima-Indians* datasets, and $\text{FSVM} - \text{CIL}_{\text{lin}}^{\text{hyp}}$ gave the highest results for the miRNA dataset; the $\text{FSVM} - \text{CIL}_{\text{exp}}^{\text{hyp}}$ method gave the highest Gm for *Abanole, Haberman*, and *Waveform* datasets. This is due to the characteristic inherited from FSVMs that the optimal way of defining $f(x_i)$ is dependent on the dataset.

C. Existing Class Imbalance Learning Results

In order to compare the effectiveness of the proposed FSVM-CIL method with the existing CIL methods available for SVMs, we applied random undersampling (Under), random oversampling (Over), SMOTE, DEC, and zSVM methods to develop normal SVM classifiers for each imbalanced dataset and evaluated their performance using the *outer—inner–cv* method. The resampling methods were applied until the training partitions were balanced while keeping the testing partitions in their original imbalanced distributions. Since randomness is associated with all of these resampling methods, each method was repeated five times for each cross-validation run in the *outer–inner–cv* method, and the results were averaged. In the SMOTE algorithm, the number of nearest neighbors (k) was set to 5. When applying the DEC method, the value of C^-/C^+ was set to the positive-to-negative class ratio of the dataset by following the findings in [5]. In the zSVM method, the range of z was fixed such that $z \in \{1 + 10^{-10}, 1 + 10^{-9}, \ldots, 1 + 10^{-3}\}$, and the value of z giving the highest classification Gm for the training dataset partition was selected as the optimal value of z to modify the coefficients of the positive-support vectors.

Tables V and VI present the results obtained by applying these existing CIL methods with normal SVMs for each dataset. The results on each dataset are sorted in the descending order of the Gm metric value. From the results presented in Tables V and VI, we can observe that all the existing imbalance-learning methods obtained better classification results than the results given by the normal SVM training for all the imbalanced datasets, as expected. Moreover, for different datasets, the highest Gm was yielded by different existing CIL methods. This evidence supported the fact reported in [18] that the CIL method giving the best results is dataset-dependent.

D. Comparison of Results

Next, we compare the results obtained by the proposed FSVM-CIL settings with the results obtained by the existing CIL methods, which were applied for normal SVMs by using the Gm metric values. Tables VII–X compare these results for each dataset separately. The normal SVM results and the best FSVM results are also presented for comparative reasons. For each dataset, the results are sorted in the descending order of the Gm metric value.

TABLE V
CLASSIFICATION RESULTS OBTAINED BY APPLYING THE EXISTING CIL METHODS WITH NORMAL SVMs FOR *PAGE-BLOCKS, ABANOLE, YEAST. MIRNA, SATIMAGE, E. COLI, TRANSFUSION*, AND *HABERMAN* DATASETS

Dataset	Learning method	SE (%)	SP (%)	Gm (%)
Page-blocks	**DEC**	93.04	94.87	**93.95**
	Over	91.83	94.88	93.34
	Under	93.74	92.52	93.13
	SMOTE	87.30	95.46	91.29
	zSVM	85.22	95.65	90.28
	Normal SVM	58.26	99.52	76.14
Abanole	**Under**	79.12	67.66	**73.17**
	Over	74.11	71.38	72.73
	SMOTE	74.94	70.50	72.69
	DEC	74.67	70.08	72.34
	zSVM	41.71	75.87	56.26
	Normal SVM	4.86	99.02	21.93
Yeast	**Over**	81.09	86.55	**83.77**
	DEC	80.36	87.09	83.66
	Under	80.80	85.91	83.31
	SMOTE	79.56	87.06	83.23
	zSVM	66.91	93.02	78.89
	Normal SVM	35.09	98.12	58.68
miRNA	**zSVM**	89.87	97.10	**93.42**
	Over	89.93	96.53	93.17
	DEC	90.02	96.30	93.11
	SMOTE	89.15	97.04	93.01
	Under	91.03	93.02	92.02
	Normal SVM	82.78	99.45	90.73
Satimage	**DEC**	90.89	90.27	**90.58**
	Under	92.11	86.25	89.13
	Over	88.30	89.21	88.75
	SMOTE	85.49	89.85	87.64
	zSVM	79.07	94.47	86.43
	Normal SVM	68.53	96.97	81.52
Ecoli	**Over**	91.73	86.02	**88.83**
	zSVM	87.08	89.97	88.51
	SMOTE	88.40	87.95	88.17
	Under	91.95	84.41	88.10
	DEC	90.92	84.17	87.48
	Normal SVM	76.67	94.21	84.99
Transfusion	**Under**	72.06	64.18	**68.00**
	Over	67.38	68.35	67.86
	DEC	66.78	68.60	67.68
	SMOTE	64.2	69.86	66.97
	zSVM	55.02	76.14	64.28
	Normal SVM	32.03	92.63	54.47
Haberman	**Over**	53.72	77.42	**64.49**
	zSVM	54.26	72.89	62.89
	SMOTE	48.19	80.62	62.33
	DEC	50.81	76.45	62.32
	Under	51.51	74.76	62.06
	Normal SVM	19.78	89.33	42.04

From results reported in Tables VII–X, we can observe that for nine out of ten datasets that were considered in this paper (all datasets except the *Pima-Indians* dataset), the results given by the best FSVM-CIL setting are better than the results given by all the existing imbalance-learning methods, which were applied for normal SVMs, with respect to the Gm metric. Although, for

TABLE VI
CLASSIFICATION RESULTS OBTAINED BY APPLYING THE EXISTING CIL
METHODS WITH NORMAL SVMs FOR *WAVEFORM* AND *SATIMAGE* DATASETS

Dataset	Learning method	SE (%)	SP (%)	Gm (%)
Waveform	**Over**	90.21	87.41	**88.80**
	zSVM	87.57	90.04	88.78
	DEC	90.41	87.11	88.74
	Under	90.73	86.47	88.58
	SMOTE	89.61	87.19	88.39
	Normal SVM	82.86	92.76	87.67
Satimage	**DEC**	90.89	90.27	**90.58**
	Under	92.11	86.25	89.13
	Over	88.30	89.21	88.75
	SMOTE	85.49	89.85	87.64
	zSVM	79.07	94.47	86.43
	Normal SVM	68.53	96.97	81.52

TABLE VII
COMPARISON OF THE CLASSIFICATION RESULTS OF *PAGE-BLOCKS* DATASET

Dataset	Learning method	Gm (%)
Page-blocks	$FSVM-CIL_{lin}^{sph}$	**95.19**
	$FSVM-CIL_{exp}^{sph}$	94.85
	$FSVM-CIL_{exp}^{hyp}$	94.64
	$FSVM-CIL_{lin}^{cen}$	94.51
	$FSVM-CIL_{exp}^{cen}$	94.14
	DEC	93.95
	$FSVM-CIL_{lin}^{hyp}$	93.55
	Over	93.34
	Under	93.13
	SMOTE	91.29
	zSVM	90.28
	Best FSVM	81.63
	Normal SVM	76.14

The best results are presented in bold type. The results obtained by
the FSVM–CIL $_{exp}^{hyp}$ method are underlined.

TABLE VIII
COMPARISON OF THE CLASSIFICATION RESULTS OF *ABANOLE*, *YEAST*, AND
miRNA DATASETS

Dataset	Learning method	Gm (%)
Abanole	$FSVM-CIL_{exp}^{hyp}$	**73.98**
	Under	73.17
	Over	72.73
	SMOTE	72.69
	DEC	72.34
	$FSVM-CIL_{lin}^{hyp}$	71.90
	$FSVM-CIL_{exp}^{sph}$	71.13
	$FSVM-CIL_{exp}^{cen}$	70.10
	$FSVM-CIL_{lin}^{sph}$	68.87
	$FSVM-CIL_{lin}^{cen}$	67.34
	zSVM	56.26
	Best FSVM	21.94
	Normal SVM	21.93
Yeast	$FSVM-CIL_{lin}^{cen}$	**85.03**
	$FSVM-CIL_{exp}^{cen}$	84.93
	$FSVM-CIL_{exp}^{hyp}$	84.46
	$FSVM-CIL_{exp}^{sph}$	83.93
	$FSVM-CIL_{lin}^{hyp}$	83.86
	Over	83.77
	DEC	83.66
	$FSVM-CIL_{lin}^{sph}$	83.40
	Under	83.31
	SMOTE	83.23
	zSVM	78.89
	Best FSVM	65.11
	Normal SVM	58.68
miRNA	$FSVM-CIL_{lin}^{hyp}$	**93.43**
	$FSVM-CIL_{exp}^{hyp}$	93.42
	zSVM	93.42
	$FSVM-CIL_{exp}^{cen}$	93.41
	$FSVM-CIL_{exp}^{sph}$	93.40
	$FSVM-CIL_{lin}^{sph}$	93.30
	$FSVM-CIL_{lin}^{cen}$	93.17
	Over	93.17
	DEC	93.11
	SMOTE	93.01
	Under	92.02
	Best FSVM	90.81
	Normal SVM	90.73

The best results are presented in bold type. The results obtained by
the FSVM–CIL $_{exp}^{hyp}$ method are underlined.

the *Pima-Indians* dataset, the zSVM method yielded the best results, the results given by FSVM-CIL settings were still competitive with the results given by other imbalance-learning methods.

IX. STATISTICAL COMPARISON OF RESULTS

In this section, we present a statistical test used to validate the significance of the results obtained by the proposed FSVM-CIL methods over the results obtained by the existing CIL methods applied for normal SVMs. Demsar [56] recommends several nonparametric statistical tests that can be used to compare classifiers, which are developed on multiple datasets, when the conditions to apply the parametric tests are not satisfied. Among these tests, we selected the Friedman test [57], [58] to compare the proposed FSVM-CIL method with the existing CIL methods considered. This test first ranks the algorithms considered for each dataset separately, such that the best-performing algorithm would get rank 1, the second best one would get rank 2, etc. In case of ties, it assigns average ranks. Let r_i^j be the rank of the jth algorithm out of k algorithms on the ith dataset out of

TABLE IX
COMPARISON OF THE CLASSIFICATION RESULTS OF *SATIMAGE, E. COLI,* AND *TRANSFUSION* DATASETS

Dataset	Learning method	Gm (%)
Satimage	$FSVM - CIL_{\exp}^{cen}$	**94.14**
	$FSVM - CIL_{\exp}^{hyp}$	90.76
	$FSVM - CIL_{lin}^{hyp}$	90.59
	DEC	90.58
	$FSVM - CIL_{lin}^{cen}$	90.01
	$FSVM - CIL_{\exp}^{sph}$	89.27
	Under	89.13
	Over	88.75
	$FSVM - CIL_{lin}^{sph}$	88.35
	SMOTE	87.64
	zSVM	86.43
	Best FSVM	82.99
	Normal SVM	81.52
Ecoli	$FSVM - CIL_{\exp}^{cen}$	**90.11**
	$FSVM - CIL_{\exp}^{hyp}$	90.72
	$FSVM - CIL_{\exp}^{sph}$	89.72
	$FSVM - CIL_{lin}^{hyp}$	89.52
	Over	88.83
	zSVM	88.51
	$FSVM - CIL_{lin}^{cen}$	88.47
	SMOTE	88.17
	Under	88.10
	DEC	87.48
	$FSVM - CIL_{lin}^{sph}$	87.45
	Best FSVM	87.36
	Normal SVM	84.99
Transfusion	$FSVM - CIL_{lin}^{sph}$	**68.53**
	$FSVM - CIL_{\exp}^{hyp}$	68.42
	Under	68.00
	Over	67.86
	$FSVM - CIL_{lin}^{cen}$	67.82
	$FSVM - CIL_{lin}^{hyp}$	67.78
	DEC	67.68
	SMOTE	66.97
	$FSVM - CIL_{\exp}^{cen}$	66.59
	$FSVM - CIL_{\exp}^{sph}$	65.89
	zSVM	64.28
	Best FSVM	57.36
	Normal SVM	54.47

The best results are presented in bold type. The results obtained by the FSVM–CIL $_{\exp}^{hyp}$ method are underlined.

TABLE X
COMPARISON OF THE CLASSIFICATION RESULTS OF *HABERMAN, WAVEFORM,* AND *PIMA-INDIANS* DATASETS

Dataset	Learning method	Gm (%)
Haberman	$FSVM - CIL_{\exp}^{hyp}$	**65.86**
	$FSVM - CIL_{lin}^{hyp}$	64.85
	Over	64.49
	$FSVM - CIL_{\exp}^{sph}$	64.13
	$FSVM - CIL_{\exp}^{cen}$	63.96
	$FSVM - CIL_{lin}^{sph}$	63.44
	zSVM	62.89
	SMOTE	62.33
	DEC	62.32
	Under	62.06
	$FSVM - CIL_{lin}^{cen}$	61.41
	Best FSVM	49.42
	Normal SVM	42.04
Waveform	$FSVM - CIL_{\exp}^{hyp}$	**89.07**
	$FSVM - CIL_{lin}^{hyp}$	88.90
	Over	88.80
	zSVM	88.78
	$FSVM - CIL_{\exp}^{cen}$	88.78
	$FSVM - CIL_{lin}^{cen}$	88.76
	$FSVM - CIL_{\exp}^{sph}$	88.75
	$FSVM - CIL_{lin}^{sph}$	88.74
	DEC	88.74
	Under	88.58
	SMOTE	88.39
	Best FSVM	88.32
	Normal SVM	87.67
Pima-Indians	zSVM	**74.94**
	$FSVM - CIL_{lin}^{sph}$	73.52
	$FSVM - CIL_{\exp}^{sph}$	73.22
	$FSVM - CIL_{\exp}^{hyp}$	73.13
	SMOTE	73.08
	Over	72.77
	$FSVM - CIL_{\exp}^{cen}$	72.63
	Under	72.51
	DEC	72.44
	$FSVM - CIL_{lin}^{cen}$	71.36
	$FSVM - CIL_{lin}^{hyp}$	71.17
	Best FSVM	69.12
	Normal SVM	68.94

The best results are presented in bold type. The results obtained by the FSVM–CIL $_{\exp}^{hyp}$ method are underlined.

TABLE XI
RANKS OF DIFFERENT CIL METHODS FOR EACH DATASET

Dataset	Best FSVM-CIL	Over	Under	SMOTE	DEC	zSVM
Pageblocks	1	3	4	5	2	6
Abanole	1	3	2	4	5	6
Yeast	1	2	4	5	3	6
miRNA	1	3	6	5	4	2
Satimage	1	4	3	5	2	6
Ecoli	1	2	5	4	6	3
Transfusion	1	3	2	5	4	6
Haberman	1	2	6	4	5	3
Waveform	1	2	5	6	4	3
Pima-Indians	2	4	5	3	6	1
Avg. Rank	**1.1**	**2.8**	**4.2**	**4.6**	**4.1**	**4.2**

TABLE XII
DIFFERENCES BETWEEN THE AVERAGE RANKS OF THE EXISTING
IMBALANCE-LEARNING METHODS AND THE AVERAGE RANK OF THE BEST
FSVM-CIL METHOD

Method	Average Rank	Difference
Best FSVM-CIL	1.1	N/A
Over	2.8	1.7
DEC	4.1	3.0
Under	4.2	3.1
zSVM	4.2	3.1
SMOTE	4.6	3.5

N datasets. The Friedman test compares the average ranks of algorithms $R_j = \frac{1}{N}\sum_i r_i^j$. Under the null-hypothesis (i.e., all the algorithms are equivalent so that any differences among their average ranks R_j are merely random), the Friedman statistic

$$\chi_F^2 = \frac{12N}{k(k+1)}\left[\sum_{j=1}^k R_j^2 - \frac{k(k+1)^2}{4}\right] \qquad (24)$$

is distributed according to the χ_F^2 distribution with $k-1$ degrees of freedom when N and k are large enough. However, Iman and Devenport [59] showed that the Friedman's χ_F^2 statistic is undesirably conservative and derived a better statistic

$$F_F = \frac{(N-1)\chi_F^2}{N(k-1)-\chi_F^2} \qquad (25)$$

which is distributed according to the F-distribution with $k-1$ and $(k-1)(N-1)$ degrees of freedom.

In order to compare the results obtained by the best FSVM-CIL method with the results of the existing CIL methods considered (Over, Under, SMOTE, DEC, and zSVM), we first determined the ranks of each algorithm for each dataset separately, which are given in Table XI.

Then, we calculated the following statistics when $N = 10$, and $k = 6$:

$$\chi_F^2 = \frac{12 \times 10}{6 \times 7}$$
$$\times \left[(1.1^2 + 2.8^2 + 4.2^2 + 4.6^2 + 4.1^2 + 4.2^2) - \frac{6 \times 7^2}{4}\right]$$
$$= 25.14$$
$$F_F = \frac{9 \times 25.14}{10 \times 5 - 25.14} = 9.10 \sim F(5, 45).$$

The critical value of $F(5, 45)$ at $\alpha = 0.05$ is 2.42. Since $F_F > 2.42$, we can reject the null-hypothesis, i.e., we can say that the compared algorithms are not equivalent at $\alpha = 0.05$.

Since, we have rejected the null-hypothesis, we can now use the Benferroni–Dunn test [60] to compare the best FSVM-CIL method to the other imbalance-learning methods considered. Under this test, it is considered that the performance of the new algorithm (i.e., the control algorithm) is significantly different from another algorithm if the corresponding average ranks differ by at least the critical difference (CD), which is defined as follows:

$$CD = q_\alpha \sqrt{\frac{k(k+1)}{6N}}. \qquad (26)$$

The precalculated q_α values are given in [56].

For our results, CD $= 2.576\sqrt{6 \times 7/6 \times 10} = 2.15$ at $\alpha = 0.05$. Table XII gives the differences between the average ranks of the existing CIL methods and the average rank of the best FSVM-CIL method.

We can state that the best FSVM-CIL method is significantly different from (i.e., better than) DEC, Under, zSVM, and SMOTE methods at $\alpha = 0.05$, since the differences in average ranks are greater than the CD $= 2.15$. However, based on the difference in average ranks, we cannot say that the best FSVM-CIL method is significantly different from the Over method. Nevertheless, for all the datasets, the best FSVM-CIL method performed better than the Over method (see the actual results in Tables VII–X and the ranks in Table XI). We believe that this statistical test is less powerful in finding the difference between best FSVM-CIL method and the Over method due to the "granularity" property of the nonparametric tests, i.e., since the Over method performed better than the other existing CIL methods (DEC, Under, zSVM, and SMOTE) for most of the datasets, and therefore, it has obtained the best average rank among these existing CIL methods, we cannot find a significant difference between the average ranks of the best FSVM-CIL method and the Over method, despite how much better the best FSVM-CIL method performed over the Over method. However, by considering all these results, we can state that the best FSVM-CIL method is significantly better than the existing imbalance-learning methods, which are considered in this paper. The reason of FSVM-CIL method to perform better than the existing CIL methods, which were applied for normal SVMs, would be the additional capability of the FSVM-CIL method to handle the problem of outliers and noise, to which the existing imbalance-learning methods available for normal SVMs are still sensitive.

Except for the small computational time required to calculate the membership values for the training examples, the proposed FSVM-CIL method does not increase the computational requirements of the underlying SVM training algorithm at all. However, as we observed, the best FSVM-CIL setting giving the highest results was dataset-dependent. Therefore, to explore all the FSVM-CIL settings in order to find the optimal one would be computationally very demanding. However, we can further

observe, from the results obtained, that for the nine datasets, for which the FSVM-CIL method gave the best results, the $FSVM - CIL_{exp}^{hyp}$ setting gave higher results than the results given by the existing CIL methods. The results given by the $FSVM - CIL_{exp}^{hyp}$ method are underlined in Tables VII–X. Based on these results, we can recommend that the $FSVM - CIL_{exp}^{hyp}$ setting with the appropriate selection of β value would be a clever choice when applying FSVM-CIL method for any imbalanced dataset.

X. CONCLUSION

In this paper, we proposed an improved FSVM method, which is called FSVM-CIL, for learning from imbalanced datasets in the presence of outliers and noise. In this method, we assign fuzzy-membership values for training examples in order to handle both the problems of class imbalance and outliers/noise. We considered three ways to assign fuzzy-membership values and six FSVM-CIL settings. We used ten real-world imbalanced datasets to validate the proposed method. We compared the effectiveness of the FSVM-CIL method with five other popular existing imbalance-learning methods that are available for normal SVM training. We also applied a nonparametric statistical test to validate the significance of the performance of the proposed method over the existing CIL methods considered. From the overall results obtained, we can conclude that the proposed FSVM-CIL method could result in significantly better classification results than the existing imbalance-learning methods applied for normal SVMs, especially in the presence of outliers/noise in the datasets. Finally, as a rule of thumb, we can recommend that among the six FSVM-CIL methods considered, the $FSVM - CIL_{exp}^{hyp}$ setting with the appropriate selection of β value would be an effective choice for learning from any imbalanced dataset. As future work, it would be interesting to investigate the effectiveness of using the FSVM method with other existing imbalance-learning methods, such as resampling techniques, for imbalanced dataset learning.

REFERENCES

[1] V. Vapnik, *The Nature of Statistical Learning Theory*. Berlin, Germany: Springer-Verlag, 1995.

[2] C. Cortes and V. Vapnik, "Support vector networks," *Mach. Learning*, vol. 20, pp. 273–297, 1995.

[3] J. Showe-Taylor and N. Christianini, *Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge, U.K.: Cambridge Univ. Press, 2000.

[4] K. Veropoulos, C. Campbell, and N. Cristianini, "Controlling the sensitivity of support vector machines," in *Proc. Int. Joint Conf. Artif. Intell.*, Stockholm, Sweden, 1999, pp. 55–60.

[5] R. Akbani, S. Kwek, and N. Japkowicz, "Applying support vector machines to imbalanced datasets," in *Proc. 15th Eur. Conf. Mach. Learning*, Pisa, Italy, 2004, pp. 39–50.

[6] G. Wu and E. Chang, "Class-boundary alignment for imbalanced data set learning," presented at the Int. Conf. Data Mining, Workshop Learning Imbalanced Data Sets II, Washington, DC, 2003.

[7] G. Wu and E. Chang, "KBA: Kernel boundary alignment considering imbalanced data distribution," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 6, pp. 786–795, Jun. 2005.

[8] B. Raskutti and A. Kowalczyk, "Extreme re-balancing for SVMs: A case study," *ACM SIGKDD Explor. Newslett.*, vol. 6, no. 1, pp. 60–69, 2004.

[9] T. Imam, K. Ting, and J. Kamruzzaman, "z-SVM: An SVM for improved classification of imbalanced data," in *Proc. 19th Aust. Joint Conf. AI*, Hobart, Australia, 2006, pp. 264–273.

[10] S. Zou, Y. Huang, Y. Wang, J. Wang, and C. Zhou, "SVM learning from imbalanced data by GA sampling for protein domain prediction," in *Proc. 9th Int. Conf. Young Comput. Sci.*, Hunan, China, 2008, pp. 982–987.

[11] Z. Lin, Z. Hao, X. Yang, and X. Lium, "Several SVM ensemble methods integrated with under-sampling for imbalanced data learning," in *Advanced Data Mining and Applications*. Berlin, Germany: Springer-Verlag, 2009, pp. 536–544.

[12] P. Kang and S. Cho, "EUS SVMs: Ensemble of under sampled SVMs for data imbalance problems," in *Proc. 13th Int. Conf. Neural Inf. Process.*, Hong Kong, 2006, pp. 837–846.

[13] Y. Liu, A. An, and X. Huang, "Boosting prediction accuracy on imbalanced data sets with SVM ensembles," in *Proc. 10th Pac.-Asia Conf. Adv. Knowl. Discov. Data Mining*, Singapore, 2006, pp. 107–118.

[14] H. Haibo and E. Garcia, "Learning from imbalanced data," *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 9, pp. 1263–1284, Sep. 2009.

[15] B. Boser, I. Guyon, and V. Vapnik, "A training algorithm for optimal margin classifiers," in *Proc. 5th Annu. ACM Workshop Comput. Learning Theory*, Pittsburgh, PA, 1992, pp. 144–152.

[16] X. Zhang, "Using class centres vectors to build support vector machines," in *Proc. IEEE Signal Process. Soc. Workshop*, Madison, WI, 1999, pp. 3–11.

[17] C.-F. Lin and S.-D. Wang, "Fuzzy support vector machines," *IEEE Trans. Neural Netw.*, vol. 13, no. 2, pp. 464–471, Mar. 2002.

[18] G. Weiss, "Mining with rarity: A unifying framework," *SIGKDD Explor. Newslett.*, vol. 6, no. 1, pp. 7–19, 2004.

[19] N. Chawla, N. Japkowicz, and A. Kolcz, "Editorial: Special issue on learning from imbalanced data sets," *SIGKDD Explor. Newslett.*, vol. 6, no. 1, pp. 1–6, 2004.

[20] N. Chawla, K. Bowyer, and P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, 2002.

[21] J. Zhang and I. Mani, "KNN approach to unbalanced data distributions: A case study involving information extraction," in *Proc. Int. Conf. Mach. Learning, Workshop: Learning Imbalanced Data Sets*, Washington, DC, 2003, pp. 42–48.

[22] T. Jo and N. Japkowicz, "Class imbalances versus small disjuncts," *ACM SIGKDD Explor. Newslett.*, vol. 6, no. 1, pp. 40–49, 2004.

[23] X.Y. Liu, J. Wu, and Z. H. Zhou, "Exploratory under sampling for class imbalance learning," in *Proc. 6th IEEE Int. Conf. Data Mining*, Hong Kong, 2006, pp. 965–969.

[24] N. Chawla, A. Lazarevic, L. Hall, and K. Bowyer, "SMOTEBoost: Improving prediction of the minority class in boosting," in *Proc. 7th Eur. Conf. Principles Pract. Knowl. Discov. Databases*, Croatia, 2003, pp. 107–119.

[25] H. Guo and H. Viktor, "Learning from imbalanced data sets with boosting and data generation: The DataBoost IM approach," *ACM SIGKDD Explor. Newslett.*, vol. 6, no. 1, pp. 30–39, 2004.

[26] Z.-H. Zhou and X.-Y. Liu, "Training cost-sensitive neural networks with methods addressing the class imbalance problem," *IEEE Trans. Knowl. Data Eng.*, vol. 18, no. 1, pp. 63–77, Jan. 2006.

[27] D. Cieslak and N. Chawla, "Learning decision trees for unbalanced data," in *Machine Learning and Knowledge Discovery in Databases*. Berlin, Germany: Springer-Verlag, 2008, pp. 241–256.

[28] A. Fernandez, M. Jesus, and F. Herrera, "Hierarchical fuzzy rule based classification systems with genetic rule selection for imbalanced datasets," *Int. J. Approx. Reason.*, vol. 50, no. 3, pp. 561–577, 2009.

[29] A. Fernández, M. Jesus, and F. Herrera, "Improving the performance of fuzzy rule based classification systems for highly imbalanced datasets using an evolutionary adaptive inference system," in *Bio-Inspired Systems: Computational and Ambient Intelligence*. Berlin, Germany: Springer-Verlag, 2009, pp. 294–301.

[30] Y. Wang, S. Wang, and K. Lai, "A new fuzzy support vector machine to evaluate credit risk," *IEEE Trans. Fuzzy Syst.*, vol. 13, no. 6, pp. 820–831, Dec. 2005.

[31] Y.-Y. Hao, Z.-X. Chi, and D.-Q. Yan, "Fuzzy support vector machine based on vague sets for credit assessment," in *Proc. 4th Int. Conf. Fuzzy Syst. Knowl. Discov.*, Changsha, China, 2007, vol. 1, pp. 603–607.

[32] E. Spyrou, G. Stamou, Y. Avrithis, and S. Kollias, "Fuzzy support vector machines for image classification fusing mpeg-7 visual descriptors," in *Proc. 2nd Eur. Workshop Integr. Knowl., Semantics Dig. Media Technol.*, London, U.K., 2005, pp. 23–30.

[33] L. Chen *et al.*, "Fuzzy support vector machines for emg pattern recognition and mycroelectrical prosthesis control," in *Proc. 4th Int. Symp. Neural Netw.*, Nanjing, China, 2007, pp. 1291–1298.

[34] Z. Xie, Q. Hu, and D. Yu, "Fuzzy output support vector machine for classification," in *Proc. Int. Conf. Adv. Natural Comput.*, Changsha, China, 2005, pp. 1190–1197.

[35] T. Inoue and S. Abe, "Fuzzy support vector machine for pattern classification," in *Proc. Int. Conf. Neural Netw.*, Washington, DC, 2001, pp. 1449–1457.

[36] J. Mill and A. Inoue, "An application of fuzzy support vectors," in *Proc. 22nd Int. Conf. N. Amer. Fuzzy Inf. Process. Soc.*, Chicago, IL, 2003, pp. 302–306.

[37] J.-H. Chiang and P.-Y. Hao, "Support vector learning mechanism for fuzzy rule-based modeling: A new approach," *IEEE Trans. Fuzzy Syst.*, vol. 12, no. 1, pp. 1–12, Feb. 2004.

[38] Y. Chen and J. Wang, "Support vector learning for fuzzy rule-based classification systems," *IEEE Trans. Fuzzy Syst.*, vol. 11, no. 6, pp. 716–728, Dec. 2003.

[39] A. Chaves, M. Vellasco, and R. Tanscheit, "Fuzzy rule extraction from support vector machines," presented at the 5th Int. Conf. Hybrid Intell. Syst., Rio de Janeiro, Brazil, 2005.

[40] J. Castro, L. Flores-Hidalgo, C. Mantas, and J. Puche, "Extraction of fuzzy rules from support vector machines," *Fuzzy Sets Syst.*, vol. 158, pp. 2057–2077, 2007.

[41] T. Ross, *Fuzzy Logic With Engineering Applications*. New York: Wiley Blackwell, 2004.

[42] J.-S. R. Jang, "ANFIS: Adaptive-network-based fuzzy inference systems," *IEEE Trans. Syst., Man Cybern.*, vol. 23, no. 3, p. 665—685, May/Jun. 1993.

[43] J. J. Buckley and Y. Hayashi, "Neural networks for fuzzy systems," *Fuzzy Sets Syst.*, vol. 71, pp. 265–276, 1995.

[44] T. Lamata and S. Moral, "Classification of fuzzy measures," *Fuzzy Sets Syst.*, vol. 33, no. 2, pp. 243–253, 1989.

[45] S. Ling, F. Leung, and P. Tam, "Daily load forecasting with a fuzzy-input-neural network in an intelligent home," in *Proc. 10th IEEE Int. Conf. Fuzzy Syst.*, Melbourne, Australia, 2001, pp. 449–452.

[46] C.-F. Lin and S.-D. Wang, "Training algorithms for fuzzy support vector machines with noisy data," in *Proc. 13th IEEE Workshop Neural Netw. Signal Process.*, France, 2003, pp. 517–526.

[47] S. Xiong, H. Liu, and X. Niu, "Fuzzy support vector machines based on λ-cut," in *Proc. Int. Conf. Nat. Comput.*, Changsha, China, 2005, pp. 292–600.

[48] H. Tang and L.-S. Qu, "Fuzzy support vector machines with a new fuzzy membership function for pattern classification," in *Proc. 7th Int. Conf. Mach. Learning Cybern.*, Kunming, China, 2008, pp. 768–773.

[49] H.-B. Liu, S.-W. Xiong, and X.-X. Niu, "Fuzzy support vector machines based on spherical regions," in *Proc. 3rd Int. Symp. Neural Netw.*, Chengdu, China, 2006, pp. 949–954.

[50] H. Xia and B. Hu, "Feature selection using fuzzy support vector machines," *Fuzzy Optim. Decis. Making*, vol. 5, no. 2, pp. 187–192, 2006.

[51] A. Asuncion D. Newman. (2007). *UCI Repository of Machine Learning Database* (School of Information and Computer Science), Irvine, CA: Univ. of California [Online]. Available: http://www.ics.uci.edu/~mlearn/MLRepository.html

[52] R. Batuwita and V. Palade, "microPred: Effective classification of pre-miRNAs for human miRNA gene prediction," *Bioinformatics*, vol. 25, pp. 989–995, 2009.

[53] R. Batuwita and V. Palade, "An improved non-comparative classification method for human miRNA gene prediction," in *Proc. 8th IEEE Int. Conf. Bioinf. Bioeng.*, Athens, Greece, 2008, pp. 1–6.

[54] C.-C. Chang and C.-J. Lin. (2001). "LIBSVM: A library for support vector machines," [Online]. Available: http://www.csie.ntu.edu.tw/~cjlin/libsvm

[55] M. Kubat and S. Matwin, "Addressing the curse of imbalanced training sets: One-sided selection," in *Proc. 14th Int. Conf. Mach. Learning*, Nashville, TN, 1997, pp. 179–186.

[56] J. Demsar, "Statistical comparisons of classifiers over multiple data sets," *J. Mach. Learning Res.*, vol. 7, p. 1, 2006.

[57] M. Friedman, "The use of ranks to avoid the assumption of normality implicit in the analysis of variance," *J. Amer. Statist. Assoc.*, vol. 32, pp. 675–701, 1937.

[58] M. Friedman, "A comparison of alternative tests of significance for the problem of m rankings," *Ann. Math. Statist.*, vol. 11, pp. 86–92, 1940.

[59] R. Iman and J. Devenport, "Approximations of the critical region of the Friedman statistics," *Commun. Statist.*, vol. 9, pp. 571–595, 1980.

[60] O. Dunn, "Multiple comparison among means," *J. Amer. Statist. Assoc.*, vol. 56, pp. 52–64, 1961.

Author's photographs and biographies not available at the time of publication.