Pak.j.stat.oper.res. Vol.17 No.2 2021 pp 483-496

DOI: http://dx.doi.org/10.18187/pjsor.v17i2.3640

Pakistan Journal of Statistics and Operation Research

A Novel Weighted Ensemble Method to Overcome the Impact of Under-fitting and Over-fitting on the Classification Accuracy of the Imbalanced Data Sets

Ghulam Fatima[1], Sana Saeed[2]*

*Corresponding author

1. College of Statistical and Actuarial Sciences, University of the Punjab, Pakistan, gulamfatima123@yahoo.com
2. College of Statistical and Actuarial Sciences, University of the Punjab, Pakistan, sana.stat@pu.edu.pk

Abstract

In the data mining communal, imbalanced class dispersal data sets have established mounting consideration. The evolving field of data mining and information discovery seeks to establish precise and effective computational tools for the investigation of such data sets to excerpt innovative facts from statistics. Sampling methods re-balance the imbalanced data sets consequently improve the enactment of classifiers. For the classification of the imbalanced data sets, over-fitting and under-fitting are the two striking problems. In this study, a novel weighted ensemble method is anticipated to diminish the influence of over-fitting and under-fitting while classifying these kinds of data sets. Forty imbalanced data sets with varying imbalance ratios are engaged to conduct a comparative study. The enactment of the projected method is compared with four customary classifiers including decision tree(DT), k-nearest neighbor (KNN), support vector machines (SVM), and neural network (NN). This evaluation is completed with two over-sampling procedures, an adaptive synthetic sampling approach (ADASYN) and a synthetic minority over-sampling (SMOTE) technique. The projected scheme remained efficacious in diminishing the impact of over-fitting and under-fitting on the classification of these data sets.

Key Words: Imbalanced Data Sets; Under-Fitting; Over-Fitting; Over-Sampling Techniques; Ensemble Method; Weighted Method

Mathematical Subject Classification: 62H30.

1. Background

In the data mining community, imbalanced class distribution data sets have received mounting consideration (He and Garcia, 2009). These data sets have established a vigorous share in every field of exploration (Lewis and Catlett, 1994;Chawla et al., 2002; Yang et al., 2009; Tavallaee et al., 2010). An imbalanced data set has unequal majority and minority class examples (Chawla et al., 2004). While learning from these data sets, two significant issues over-fitting and under-fitting have to be faced by investigators also these two are the foremost reasons for the poor enactment of the machine learning (ML) algorithms. Overfitting states a modeling error that occurs when a function too narrowly agrees to a data set and under-fitting refers to a description that cannot simulate a training data set or generalize it to a new data set (Ying, 2019). Classifier learning with such imbalanced data sets is a thought-provoking task (Leevy et al., 2018). Classifiers show meager enactment as they could not be trained for minority classes (Mathew et al., 2017; Zhang and Chen, 2019; Paing et al., 2018). Numerous sampling procedures have been anticipated which pre-processed the data and balanced its class distribution, henceforth improved classification results are obtained (Pattanayak and Rout, 2018; Kong et al., 2020). Over-sampling methods are desired over under-sampling methods as they stunned the restraint of data loss and deliver better-quality classification performance (Barandela et al., 2003; Kaur and Gosain, 2018). SMOTE and ADASYN are two noteworthy over-sampling methods. SMOTE (Chawla et al., 2002) balances data distribution by synthetically spawning conceivable minority class instances. Whereas, its modification MSMOTE (Modified SMOTE) method which takes into account the latent noise of data and minority data distribution and works on eliminating the noisy data was proposed by Hu et al. (2009). ADASYN generates more synthetic examples for difficult-to-learn minority class examples. Consequently, provides a more presentable form of data (He and Garcia, 2009). ADASYN sufficiently reduces the class imbalance bias and concentrates more on difficult to learn minority class examples by adaptively transferring the classification decision boundary to them. A combination of under-sampling and over-sampling techniques was also suggested by researchers which can better handle the imbalanced nature of data (Barandela et al., 2003). In addition to sampling methods, ensemble methods have gained much popularity to knob imbalanced data sets (Schclar et al., 2009; Zhou, 2012; Hsu, 2017). An ensemble method combines numerous classifiers to achieve robust classification and generalized results. Several investigations have shown that balanced data sets enhance the performance of base classifiers contrasted to imbalanced data sets (Laurikkala, 2001; Estabrooks et al., 2004). Sampling is one of the broadly implemented techniques to manipulate imbalanced data sets. Kubat et al. (1998) introduced a one-sided selection process that scraps borderline or noisy majority class examples. Another study suggested the Edited Nearest Neighbor Rule (ENN), this method erases the examples whose class label was different from the class of at least two of its three Nearest Neighbors (NN) (Li and Zhang, 2011). The objective of this procedure was to diminish the majority class examples nearby the borderline of distinct classes corresponding to the notion of NN.

Many studies have recognized that it is very essential to utilize sampling approaches to improve classification results but it is not clear that which sampling method performs best, what sampling rate should be applied, and proper choice may be domain-specific. Researches have shown that many classifiers such as DT, SVM, KNN and NN etc. suffer from classification issues and they either under-fit or over-fit imbalanced data sets. A research work stated that existing imbalanced learning methods which employ normal SVMs substantially ignore the vital facts of majority class and thus generate over-fitted results (Zhang and Wang, 2013). Few authors explained that SVM shows biasness while learning from imbalanced data sets, as it pushes the boundary away from the positive/minority class when in fact the learned boundary is too close to the minority class.

KNN classifiers are also termed as lazy learners (Akbani et al., 2004; Dasarathy and Sheela, 1979). KNN seeks to find K training points that are very close to the testing point's attributes in order to evaluate its class value. A wrong value for K can lead to problems of over-fitting or over-generalization (Tan et al., 2016). NN consist of neurons arranged in a layer. Each unit takes input and after applying a linear function, its output is transferred to the next unit. But neural network does not converge in classification problems of imbalanced data sets (Panchal et al., 2011; Piotrowski and Napiorkowski, 2013). This failure may occur due to the few hidden neurons (Zhang and Wang, 2013). To overcome the issues related to individual classifier learning, ensemble of multiple classifiers is generated (Hsu, 2017). The ensemble methods are considered very effective to overcome the individual classifier's performance limitations (Lemaître et al., 2017). Ensembles are generated to enhance the robustness of a classifier by training several classifiers and accumulating their results to provide a single class label (Zhou, 2012; Polikar, 2006; Rokach, 2010). Galar derived the taxonomy of ensemble-based methods and conducted an empirical comparison of the methods in a family to observe their performance (Galar et al., 2013). A large number of ensemble methods have been presented in last few years such as RUSBoost (Seiffert et al., 2009), BalanceCadcade (Liu et al., 2008), EasyEnsemble method (Liu and Zhou, 2013) are to name a few.

By reviewing all the accessible material, it comes to our knowledge that the existing material lacks those ensemble methods which can efficiently handle fitting and generalization problems simultaneously. To counter these issues related to the imbalanced data sets, the following are the obectives of this research work:

1. A novel weighted ensemble method will be anticipated.

2. The novel method will aim to balance both fitting and generalization by weighting the evaluation measures generated by individual classifiers and thus improving the accuracy of classification.

3. The focus will be to discover the configuration of a combination of models to regulate whether a group of classifiers can overcome the over-fitting and under-fitting issues.

The rest of the paper is organized as follows: material and methods related to this research work is discussed in Section 2, the proposed weighted ensemble method is presented in Section 3, the results of a simulation study is discussed in Section 4. Thereafter, results and conclusions are summarized in Section 5. References are provided at the end of this paper.

2. Material and Methods

Four well-established classifiers are involved in this analysis. A brief introduction of these classifiers is provided in the following subsections.

1. Decision Tree
2. K Nearest Neighbor
3. Support Vector Machine
4. Neural network

2.1. Decision Tree

DT learning is the most widely used predictive modeling technique for classification (Anyanwu and Shiva, 2009). DTs partition the input space into the cells, where each cell belongs to a class. Partitioning is seen as a series of measures. Every interior node in the DT corresponds to one test value of some input variable. Some input variables and node divisions are labeled with the potential test results. The leaf nodes represent the cells and define the class to be returned if the leaf node is reached. The classification of a particular instance is then carried out by beginning at the root node and, based on the results of the test, following the necessary branches until the leaf node is reached (Freund and Mason, 1999). The complete steps taken for the implementation of a DT is shown in a pseudo code 1.

Algorithm 1 Pseudo Code of DT (Saxena, 2017)

```
Begin
    1. Tree learning
    2. Develop the root node by placing the best attributes of data at the root of the tree
    3. Prepare subsets of training data such as
    4. Each subset has data with the same value of the attribute
       repeat
    5. find leaf nodes for all branches of the tree
End
```

2.2. K Nearest Neighbour

KNN is a trendy and simple supervised machine learning (ML) algorithm. The notion for this algorithm is k-nearest training examples in sample space and output depends upon classification. KNN algorithm intends to classify a new example centered on its characteristics and labeled training examples. KNN is a memory-based algorithm and does not require a model to be fit. Given an example x to be classified, k nearest points (based on Euclidean distance) of x are noticed, and permitting to the majority of its neighbors found, x is classified to its relevant class. Any ties in voting are broken at random. KNN is known as a lazy learner as it slows down the procedure of modeling until it is needed to classify an example. It does not take into account any assumption of data distribution, it is working on. Thus, it is also mentioned as a non-parametric procedure. The steps adopted for KNN are shown in pseudo code 2

Algorithm 2 Pseudo Code of KNN (Tay et al., 2014)

```
Begin
  1. Classify $(X, Y, x)$ // X: training data, Y: class labels of X, x: unknown sample
  2. for $i = 1$ to $m$ do
       Compute distance $d(X_i, x)$
     end for
  3. Compute set $I$ containing indices for the $k$ smallest distances $d(X_i, x)$.
     return
  4. Majority label for $Y_i$ where $i \epsilon I$
End
```

2.3. Support Vector Machine

SVM is a supervised learning algorithm that linearly classifies two-class imbalanced data problems. For multi-class imbalanced data problems, this data is first converted into two classes and then categorized by SVM. This algorithm builds a hyperplane between two sets of data. Since the data adjoining on the borderline is tricky to discover so it is essential to pick the precise hyperplane for increased accuracy. The best hyperplane is the one that has an extreme distance to the nearest data point (Bennett and Blue, 1998). Given training data (x_i, yi) for *i = 1 ...N*, with $xi \in R^d$ and $yi \in -1, 1$, learn a classifier $f(x)$ such that

$$\begin{aligned} f(xi)\{ &\geq 0 \quad yi = +1 \\ \{ &< 0 \quad yi = -1 \end{aligned} \tag{1}$$

i.e. $y_i f(x_i) > 0$ for a correct classification. The easiest model of SVM begins with a maximal margin classifier. Its running is restricted to linearly distinct instances in feature space and it presumes that there is no training error. The pseudo code of SVM is provided in 3.

Algorithm 3 Training an SVM (Pedersen and Schoeberl, 2006)

```
Begin
  1. Training an SVM
  2. Require: X and y loaded with training labeled data, $- \Leftarrow 0$ or $- \Leftarrow$ partially trained SVM
  3. Assign the values C
     repeat
  4. for all $x_i, y_i, x_j, y_j$ do
  5. Optimize $-i$ and $-j$ end for
  6. Until no changes in $-$ or other resource constraint criteria met
     Ensure Retain only the support vectors $(-_i > 0)$
End
```

2.4. Neural Network

NN is a computational model intended by biological neurons. It is expounded as a directed graph where neurons indicate nodes and their connections indicate edges. Weight on each edge reveals the simulating nature and power of collaboration among neurons. Perceptron is one of the easiest neural networks that involve only one neuron with multiple binary inputs and outputs. The input of neurons is directed across weighted edges and the accumulation of all the weighted inputs is treated as net input of a neuron. The net input is matched with a threshold if it is greater than

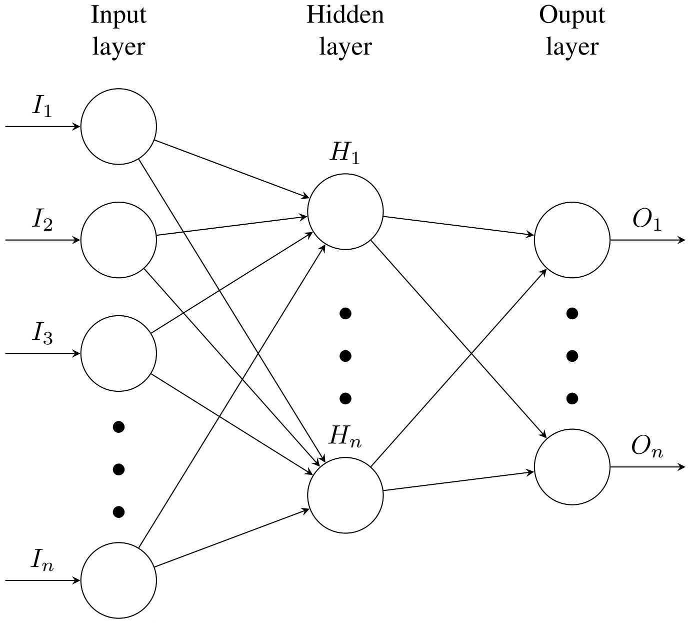

Figure 1: Flow of Neural Network

the threshold it will generate “1” as an output or else “0”. In general, there are three layers in a neuron; input layer, hidden layer, and output layer. The input layer merely accelerates the recorded values to the following layer. There may be numerous hidden layers in a single neural network. In the output layer, a specific class is represented by each node. This whole course of layers results in the assignment of values to each output node and it classifies the record to the class node with the highest value (Hansen and Salamon, 1990; Abdi), 1994). The activation function used for NN is sigmoid (see Eq 2).

$$y = \frac{1}{1+e^{-x}} \tag{2}$$

The complete working of NN is shown in the flow chart in Figure 1.

2.5. Synthetic Minority Over Sampling Technique

SMOTE is an over-sampling technique for the imbalanced data sets proposed by Chawla et al. (2002). In contrast to the other techniques, SMOTE uses an over-sampling of the minority class by creating synthetic samples. Subject to the amount of over-sampling requirement, neighbors from the k nearest points are carefully chosen. For example, if the enormity of the required over-sampling is 200 percent then only two are chosen from the five nearest neighbors and generate one sample in the direction of each. The steps which are taken to produce a synthetic sample by using SMOTE are provided in pseudo code 4.

Algorithm 4 Procedure of SMOTE

```
Begin
    1. Compute the difference between the NN and the feature vector (sample).
    2. Multiply the difference by a random number generated between 0 and 1.
    3. Add it to the concerned feature vector. This will invent the choice of a random point near the line segment
       between the two features.
end
```

This technique successfully broader the decision area of the minority class. The implementation of this algorithm requires five nearest neighbors (Han et al., 2005).

2.6. Adaptive Synthetic Sampling Approach

ADASYN produces the samples of the minority class centered on their density distributions. It determines the nearest K-neighbors of each minority instance, then gets the minority-majority class ratio to generate new samples. By replicating this process, it adaptively shifts the decision boundary to concentrate on samples that are challenging to learn (He et al., 2008; He and Garcia, 2009). The complete steps of ADASYN is shown in pseudo code 5.

Algorithm 5 Pseudo Code of ADASYN (He et al., 2008)

```
Begin
    1. Calculate the ratio of minority to majority examples by using
       $d = m_s/m_l$, where $m_s$ and $m_l$ are the number of minority and majority class examples
    2. Calculate the total number of synthetic minority data to generate
       $G = (m_l - m_s)\beta$, where $G$ is the total number of minority data to generate. $\beta$ is the ratio of minority
    3. Find the k-Nearest Neighbours of each minority example and calculate the r- value. After this step, each minority
       example should be associated with a different neighborhood.
    4. Calculate the number of synthetic samples to generate per neighbor.
    5. Calculate $G_i$ for each neighborhood.
end
```

3. Proposed Weighted Ensemble Method

The proposed method, the weighted ensemble method (WEM)is designed to enhance the accuracy of classifiers. The novelty here is examining various classifiers combinations and techniques and thus crave out a design that compensates for the issue of under-fitting and over-fitting and provides the most accurate predictive output.

Predictive accuracy can be achieved by developing a simple but successful method based on the theory of diversity to direct the process of generating the optimal combination of classifiers, that does not suffer from under-fitting, over-fitting, and overgeneralization issues, leading to better accuracy of expected outputs. Consequently, the usage of a combination of various classifiers will increase prediction accuracy by excluding the limitations of certain data knots and incorporating the strengths of other data knots to obtain an accurate prediction. Moreover, ensemble combination methods have a huge impact on the overall design outcome, in addition to the generalization capability of the design itself. Combination approaches are used to combine classifiers of the same type, thereby producing better results, and promoting different applications of both ML and pattern recognition. This motivates us to explore various aspects of combination methods, such as output types, level of operation, combination, and algebraic rules, as well as to explore different methods for classifier combination.

Sophisticated combination methods are currently utilized in science and technology, for more complicated problems, the more advanced the model should be to deliver the best prediction and decision-making outcomes. The motivation behind this study is becoming acquainted with some of the most contemporary combination methods. To offer a superior algorithm, two essential things should be taken care of: collective accuracy and decline in the error prediction. Ensemble methods are no doubt essential for the classification task. However, any ensemble design and every combination approach have its flaws and constraints when it extends to the real application.

The mounting assumption is that individual classifiers typically do not seek to strike a balance between fitting and generalization when they infer the model from data sets. Thus, the models they infer can suffer from problems of under-fitting, over-fitting, and overgeneralization, and this triggers their poor performance. For that reason, a novel weighted ensemble method is presented to counter the issue of under-fitting and over-fitting of individual classifiers and to get more comprehensive results of classification.

Over-fitting occurs when a model can accurately distinguish a data point that is very closely related to training data but does not work perfectly with the data that are not closely connected to the training data. On the other hand, under-fitting happens when the model is incapable to secure the underlying data pattern, or intuitively does not fit the data well adequately. Over-fitting and under-fitting result in bad estimates in new data sets. Over-generalization happens when a model erroneously claims to be able to accurately classify large amounts of data that are not closely related to the training data. In general, the proposed ensemble method intends to minimize the false-negative and false-positive rates and thus exploiting accuracy in a weighted fashion. The intended framework consists of the following phases: data processing, over-sampling, clustering of data, model building, weights application, model evaluation. All the processing is done by utilizing Matlab software.

Exploratory data analysis is conducted out before any structured modeling commences. As the first stage in every predictive modeling, the task is to understand the purpose and data of the modeling process. The dilemma addressed by the proposed ensemble method is thus identified, along with the anticipated results. In this process, an assessment of the currently current data is done, and the features of the data to deliver ensembles and all the necessary tools are explored. While this phase has a major impact on the superiority of the ensembles (following the popular " garbage in, garbage out" rule), and preparation of data has a substantial impact on the predictive capability of the model. In this work, we have noticed the number of examples, imbalanced ratio (IR), a total of attributes, etc. of each data set.

Imbalance data sets have an inequitable number of positive and negative class examples. Without over-sampling, when the data sets are passed on to these classifiers, they show poor performance, i.e. the classifiers could not be trained for the minority class. Therefore, data sets are over-sampled to generate synthetic samples for proper classification. Therefore, it is necessary to re-balance, imbalanced data sets before classification. In this paper, SMOTE and ADASYN are applied to over-sample the minority class. SMOTE balances the data distribution by synthetically generating minority class examples. This methodology is efficient as it creates the synthetic minority class instances that are plausible; the instances are relatively close in feature space to current minority instances. Neighbors from the nearest k neighbors are chosen arbitrarily. The default settings implementation uses five of the closest neighbors. This approach effectively forces the decision-making area of the minority class to be more general.

Data partition has a decisive effect on the quality of the final models. For example, if the training data set is small, the learning algorithms used to create the models will not be efficient to capture the underlying patterns. On the other hand, a small test data set will have restricted skills to evaluate the execution of models. In this study, for each data set, the cluster of positive and negative instances is made and then data is partitioned into 40:60 ratios for training and testing purposes from both clusters. Thus, both positive and negative classes have equal representation in training and testing data sets. The spawned data partitions are utilized as inputs to the next model building stage.

The model building process is an extremely essential step of the proposed WEM. A combination of multiple classifiers will likely outperform a single classifier. However, in some cases, the fusion of different models could only add to the complexity of the system without any change in performance. Thus, for the ensemble to be efficient, the errors made by its base models should not be highly correlated and the output of each base model should be appropriate. Another main design function is the selection of the optimal models for learning algorithms leading to the maximum performance of the individual models/classifiers. A combination of four classifiers (i.e. DT, KNN, SVM, and NN) is the ensemble for classification. Model evaluation is key in any predictive modeling journey. In ensemble predictive modeling, the relative performance and variety of classifiers must be thoroughly evaluated, which becomes even more important. Usually, some accuracy test is used to determine classifier efficacy. Nevertheless, accuracy can be measured differently, each with a minor discrepancy.

The evaluation measures are computed for each classifier after modeling of data sets. The results generated by each classifier are given random weights spawned by uniform distribution U(0,1). Therefore, in the end, the weighted average of all evaluation measures is obtained and thus optimal accuracy is attained by diminishing errors and retrieving maximum G, F, and Acc values. The performance of the intended method is compared with the individual classifier's results. It is seen that the proposed novel WEM works adequately improved than the individual classifiers and effectively tackle the issue of over-fitting, under-fitting, and over-generalization. The pseudo code of the proposed method is given.

4. Simulation Study

The performance of the intended technique is assessed with the help of a comparative evaluation by achieving all possible analogies with other traditional classifiers including DT, KNN, SVM, and NN. The comparisons are rendered by using two types of over-sampling methods ADASYN and SMOTE. Forty imbalanced data sets with varying imbalance ratio (IR) are taken from a well-know data sets repository KEEL (Alcalá-Fdez et al., 2011). The comprehensive details of the these data sets are provided in Table 1. All the data sets are partitioned by using the ratio of 40:60 for training and test sets respectively. For NN, ten hidden layers are applied.

Algorithm 6 Pseudo Code of Proposed WEM

```
Begin
    1. Process the Imbalance Data sets
    2. Generate Synthetic samples of minority class by SMOTE or ADASYN
    3. Generate clusters of positive and negative instances
    4. Model four classifiers i.e. DT, SVM, KNN and NN on training and testing datasets
    5. Generate random weights by U (0,1)
    6. Compute the evaluation measures by assigning following weights $w_i/\Sigma(w_i)$ for $i$=1,2,3,4
    7. Access the weighted results of all evaluation measures
end
```

Table 1: Data Sets Description

Data set	Abbreviation	IR	Data set	Abbreviation	IR
Glass1	D1	1.82	Wisconsin	D2	1.86
Pima	D3	1.87	Iris0	D4	2
Paw02	D5	5	Ecoli1	D6	9
Yeast1	D7	9.08	Glass1	D8	9.12
Glass2	D9	9.22	Yeast2	D10	9.35
Glass2	D11	10.29	Ecoli2	D12	10.59
Led7digit1	D13	10.97	Ecoli3	D14	11
Glass3	D15	11.59	Ecoli4	D16	12.28
Cleveland1	D17	12.62	Ecoli5	D18	13
Yeast3	D19	14.3	Page-Blocks1	D20	15.86
Dermatology1	D21	16.9	Zoo	D22	19.2
Glass4	D23	19.44	Shuttle1	D24	20.5
Shuttle2	D25	22	Yeast4	D26	22.1
Wine-Red1	D27	29.17	Poker1	D28	29.5
Yeast5	D29	30.57	Wine-White1	D30	32.6
Wine-Red2	D31	35.44	Ecoli6	D32	39.14
Wine-White2	D33	44	Wine-Red3	D34	46.5
Wine-White3	D35	58.28	Poker2	D36	58.4
Shuttle3	D37	66.67	Wine-Red4	D38	68.1
Poker3	D39	82	Poker4	D40	85.88

Table 2: Results of All Classifiers by using ADASYN

Data	DT			SVM			KNN			NN			WEM		
	G	F	Acc	G	F	Acc	G	F	Acc	G	F	Acc	G	F	Acc
D1	1	1	100	1	1	100	0.781	0.778	72.22	0.975	0.975	97.53	0.993	0.993	**99.23**
D2	1	1	100	1	1	100	0.9893	0.9892	98.89	1	1	100	0.9999	0.999	**99.99**
D3	1	1	100	0.712	0.715	71.23	0.986	0.986	98.63	1	1	100	0.993	0.993	**99.38**
D4	1	1	100	1	1	100	1	1	100	1	1	100	1	1	100
D5	1	1	100	0.726	0.772	74.04	1	1	100	1	1	100	0.998	0.998	**99.84**
D6	1	1	100	0.784	0.761	80.46	0.828	0.769	80.46	1	1	100	0.981	0.975	**97.93**
D7	1	1	100	1	1	100	0.950	0.942	94.59	1	1	100	0.999	0.999	**99.99**
D8	1	1	100	1	1	100	0.994	0.994	99.45	0.994	0.994	99.45	0.999	0.999	**99.96**
D9	0.896	0.890	87.87	0.938	0.942	93.94	0.961	0.960	95.96	0.952	0.951	94.94	0.953	0.953	**95.20**
D10	1	1	100	1	1	100	0.968	0.969	96.84	1	1	100	0.999	0.999	**99.97**
D11	1	1	100	1	1	100	0.995	0.995	99.52	1	1	100	0.999	0.999	**99.99**
D12	1	1	100	0.785	0.80	78.80	0.887	0.883	86.95	1	1	100	0.963	0.965	**96.30**
D13	1	1	100	1	1	100	0.925	0.923	91.80	1	1	100	0.999	0.999	**99.96**
D14	1	1	100	0.836	0.823	84.79	0.832	0.769	80.60	1	1	100	0.970	0.967	**96.21**
D15	1	1	100	1	1	100	1	1	100	1	1	100	1	1	100
D16	1	1	100	0.754	0.725	77.86	0.972	0.972	97.26	1	1	100	0.988	0.987	**98.96**
D17	1	1	100	0.839	0.829	84.29	0.974	0.974	97.38	1	1	100	0.990	0.990	**99.01**
D18	1	1	100	0.794	0.773	81.47	0.819	0.759	79.55	0.996	0.996	99.68	0.972	0.963	**96.89**
D19	1	1	100	1	1	100	0.998	0.998	99.80	1	1	100	0.999	0.999	**99.99**
D20	0.896	0.898	89.64	0.788	0.7721	73.25	0.974	0.974	97.36	0.987	0.987	98.68	0.969	0.971	**96.99**
D21	1	1	100	1	1	100	1	1	100	1	1	100	1	1	100
D22	1	1	100	1	1	100	0.952	0.950	94.82	1	1	100	0.999	0.999	**99.90**
D23	0.852	0.842	81.34	0.822	0.859	83.73	0.981	0.981	98.08	0.934	0.932	92.82	0.964	0.966	**96.39**
D24	1	1	100	1	1	100	1	1	100	0.993	0.993	99.32	0.999	0.999	**99.97**
D25	1	1	100	0.988	0.988	98.86	1	1	100	1	1	100	0.999	0.999	**99.98**
D26	1	1	100	1	1	100	0.977	0.977	97.72	1	1	100	0.999	0.999	**99.96**
D27	1	1	100	1	1	100	0.986	0.986	98.64	1	1	100	0.999	0.999	**99.97**
D28	1	1	100	0.996	0.996	99.64	1	1	100	1	1	100	0.999	0.999	**99.99**
D29	1	1	100	1	1	100	0.999	0.999	99.90	1	1	100	0.999	0.999	**99.99**
D30	1	1	100	0.968	0.970	96.93	0.989	0.989	98.98	1	1	100	0.998	0.998	**99.86**
D31	1	1	100	0.969	0.970	96.98	0.963	0.963	96.32	1	1	100	0.997	0.997	**99.72**
D32	1	1	100	0.941	0.944	94.22	0.919	0.916	90.88	1	1	100	0.991	0.991	**99.11**
D33	1	1	100	0.819	0.803	83.50	0.951	0.944	94.78	1	1	100	0.986	0.984	**98.58**
D34	1	1	100	0.996	0.996	99.60	0.978	0.978	97.81	1	1	100	0.998	0.998	**99.81**
D35	1	1	100	0.850	0.839	86.02	0.928	0.915	92.12	1	1	100	0.988	0.986	**98.84**
D36	1	1	100	1	1	100	1	1	100	1	1	100	1	1	100
D37	1	1	100	0.989	0.989	98.95	1	1	100	0.999	0.999	99.97	0.999	0.999	**99.98**
D38	1	1	100	1	1	100	0.995	0.995	99.50	1	1	100	0.999	0.999	**99.99**
D39	1	1	100	0.944	0.942	94.58	0.945	0.937	94.10	1	1	100	0.989	0.988	**98.89**
D40	1	1	100	1	1	100	1	1	100	1	1	100	1	1	100

4.1. Performance Evaluation Measures

Three commonly used performance evaluation measures including G-mean (G), F-measure (F), and accuracy (acc) are computed to assess the enactment of all the classifiers (see Eq 3). G is the square root of the sensitivity and specificity of the classifiers whereas, F is computed with the help of the following formulas:

$$\begin{aligned} G &= \sqrt{Sensitivity * Specificity} \\ F &= \frac{2 * Percision * Sensitivity}{(Percision + Sensitivity)} \\ acc &= \frac{total \quad number \quad of \quad correct \quad predictions}{total \quad number \quad of \quad data \quad sets} \end{aligned} \tag{3}$$

And Acc demonstrates the exact identification of minority and majority class instances. All the data sets are over-sampled by using ADASYN and SMOTE. Subsequently, over-sampled data sets are classified by using the above mentioned four classifiers (DT, KNN, SVM, and NN). G, F, and, Acc of these data sets are monitored and are shown in Table 2. The over-fitting and under-fitting behaviors of conventional classifiers incorporating DT, SVM, KNN, and NN can be noticed from Tables 2 and 3. For most of the data sets with ADASYN, over-fitting is noticed by DT. Amongst all the classifiers, KNN showed the under-fitting classification results for most of the data sets. It is seen from Table 2 that SVM and NN demonstrated a mixed behavior of over-fitting and under-fitting. However, the proposed method (WEM) performed exceptionally by generating the well-balanced classification results without under-fitting and over-fitting for 35 imbalanced data sets out of 40 with varying IR's. Table 3 revealed the classification results of all classifiers involving their G, F, and Acc with SMOTE. Subsequent carefully monitoring the results of Table 3, it can be said that the results generated by SMOTE are substantially different from ADASYN (see Table 2). This time we observed the over-fitted results by the majority of the conventional classifiers involving (DT and NN). However, this

Table 3: Results of All Classifiers by using SMOTE

Data	DT			SVM			KNN			NN			WEM		
	G	F	Acc	G	F	Acc	G	F	Acc	G	F	Acc	G	F	Acc
D1	1	1	100	1	1	100	0.982	0.982	97.88	1	1	100	0.999	0.999	**99.98**
D2	1	1	100	0.991	0.993	99.27	0.995	0.995	99.43	1	1	100	0.998	0.999	**99.89**
D3	1	1	100	0.814	0.910	87.55	0.993	0.994	99.29	1	1	100	0.998	0.999	**99.88**
D4	1	1	100	1	1	100	1	1	100	1	1	100	1	1	100
D5	1	1	100	0.684	0.982	96.71	1	1	100	1	1	100	0.996	0.999	**99.96**
D6	1	1	100	0.972	0.966	98.53	0.939	0.937	97.07	0.991	0.991	99.63	0.992	0.991	**99.62**
D7	1	1	100	1	1	100	1	1	100	1	1	100	1	1	100
D8	1	1	100	1	1	100	0.997	0.990	99.55	1	1	100	0.999	0.999	**99.99**
D9	0.742	0.710	82.4	0.914	0.771	87.2	0.878	0.870	93.6	1	1	100	0.939	0.917	**95.46**
D10	1	1	100	1	1	100	0.993	0.993	99.71	1	1	100	0.999	0.999	**99.97**
D11	1	1	100	1	1	100	0.997	0.990	99.59	1	1	100	0.999	0.999	**99.99**
D12	1	1	100	0.957	0.902	95.98	0.985	0.977	99.10	1	1	100	0.997	0.995	**99.81**
D13	1	1	100	1	1	100	1	1	100	1	1	100	1	1	100
D14	1	1	100	0.955	0.948	98.13	1	1	100	1	1	100	0.999	0.999	**99.98**
D15	1	1	100	1	1	100	1	1	100	1	1	100	1	1	100
D16	1	1	100	0.954	0.909	96.81	0.985	0.980	99.31	1	1	100	0.994	0.992	**99.72**
D17	1	1	100	0.909	0.829	93.54	1	1	100	1	1	100	0.998	0.996	**99.87**
D18	1	1	100	0.955	0.948	98.37	1	1	100	1	1	100	0.998	0.998	**99.96**
D19	1	1	100	1	1	100	1	1	100	1	1	100	1	1	100
D20	1	1	100	0.577	0.5	90.78	0.921	0.918	97.53	1	1	100	0.971	0.969	**99.13**
D21	1	1	100	0.998	0.991	99.78	1	1	100	1	1	100	0.999	0.999	**98.99**
D22	1	1	100	1	1	100	0.912	0.909	97.67	1	1	100	0.999	0.999	**99.99**
D23	1	1	100	1	1	100	0.774	0.75	92.30	1	1	100	0.997	0.997	**99.93**
D24	1	1	100	0.971	0.971	99.39	1	1	100	1	1	100	0.999	0.999	**99.99**
D25	1	1	100	0.658	0.604	94.22	1	1	100	1	1	100	0.987	0.985	**99.79**
D26	1	1	100	1	1	100	1	1	100	1	1	100	1	1	100
D27	1	1	100	0.996	0.998	99.94	0.999	0.990	99.84	1	1	100	0.999	0.999	**99.99**
D28	1	1	100	0.998	0.979	99.66	1	1	100	1	1	100	0.999	0.999	**99.98**
D29	1	1	100	1	1	100	1	1	100	0.912	0.909	98.44	0.999	0.999	**99.99**
D30	1	1	100	0.978	0.789	96.09	1	1	100	1	1	100	0.999	0.994	**99.89**
D31	1	1	100	0.999	0.990	99.87	1	1	100	1	1	100	1	0.999	100
D32	1	1	100	0.993	0.913	98.84	1	1	100	0.977	0.979	99.71	0.997	0.996	**99.96**
D33	1	1	100	0.801	0.709	97.11	1	1	100	1	1	100	0.999	0.999	**99.99**
D34	1	1	100	0.990	0.981	99.80	1	1	100	1	1	100	0.999	0.999	**99.99**
D35	1	1	100	0.847	0.805	98.55	0.998	0.972	99.77	1	1	100	0.998	0.992	**99.94**
D36	1	1	100	1	1	100	1	1	100	1	1	100	1	1	100
D37	1	1	100	0.989	0.989	99.93	1	1	100	1	1	100	0.999	0.999	**99.99**
D38	1	1	100	1	1	100	1	1	100	1	1	100	1	1	100
D39	1	1	100	1	1	100	1	1	100	1	1	100	1	1	100
D40	1	1	100	1	1	100	1	1	100	1	1	100	1	1	100

Table 4: Description of Noisy Borderline Data Sets

Data set	Abbreviation	IR	Data set	Abbreviation	IR
Paw02a-800-7-60-BI	BD1	7	04clover5z-800-7-60-BI	BD2	7
04clover5z-800-7-50-BI	BD3	7	03subcl5-800-7-0-BI	BD4	7
Paw02a-800-7-70-BI	BD5	7			

time, SVM and KNN produced diverse results, even though mostly preceded to the over-fitting issue, particularly, from NN. Once again, the proposed method (WEM) performed well on 29 out of 40 imbalanced data sets by overcoming the impact of under-fitting and over-fitting issues of classification results. These results by WEM are highlighted in Table 3.

4.2. Performance of the proposed method with Noisy and Borderline Imbalanced data sets

The performance of the proposed method (WEM) is also explored on noisy borderline data sets. Borderline examples are categorized as examples located in the area surrounding the class boundaries, where the minority and the majority classes coincide (Saez et al., 2015). Six noisy borderline data sets are taken from KEEL (reference) and their complete details including their names, abbreviations used, and IR are provided in Table 4. The problem of under-fitting and over-fitting is observed by above mentioned traditional classifiers after applying two types of over-sampling techniques ADASYN and SMOTE. The results of all classifiers including the proposed method WEM are shown in Table 5.

It is obvious from the table that except SVM, the remaining three traditional classifiers including DT, KNN, and NN showed over-fitted results to these data sets. SVM showed its poor performance in terms of under fitted results to these data sets. However, WEM showed improved results by overcoming the over-fitting and under-fitting problems of these classifiers. Almost the same behaviors of classifiers can also be seen with SMOTE (see Table 6). The classification results obtained by conventional classifiers are unsatisfactory. DT, KNN, and NN constantly scored 1 for G and F measures and thus provided maximum accuracy. These classifiers delivered over-fitted results which are also not well

Table 5: Results of All Classifiers by using ADASYN

Data	DT			SVM			KNN			NN			WEM		
	G	F	Acc	G	F	Acc	G	F	Acc	G	F	Acc	G	F	Acc
BD1	1	1	100	0.785	0.812	79.24	1	1	100	1	1	100	0.994	0.995	**99.50**
BD2	1	1	100	0.708	0.766	72.77	1	1	100	1	1	100	0.996	0.997	**99.68**
BD3	1	1	100	0.762	0.802	77.49	1	1	100	1	1	100	0.997	0.997	**99.75**
BD4	1	1	100	0.508	0.728	62.78	1	1	100	1	1	100	0.977	0.987	**98.26**
BD5	1	1	100	0.803	0.820	80.72	1	1	100	1	1	100	0.995	0.995	**99.52**

Table 6: Results of All Classifiers by using SMOTE

Data	DT			SVM			KNN			NN			WEM		
	G	F	Acc	G	F	Acc	G	F	Acc	G	F	Acc	G	F	Acc
BD1	1	1	100	0.636	0.986	97.38	1	1	100	1	1	100	0.997	0.999	**99.98**
BD2	1	1	100	0.357	0.983	96.68	1	1	100	1	1	100	0.999	0.999	**99.99**
BD3	1	1	100	0.512	0.986	97.38	1	1	100	1	1	100	0.993	0.999	**99.96**
BD4	1	1	100	0.795	0.991	98.36	1	1	100	1	1	100	0.997	0.999	**99.97**
BD5	1	1	100	0.584	0.987	97.56	1	1	100	1	1	100	0.995	0.999	**99.97**

generalized and hence not acceptable. SVM has demonstrated poor performance as its results are under fitted both with ADASYN and SMOTE. SVM got minimum values for G and F measures with ADASYN implying that it has not performed well on minority class. SVM with SMOTE also shown worse performance for BD1, BD4, and BD5 resulting in zero score for G measure, and high values of F measure indicating the bias in SVM. Our proposed method, WEM produced excellent results with both ADASYN and SMOTE and has countered the over-fitting and under-fitting issues of four classifiers. The results obtained by WEM are well generalized and hence more acceptable.

5. Concluding Remarks

Imbalanced datasets continuously acquired the extraordinary consideration of the investigators as these data sets frequently occur in real life, predominantly in the medical sciences. The enactment of conventional ML algorithms, to knob these types of data sets are inspiring. However, few problems are unmovingly accompanying these algorithms which cannot be snubbed while doing the classification tasks of these data sets. Amongst these, the problem of over-fitting and under-fitting are two substantial matters that the investigators have to face while the classification chore of imbalanced data sets. Thus, the provocation of this work is to knob this problem by implementing some sophisticated methods. Therefore, a novel WEM is intended in this study to diminish the effect of these two substantial problems and to obtain improved accuracy for these types of data sets. The projected method ensembles the measures obtained by conventional classifiers with the help of the weights generated by the uniform distribution. The performance evaluation of the novel method is assessed by employing imbalanced data sets and noisy borderline data sets, in a comparative study by making the comparisons with the performance measures obtained from existing classifiers. It is noticed that DT, KNN, and NN are highly over-fitted, whereas SVM showed poor results i.e. under fitted results for forty imbalanced data sets both with ADASYN and SMOTE. Overall, our proposed method effectively diminished the effect of this over-fitting and under-fitting of imbalanced data sets and also shown enhanced accuracy for various types of imbalanced data sets.

References

1. Abdi, H. (1994). A neural network primer. *Journal of Biological Systems*, 2(03):247–281.
2. Akbani, R., Kwek, S., and Japkowicz, N. (2004). Applying support vector machines to imbalanced datasets. In *European conference on machine learning*, pages 39–50. Springer.
3. Alcalá-Fdez, J., Fernández, A., Luengo, J., Derrac, J., García, S., Sánchez, L., and Herrera, F. (2011). Keel data-mining software tool: data set repository, integration of algorithms and experimental analysis framework. *Journal of Multiple-Valued Logic & Soft Computing*, 17.
4. Anyanwu, M. N. and Shiva, S. G. (2009). Comparative analysis of serial decision tree classification algorithms. *International Journal of Computer Science and Security*, 3(3):230–240.
5. Barandela, R., Sánchez, J. S., Garca, V., and Rangel, E. (2003). Strategies for learning in class imbalance problems. *Pattern Recognition*, 36(3):849–851.
6. Bennett, K. P. and Blue, J. (1998). A support vector machine approach to decision trees. In *1998 IEEE International Joint Conference on Neural Networks Proceedings. IEEE World Congress on Computational Intelligence (Cat. No. 98CH36227)*, volume 3, pages 2396–2401. IEEE.
7. Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P. (2002). Smote: synthetic minority oversampling technique. *Journal of artificial intelligence research*, 16:321–357.
8. Chawla, N. V., Japkowicz, N., and Kotcz, A. (2004). Special issue on learning from imbalanced data sets. *ACM SIGKDD explorations newsletter*, 6(1):1–6.
9. Dasarathy, B. V. and Sheela, B. V. (1979). A composite classifier system design: Concepts and methodology. *Proceedings of the IEEE*, 67(5):708–713.
10. Estabrooks, A., Jo, T., and Japkowicz, N. (2004). A multiple resampling method for learning from imbalanced data sets. *Computational intelligence*, 20(1):18–36.
11. Freund, Y. and Mason, L. (1999). The alternating decision tree learning algorithm. In *icml*, volume 99, pages 124–133.
12. Galar, M., Fernández, A., Barrenechea, E., and Herrera, F. (2013). Eusboost: Enhancing ensembles for highly imbalanced data-sets by evolutionary undersampling. *Pattern recognition*, 46(12):3460–3471.
13. Han, H., Wang, W.-Y., and Mao, B.-H. (2005). Borderline-smote: a new over-sampling method in imbalanced data sets learning. In *International conference on intelligent computing*, pages 878–887. Springer.
14. Hansen, L. K. and Salamon, P. (1990). Neural network ensembles. *IEEE transactions on pattern analysis and machine intelligence*, 12(10):993–1001.
15. He, H., Bai, Y., Garcia, E. A., and Li, S. (2008). Adasyn: Adaptive synthetic sampling approach for imbalanced learning. In *2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence)*, pages 1322–1328. IEEE.
16. He, H. and Garcia, E. A. (2009). Learning from imbalanced data. *IEEE Transactions on knowledge and data engineering*, 21(9):1263–1284.
17. Hsu, K.-W. (2017). A theoretical analysis of why hybrid ensembles work. *Computational intelligence and neuroscience*, 2017.
18. Hu, S., Liang, Y., Ma, L., and He, Y. (2009). Msmote: Improving classification performance when training data is imbalanced. In *2009 second international workshop on computer science and engineering*, volume 2, pages 13–17. IEEE.
19. Kaur, P. and Gosain, A. (2018). Comparing the behavior of oversampling and undersampling approach of class imbalance learning by combining class imbalance problem with noise. In *ICT Based Innovations*, pages 23–30. Springer.
20. Kong, J., Rios, T., Kowalczyk, W., Menzel, S., and Bäck, T. (2020). On the performance of oversampling techniques for class imbalance problems. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 84–96. Springer.
21. Kubat, M., Holte, R. C., and Matwin, S. (1998). Machine learning for the detection of oil spills in satellite radar images. *Machine learning*, 30(2-3):195–215.
22. Laurikkala, J. (2001). Improving identification of difficult small classes by balancing class distribution. In *Conference on Artificial Intelligence in Medicine in Europe*, pages 63–66. Springer.
23. Leevy, J. L., Khoshgoftaar, T. M., Bauder, R. A., and Seliya, N. (2018). A survey on addressing high-class imbalance in big data. *Journal of Big Data*, 5(1):42.
24. Lemaître, G., Nogueira, F., and Aridas, C. K. (2017). Imbalanced-learn: A python toolbox to tackle the curse

of imbalanced datasets in machine learning. *The Journal of Machine Learning Research*, 18(1):559–563.

25. Lewis, D. D. and Catlett, J. (1994). Heterogeneous uncertainty sampling for supervised learning. In *Machine learning proceedings 1994*, pages 148–156. Elsevier.
26. Li, Y. and Zhang, X. (2011). Improving k nearest neighbor with exemplar generalization for imbalanced classification. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 321–332. Springer.
27. 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)*, 39(2):539–550.
28. Liu, X.-Y. and Zhou, Z.-H. (2013). Ensemble methods for class imbalance learning. *Imbalanced Learning: Foundations, Algorithms and Applications*, pages 61–82.
29. Mathew, J., Pang, C. K., Luo, M., and Leong, W. H. (2017). Classification of imbalanced data by oversampling in kernel space of support vector machines. *IEEE transactions on neural networks and learning systems*, 29(9):4065–4076.
30. Paing, M. P., Pintavirooj, C., Tungjitkusolmun, S., Choomchuay, S., and HAMAMOTO, K. (2018). Comparison of sampling methods for imbalanced data classification in random forest. In *2018 11th Biomedical Engineering International Conference (BMEiCON)*, pages 1–5. IEEE.
31. Panchal, G., Ganatra, A., Shah, P., and Panchal, D. (2011). Determination of over-learning and over-fitting problem in back propagation neural network. *International Journal on Soft Computing*, 2(2):40–51.
32. Pattanayak, S. S. and Rout, M. (2018). Experimental comparison of sampling techniques for imbalanced datasets using various classification models. In *Progress in Advanced Computing and Intelligent Engineering*, pages 13–22. Springer.
33. Pedersen, R. and Schoeberl, M. (2006). An embedded support vector machine. In *2006 International Workshop on Intelligent Solutions in Embedded Systems*, pages 1–11. IEEE.
34. Piotrowski, A. P. and Napiorkowski, J. J. (2013). A comparison of methods to avoid overfitting in neural networks training in the case of catchment runoff modelling. *Journal of Hydrology*, 476:97–111.
35. Polikar, R. (2006). Ensemble based systems in decision making. *IEEE Circuits and systems magazine*, 6(3):21–45.
36. Rokach, L. (2010). Ensemble-based classifiers. *Artificial intelligence review*, 33(1-2):1–39.
37. Saez, J., Luengo, J., Stefanowski, J., and Herrera, F. (2015). Addressing the noisy and borderline examples problem in classification with imbalanced datasets via a class noise filtering method-based re-sampling technique. *Inform Sci*, 291:184–203.
38. Saxena, R. (2017). How decision tree algorithm works. *URl: http://dataaspirant. com/2017/01/30/how-decision-tree-algorithm-works/.(accessed: 2019-01-28).*
39. Schclar, A., Tsikinovsky, A., Rokach, L., Meisels, A., and Antwarg, L. (2009). Ensemble methods for improving the performance of neighborhood-based collaborative filtering. In *Proceedings of the third ACM conference on Recommender systems*, pages 261–264.
40. Seiffert, C., Khoshgoftaar, T. M., Van Hulse, J., and Napolitano, A. (2009). Rusboost: A hybrid approach to alleviating class imbalance. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 40(1):185–197.
41. Tan, P.-N., Steinbach, M., and Kumar, V. (2016). *Introduction to data mining*. Pearson Education India.
42. Tavallaee, M., Stakhanova, N., and Ghorbani, A. A. (2010). Toward credible evaluation of anomaly-based intrusion-detection methods. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 40(5):516–524.
43. Tay, B., Hyun, J. K., and Oh, S. (2014). A machine learning approach for specification of spinal cord injuries using fractional anisotropy values obtained from diffusion tensor images. *Computational and mathematical methods in medicine*, 2014.
44. Yang, Z., Tang, W., Shintemirov, A., and Wu, Q. (2009). Association rule mining-based dissolved gas analysis for fault diagnosis of power transformers. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 39(6):597–610.
45. Ying, X. (2019). An overview of overfitting and its solutions. In *Journal of Physics: Conference Series*, volume 1168, page 022022. IOP Publishing.
46. Zhang, J. and Chen, L. (2019). Clustering-based undersampling with random over sampling examples and support vector machine for imbalanced classification of breast cancer diagnosis. *Computer Assisted Surgery*, 24(sup2):62–72.
47. Zhang, Y. and Wang, D. (2013). A cost-sensitive ensemble method for class-imbalanced datasets. In *Abstract*

and applied analysis, volume 2013. Hindawi.

48. Zhou, Z.-H. (2012). *Ensemble methods: foundations and algorithms*. CRC press.