Taylor & Francis
Taylor & Francis Group

Applied Artificial Intelligence
An International Journal

ISSN: 0883-9514 (Print) 1087-6545 (Online) Journal homepage: https://www.tandfonline.com/loi/uaai20

FF-SMOTE: A Metaheuristic Approach to Combat Class Imbalance in Binary Classification

Prabhjot Kaur & Anjana Gosain

To cite this article: Prabhjot Kaur & Anjana Gosain (2019): FF-SMOTE: A Metaheuristic Approach to Combat Class Imbalance in Binary Classification, Applied Artificial Intelligence, DOI: 10.1080/08839514.2019.1577017

To link to this article: https://doi.org/10.1080/08839514.2019.1577017

Published online: 18 Feb 2019.

Submit your article to this journal

Article views: 4

View Crossmark data

APPLIED ARTIFICIAL INTELLIGENCE
https://doi.org/10.1080/08839514.2019.1577017

FF-SMOTE: A Metaheuristic Approach to Combat Class Imbalance in Binary Classification

Prabhjot Kaur[a] and Anjana Gosain[b]

[a]Research Scholar in USICT, GGSIP University and working as an Associate Professor in the Department of Information Technology, Maharaja Surajmal Institute of Technology, New Delhi, India; [b]Department of Information Technology, USICT, Guru Gobind Singh Indraprastha University, New Delhi, India

ABSTRACT

Nature inspired intelligent computation-based algorithms have been grown remarkably over the previous years. These algorithms are applied for optimizing the values to a number of problem areas ranging from scientific research to industry or commerce. Class imbalance is a challenging problem of classification to identify smaller class when dealing with skewed distributions. This paper proposed a firefly-based oversampling technique to combat class imbalance in binary classification. The proposed technique is applied on 10 UCI data sets with the imbalance ratio ranging from high to low and is compared with the other state-of-the art oversampling techniques. The performance of the proposed method is assessed through performance metrics area under the curve and geometric mean. The techniques are also analyzed statistically using Friedman and Wilcoxon matched signed rank Test. Through experimental and statistical analysis, it is reported that the proposed technique outperformed other oversampling techniques.

Introduction

In human history, our approach to problem-solving has always been metaheuristic. Alan Turing was the first person to use heuristic algorithms during the Second World War when he was breaking German Enigma ciphers at Bletchley Park. Heuristic algorithms solve the problem by trial and error. Metaheuristic generally works better than simple heuristics because they use the principle of randomization and local search. Randomization is a good way to shift from local search to the search at the global level. Metaheuristic techniques can be divided into population-based and trajectory-based techniques. Population-based algorithms use multiple agents to search space and to find optimized solutions whereas trajectory-based algorithms use a single agent which moves through the

CONTACT Prabhjot Kaur thisisprabhjot@gmail.com Research Scholar in USICT, GGSIP University and working as an Associate Professor in the Department of Information Technology, Maharaja Surajmal Institute of Technology, C-4, Janakpuri, New Delhi 110058, India

Color versions of one or more of the figures in the article can be found online at www.tandfonline.com/uaai.

design space to solve the problem (Yang 2010). In some applications, heuristic and metaheuristics terms have used interchangeably.

Many metaheuristic algorithms have been reported in history. Firstly, John Holland with the help of his team developed genetic algorithms (GA) at the University of Michigan in the 1960s and 1970s. GA was based upon the Darwinian evolution and natural selection of biological systems. The next big step was simulated annealing, which was developed in 1983 by S. Kirkpatrick, C. D. Gellat, and M. P. Vecchi. In 1992, Marco Dorigo developed Ant Colony optimization, which is motivated by the swarm intelligence of social ants using pheromone as a chemical messenger. In 1995, another algorithm with the name particle Swarm optimization (PSO) was developed by James Kennedy and R. C. Eberhart. PSO is inspired by the swarm intelligence of fishes and birds. R. Stoen and K. Price developed Differential Evolution in 1997, which is considered as more efficient than GA for some applications (Yang 2010).

More new developments have been proposed with the start of the twenty-first century. In 2001, Z. W. Germ et al. developed Harmony search followed by Honey bee in 2004 by S. Nakrani and C. Tovey. In 2008, X. S. Yang proposed Firefly Algorithm (FA). It became very popular in a short time and can be applied for solving the hardest optimization problems (Fister et al. 2013). In 2009, X. S. Yang at Cambridge University and S. Deb in India proposed Cuckoo search (Yang 2010). Bat algorithm was proposed by X. S. Yang in 2010.

Metaheuristic algorithms have been used in many engineering applications including image processing, antenna design, wireless networks, industrial optimization, robotics, semantic web, etc. (Fister et al. 2013). The current study has used FA to solve class imbalance in classification. Class imbalance is a critical problem of classification when we want to identify the rare cases from the data. Established classification algorithms could not work with the unbalanced data (Bunkhumpornpat, Sinapiromsaran, and Lursinsap 2009; Chawla et al. 2002; Garcia and Herrera 2009; Han, Wang, and Mao 2005; Hu et al., 2009; Stefanowski and Wilk 2008; Yen and Lee 2009; Ying 2013). Many solutions are proposed by the researchers to tackle class imbalance. Some researchers have tried to balance the data set as a preprocessing step so that same established algorithms can be used to classify the data. Such category of algorithms is called data-level methods (Bunkumpormpat, Sinapiromsaran & Lursinsap, 2009; Chawla et al. 2002; Galar et al., 2011; Garcia and Herrera 2009; Han, Wang, and Mao 2005; Hu et al. 2009; Stefanowski and Wilk 2008; Yen and Lee 2009; Ying 2013). Data level methods are further classified into two categories. Some methods are proposed to increase the size of a smaller class so that the data can be balanced before classification and this category is known as oversampling methods (Bunkhumpornpat, Sinapiromsaran, and Lursinsap 2009; Chawla et al. 2002; Han, Wang, and Mao 2005; Hu et al. 2009; Stefanowski and Wilk 2008; Ying 2013). The other category is known as undersampling wherein researchers have tried to remove the data points from the bigger class to balance the data (Garcia and Herrera 2009; Yen and Lee 2009). Another level of research is going on at the algorithm level wherein the internal structure of the method is modified to remove the sensitivity of algorithm toward the smaller class (Batuwita and Palade 2013; Chi, Yan, and Pam 1996; Cristianini et al., 2002; Fernandez et al. 2008; Hong, Chen, and Harris 2007; Iman, Ting, and Kamruzzaman 2006; Kandola and Shawe-Taylor 2003; Lin and Wang 2002; Wu and Amari 2002; Wu and Chang 2003a, 2003b, 2005). Many authors have combined these two methods (data level and algorithm level) to solve the class imbalance problem (Chawla, Lazarevic, Hall & Bowler, 2003; Guo & Victor, 2004; Kim 2013; Wang and Japkowicz 2010; Galar et al., 2013). In the current paper, we propose an oversampling technique, which uses FA to generate synthetic data points in the smaller class to re-balance the data before classification.

The paper is organized as follows: Section 'Related Techniques' briefly reviews the other oversampling methods which are used in this paper for comparison with the proposed method. Section 'Background Information' reviews the techniques used within the proposed algorithm. Next section describes the proposed method. After that Empirical evaluation is given followed by statistical validation.

Related Techniques

This section briefly explains the other oversampling techniques which have been compared with the proposed technique.

Synthetic Smaller Oversampling (SMOTE)

SMOTE was proposed by Chawla (2002). It is a popular oversampling technique which balances the data set by synthetically generating data points within the smaller class. It uses nearest neighbor concept and interpolation method to generate the data points synthetically. It randomly selects data points from the smaller class based upon the amount of oversampling and then selects the neighborhood points around them so that synthetic data points can be generated using interpolation method (Galar et al. 2011). It gives better results when combined with undersampling, which is done by randomly eliminating data points from the bigger class till the data set is balanced. One of the limitations of SMOTE is that it generates the data points blindly, i.e. without considering whether the random data points selected for generating further points are good or noisy (Galar et al. 2011).

In case the points are noisy, then they will only generate the noise points which in turn will degrade the performance of the classifier.

Borderline SMOTE (BL_SMOTE)

This method is a variation of SMOTE and was proposed by Han, Wang, and Mao (2005). It differs from the SMOTE in the sense that it only selects those data points from the smaller class which are close to the boundary of the smaller class. It is based on the concept that those data points which are on the borderline or near to it are more subject to misclassification. This technique tries to strengthen the boundary of the smaller class. In this method, the smaller class data points are divided into three classes: noise, borderline, and safe data points based upon the presence of other smaller class data points around them. It oversamples only those data points which are in the borderline region (Galar et al. 2011).

SafeLevel SMOTE (SL_SMOTE)

It is another variation of SMOTE which was proposed in 2009 by C. Bunkhumpornpat et al. Its strength is that SL_SMOTE carefully selects the data points from the smaller class by considering a safe level. Safe level of a data point depends upon the presence of other smaller class data points within its neighborhood. If the total number of data points in the neighborhood is close to '0', then that data point is not safe and is considered as noise. If the number is more than some specified 'n' number, then the data point is considered as safe (Galar et al. 2011). This method generates data points close to the safe level only.

Selective Pre-Processing of Imbalanced Data (SPIDER)

The technique was proposed by J. Stefanowski and S. Wilk in 2008. Like the earlier techniques, it also divides the smaller class data points into safe and noisy categories using nearest neighbor rule with heterogeneous value distance metric (HVDM) (Wilson and Martinez 2000). The technique works in two steps. In the first step, it identifies the data points as safe and noisy. It provides three types of labels to the data points. In the second step, it processes the data points as per their assigned labels (weak, strong and relabel). For every weak label data point, it amplifies smaller class data point; for relabel option, it relabels bigger class data point and amplifies smaller class data point. For a strong label data point, it again amplifies smaller class data point. After that, the leftover noisy points from the bigger class are deleted (Galar et al. 2011). It is different from SMOTE as it replicates the same values instead of generating the new one.

Background Information

This section briefly discusses the techniques used to develop the proposed algorithm.

The Fuzzy C Means Algorithm

Fuzzy C Means is the most popular clustering technique developed by Bezdek (1981). It is a supervised technique which clustered the data set as per the following equation:

$$J_{FCM} = \sum_{k=1}^{m} \sum_{l=1}^{n} u_{kl}^{m} d_{kl}^{2}$$

where d_{kl} can be any distance metric which specifies the distance between the center point and the data data point. u_{kl} is the membership of data point 'x_l' in cluster 'l', and it must satisfy the following relationship:

$$\sum_{l=1}^{m} u_{kl} = 1; \; l = 1, 2, \ldots..n$$

The Fuzzy C Means algorithm iteratively optimizes the objective function with the continuous update of membership and centers until some stopping criteria are met.

Firefly Algorithm

Firefly is a metaheuristic approach which was developed by X. S. Yang at Cambridge University (Yang 2008, 2009; Yang and He 2013). The algorithm is inspired by the flashing characteristics of fireflies. This algorithm assumes the following rules to calculate optimized values (Yang 2008):

(1) All the fireflies are of the same sex type and any firefly can attract towards the other.
(2) The amount of attractiveness among the flashing fireflies depends upon the brightness of their lights and it decreases by enhancing the distance between them.
(3) The firefly with less bright light attracts toward the one having high brightness of the light. But the fireflies with the same brightness move in the random fashion.

(4) The brightness of the firefly is determined by the objective function of the problem under consideration.

Firefly's attractiveness is based upon the light intensity observed by the adjacent fireflies and it is defined as follows:

$$\beta = \beta_0 e^{-\gamma r^2} \tag{1}$$

In the above equation, $'\gamma'$ is the absorption coefficient; 'r' is the distance; $'\beta_0^{'}$ is the attractiveness at r = 0. The distance between two fireflies a & b at locations x_a and x_b can be calculated as follows:

$$r_{ab} = \|x_a - x_b\| = \sqrt{\sum_{i=1}^{j} \left(x_{a,i} - x_{b,i}\right)^2}$$

In the 2-D space, the above equation is reduced as follows:

$$r_{ab} = \sqrt{(x_a - x_b)^2 + (y_a - y_b)^2}$$

The movement of firefly 'a' toward 'b' will happen as per the following equation:

$$x_a = x_a + \beta_0 e^{-\gamma r_{ab}^2}(x_b - x_a) + \alpha \left(rand - \frac{1}{2}\right) \tag{2}$$

In the above equation, term 2 is due to attraction and term 3 is due to randomization with 'α' as the randomization parameter. $'rand'$ is a random number generator whose value is uniformly distributed between 0 and 1. The value of $'\beta_0^{'}$ is 1. $\alpha \epsilon [0, 1]$. $'\gamma'$ is the absorption coefficient which determines the variation of the attractiveness and its value is actually important to regulate the speed of the convergence and to define the behavior of the FA. For maximum applications, its value varies from 0.1 to 10 [31].

The Propose Technique, FF-SMOTE

This section explains our proposal to combat class imbalance in binary classification. SMOTE (Chawla et al. 2002) is a popular oversampling technique which uses interpolation method to generate synthetic data points in the smaller class. This paper proposes modified SMOTE wherein we are using FA (Yang 2008; Yang and He 2013) to synthetically generate data points rather than the interpolation method. Firefly is a metaheuristic approach which is used to find out optimized values among a group of data points. Authors preferred firefly over other metaheuristic methods because Firefly can deal with multimodal functions naturally and efficiently and is very much effective in terms of real-time problems.

The proposed algorithm works by first selecting the amount of oversampling required. Based upon the amount of oversampling, number of data points are selected randomly from the smaller class. Firefly method is used by providing lower and upper bounds from the randomly selected data to generate optimized values within the smaller class. The flowchart of the proposed model is shown in Figure 1.

Original (Imbalanced) data set is clustered into smaller and bigger class using Fuzzy C Means algorithm. Then, the proposed technique, FF-SMOTE, is applied on the smaller class to generate the synthetic data points and to balance the data. After that, any traditional classifier can be used to classify the balanced data set. Pseudo-code of FF-SMOTE is given in Figure 2. The amount of oversampling is given as an input parameter to the proposed algorithm.

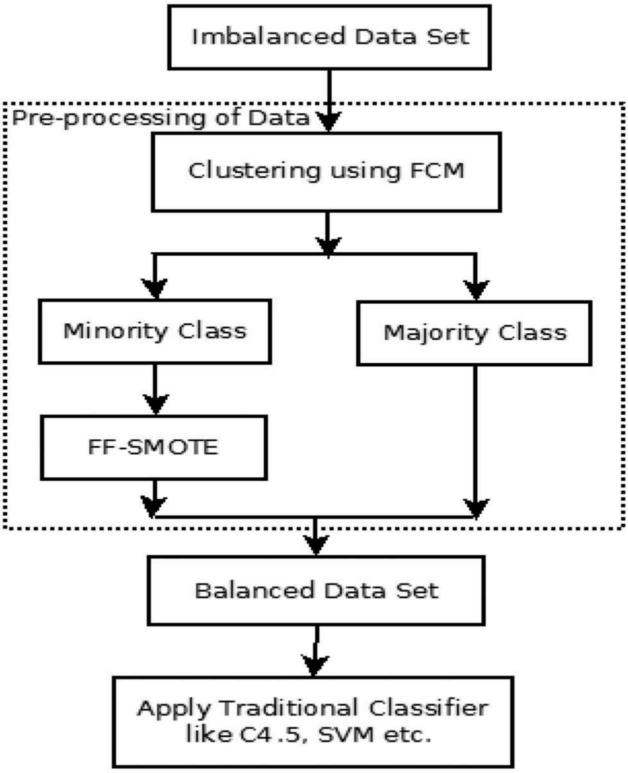

Figure 1. Proposed model for FF-SMOTE.

```
Algorithm: $FF - SMOTE(M, N)$
Input: Number of minority class instances $(M)$, percentage of Oversampling $(N\%)$
Output: balancedmin[ ][ ] Array (* balanced data − set.*)
Calculate the number of points to be generated synthetically by using formula:
        $R = (int)\ (M * {}^{N}/_{100})$   (* Number of points should be the integer value. *)
Min[ ][ ]: Array of original minority class instances.
Select 'R' number of random instances from Min.
Syndata[ ][ ]: Array of synthetic instances.
for i ← 1 to R
    call FF(i, Min, Syndata)
endfor
Combine Syndata & Min and store in balancedmin.

procedure FF(i, Min, Syndata)   (* Function to generate synthetic instances.*)
Define the objective function $f(x)$.
Initialize the number of fireflies 'F' and the number of generations 'G'.
Initialize the values of Randomness, Absorption co − efficient and Attractiveness.
Generate the initial population of fireflies, $X_i (i = 1,2,3, \ldots\ldots, F)$,
and their light intensities as $I_i$ at $X_i$.
while $G \neq 0$
        for $i \leftarrow 1$ to F (all fireflies)
            for $j \leftarrow 1$ to F (all fireflies)
                if $I_i > I_j$
                    then move firefly $X_i$ towards $X_j$ by using Eq. (2).
                endif
                Modify Attractiveness using Eq. (1).
                Evaluate new solutions and update light intensities.
            endfor j
        endfor i
        Rank the fireflies and find the current best.
        Store the current best value to Syndata.
        $G \leftarrow G - 1$
endwhile
return (* end of FF.*)
End of Pseudo Code
```

Figure 2. Pseudo-code of FF-SMOTE.

Empirical Evaluation

This section presents the setup used to empirically assess the capability of the proposed method and other state-of-the-art oversampling methods. First, we are using an imbalanced synthetic data set to demonstrate the working of propose method in Section 4.1. Then, we use 10 real-world imbalanced data sets to carry out the comparison between proposed and other oversampling methods.

Demonstration of FF_SMOTE Using Synthetic Data Set

We used MATLAB 2015a (Massachusetts 2013) and WEKA Tool (Mark et al. 2009) to demonstrate the working of the proposed method.

Figure 3 shows the synthetic data set which contains 254 data points. The number of data points in smaller and bigger class are 38 and 216, respectively, and the Imbalance ratio of the data set is 5.7.

Figure 4 shows the output of FF_SMOTE with 40% oversampling. Blue color stars show the synthetic data points that are generated by implemented FF_SMOTE within the smaller class. Figures 5 and 6 show the result with 60% and 80% oversampling, respectively. It is noticed from all the figures that synthetic data points are generated within the smaller class and no data point has been generated outside the class, which ensures that the properties of synthetic data points are same as that of smaller class data points.

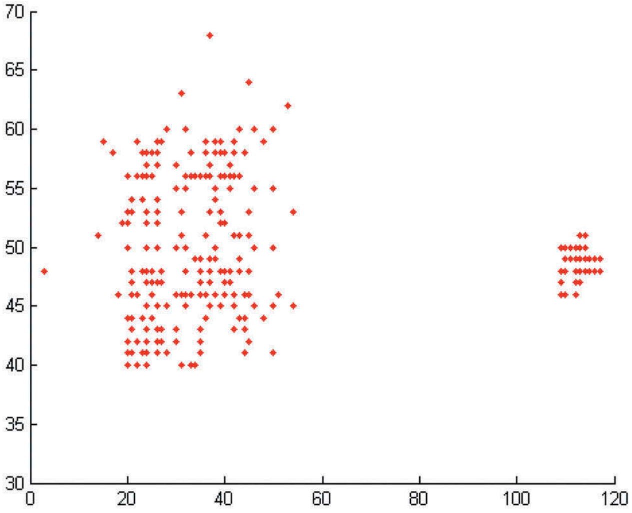

Figure 3. Synthetic data set.

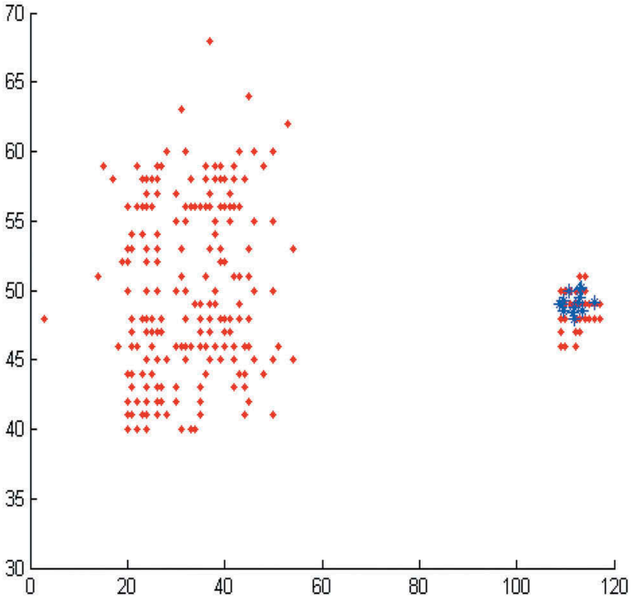

Figure 4. A 40% oversampling with the proposed method, FF-SMOTE.

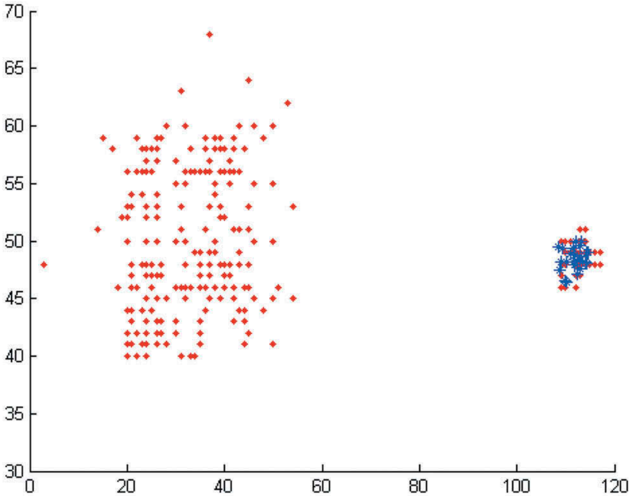

Figure 5. A 60% oversampling with the proposed method, FF-SMOTE.

Experimental Framework

In this section, the proposed method, FF_SMOTE, is assessed with 10 real-world imbalanced data sets and is compared with the popular state-of-the-art oversampling methods. We used KEEL Tool (Alcala-Fdez, Fernandez, Luengo, Derrac, Garcia, Sanchez & Herrera, 2011; Alcalafdez, Sanchez, Garcia, Del Jesus, Ventura Garrell, Otero, Romero, Bacardit, Rivas,

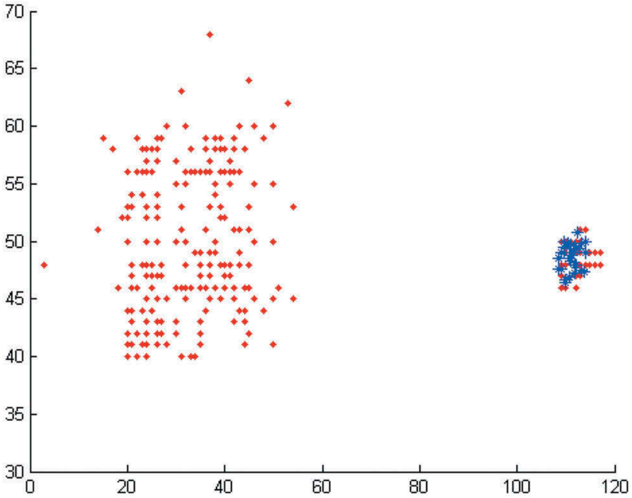

Figure 6. A 80% oversampling with the proposed method, FF-SMOTE.

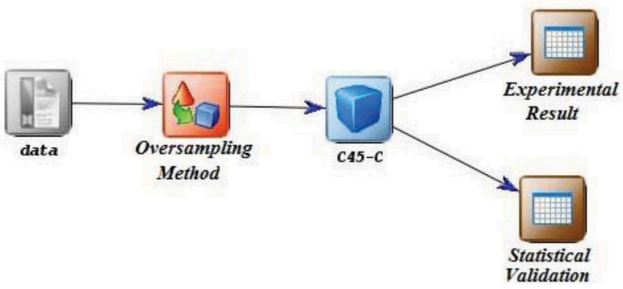

Figure 7. Setup used in KEEL software tool.

Farnandez & Herrera, 2008) to do the empirical calculations. The setup used for the experimentation is shown in Figure 7.

We used the C4.5 decision tree (Quinlan 1993) as the base classifier for all the experiments. The reason being it is a very popular traditional classifier and has been used majorly by the researchers to compare the techniques in imbalanced domains (Batista, Prati, and Monard 2004). Table 1 lists the oversampling methods and their terminology used in this paper for the comparison and Table 2 lists the initial parameters used for C4.5 and for other oversampling techniques.

Table 1. State-of-the-art oversampling methods.

Terminology	Method with description
SMOTE	Synthetic smaller oversampling
BL_SMOTE	BorderLine SMOTE
SL_SMOTE	Safe level SMOTE
SPIDER	Selective preprocessing of imbalanced data

Table 2. Initial parameters used for the base classifier and other oversampling methods.

Method	Parameters
C4.5	Pruning = true, confidence level = 0.25, data points per leaf = 2
SMOTE, BL-SMOTE, SL-SMOTE	Number of neighbors = 5, type = both, distance function = HVDM, type of interpolation = standard, α = 0.5, μ = 0.5
SPIDER	Number of neighbors = 3, distance function = HVDM
FF_SMOTE	Randomness 'α' = 0.1, attractiveness 'β' = 0.2, absorption coefficient 'γ' = 5, number of fireflies = 30, maximum generation = 20

Data Sets

We considered 10 real-world imbalanced data sets, which are publicly available with the KEEL software. As we are dealing with the binary classification, various multiclass data sets are converted to the binary classes by joining multiple classes as positive or negative classes. Table 3 lists the properties of these data sets ranging from highly imbalanced to the low imbalance. We have used the data set with 5-fold stratified cross-validation. The same setting of data is available with the KEEL software tool, so any interested researcher can reproduce the experimental results.

Performance Criteria

There are many performance metrics which are used for imbalance domains to evaluate the performance of methods. Confusion matrix is an important matrix which helps us to define various performance measures used in imbalance domains. It is actually the record of various incorrectly and correctly detected data points of the class as shown in Table 4.

Different outcomes that can be generated from this matrix to evaluate the performance of methods are:

(1) True positive rate, TPR $TP/(TP+FN)$
(2) True negative rate, TNR $TN/(FP+TN)$
(3) False positive rate, FPR $FP/(FP+TN)$
(4) False negative rate, FNR $FN/(TP+FN)$

Table 3. Properties of data sets.

SN.	Name	No. of data points	Dimensions	%age of smaller class	Imbalance ratio
1	Abalone	4174	8	0.77	128.87
2	Glass	214	9	4.2	22.81
3	Ecoli	336	7	6.54	13.84
4	Yeast	459	8	6.75	13.87
5	PageBlock	5472	10	10.21	8.79
6	Segment	2308	19	14.25	6.02
7	Vehicle	846	18	25.05	2.99
8	Iris	150	4	33.3	2
9	Wisconsin	699	9	34.47	1.9
10	Pima	768	8	34.89	1.87

Table 4. Confusion matrix for binary classification.

	Correctly detected	Incorrectly detected
Smaller class (Ppsitive)	True positive (TP)	False negative (FN)
Bigger class (negative)	False positive (FP)	True negative (TN)

The limitation of the abovementioned metrics is that they individually do not assess the technique well; we need a combination of these to assess any method. So other metrics are defined by combining these metrics. Area under the curve (AUC) and Receiver Operating Characteristic (ROC) are very well-known performance metrics used in the imbalance domain to evaluate the performance of techniques (Green and Swets 1966; Spacman 1989). ROC is the curve wherein the false positive rate is plotted on x-axis and true-positive rate on y-axis. AUC is the quantitative representation of ROC curve (Hanley and McNeil 1983; Metz 1978) and it is defined as follows:

$$AUC = \frac{1 + TPR - FPR}{2}$$

G-mean (Geometric mean) (Barandela, Sanchez, Garcia & Rangel, 2003; Galar et al. 2011) is the geometric mean of the accuracy of classes and is recently used by the papers to evaluate the performance of methods. It is defined as follows:

$$G - Mean = \sqrt{TPR.TNR}$$

As there is no standard criteria which are reported in the literature to assess the performance of methods designed for imbalanced domains, we used AUC and G-mean to assess the performance of the proposed method. Table 5 lists the readings for AUC and G-mean. Best readings are highlighted in bold. Although SMOTE gave best results for Glass and Ecoli, SPIDER gave best results for Yeast data set but FF_SMOTE performed in a better way in rest of the data sets.

Statistical Validation

In our study, we are comparing five algorithms which are applied on 10 real data sets. In such a scenario, the best option is to do statistical analysis for the appropriate comparison. There are mainly two statistical inferential tests, namely, parametric and non-parametric. Most of the parametric tests rely on the assumption that data have come from the similar type of distribution whereas non-parametric tests do not depend upon the data that belongs to

Table 5. Results of performance criteria for all the methods.

			FF_SMOTE		SMOTE		BL-SMOTE		SL-SMOTE		SPIDER	
SN.	Data set	Imbalance ratio	AUC	G-mean	AUC	G-mean	AUC	G-mean	AUC	G-mean	AUC	G-mean
1	Abalone	128.87	**0.613**	**0.517**	0.545	0.355	0.507	0.159	0.582	0.448	0.509	0.162
2	Glass	22.81	0.892	0.876	**0.976**	**0.976**	0.941	0.941	0.725	0.703	0.905	0.901
3	Ecoli	13.84	0.853	0.842	**0.951**	**0.951**	0.843	0.831	0.897	0.898	0.815	0.799
4	Yeast	13.87	0.702	0.701	0.68	0.663	0.616	0.515	0.688	0.687	**0.752**	**0.726**
5	PageBlock	8.79	**0.941**	**0.941**	0.94	0.94	0.94	0.94	0.9	0.899	0.939	0.939
6	Segment	6.02	**0.989**	**0.989**	0.988	0.988	0.98	0.98	0.987	0.987	0.983	0.984
7	Vehicle	2.99	**0.773**	**0.771**	0.709	0.709	0.688	0.682	0.734	0.734	0.7	0.696
8	Iris	2	**0.994**	**0.994**	0.983	0.983	0.99	0.99	0.99	0.99	0.99	0.99
9	Wisconsin	1.9	**0.96**	**0.96**	**0.96**	**0.96**	**0.96**	**0.96**	0.945	0.946	0.951	0.951
10	Pima	1.87	**0.747**	**0.744**	0.724	0.725	0.705	0.704	0.731	0.731	0.694	0.695
	Average		**0.8464**	**0.8335**	0.8456	0.825	0.817	0.7702	0.8179	0.8023	0.8238	0.7843

a particular distribution. Non-parametric tests are the best option for the ordinal data and for those techniques where one model is not fixed (Derrac et al. 2011).

As most of the data set used in our study are in ordinal form and as suggested in the literature (Demsar, 2006; Garcia, Fernandez, Luengo & Herrera, 2010; Garcia & Herrera, 2008), we used non-parametric tests to do the statistical validation. More information regarding this can be found on the website http://sci2s.ugr.es/sicidm/. Non-parametric tests are used to perform two types of analysis: multiple and pairwise comparisons. We used Friedman aligned rank test (Hodges and Lehmann 1962) for multiple comparisons and Wilcoxon Matched pair signed rank test (Wilcoxon 1945) for comparing two algorithms. We performed these tests on AUC reading (Table 5) using KEEL software tool. From Table 5, we observed that the proposed technique, FF_SMOTE, has overall performed well with most of the data sets including the highest imbalanced data set (Abalone). So we will validate this using the statistical analysis.

First step toward statistical analysis is to compare all the algorithms for any significant differences and if any kind of significant difference found then any post-hoc test can be applied to analyze the differences. We apply Friedman Aligned rank test (Hu et al. 2009) to compare all the algorithms. It is a non-parametric analog of the parametric two-way analysis of variance. Null hypothesis of Friedman states that '[t]here is no significant difference between the algorithms'. This test computes ranks for every algorithm as per the following equation:

$$F_{AR} = \frac{(k-1)\left[\sum_{j=1}^{k} \hat{R}_j^2 - \left(\frac{kn^2}{4}\right)(kn+1)^2\right]}{\{[kn(kn+1)(2kn+1)]/6\} - (1/k)\ \sum_{i=1}^{n} \hat{R}_i^2}$$

where $\hat{R}_i$ is equal to the rank total of the i$^{\text{th}}$ data set and $\hat{R}_j$ is the rank total of the j$^{\text{th}}$ algorithm. As per the equation, the best performing algorithm will have the lowest rank. To check the statistical differences among the methods, we have to compare the test statistic F_{AR} with the Chi-Square (χ^2) distribution with k-1 degree of freedom. Here, 'k' is the number of algorithms. If the test statistic of Friedman aligned rank test is more than the χ^2 table value and the *p*-value is less than 0.05, then the null hypothesis for no significant differences among the algorithms is rejected. The average aligned ranks computed by the Friedman aligned rank test are shown in Figure 8. By looking at the figure, it can be found easily that FF_SMOTE is the best performer followed by the SMOTE technique. Worst performance is shown by BL_SMOTE method. The test statistics of Friedman aligned rank test are given in the Table 6. 'N' is the number of data sets. Degree of freedom is calculated as k–1. So the value of k is 4 (total number of algorithms – 1; 5–1 = 4). The table value of χ^2 for 4 degree of freedom and α = 0.05 is 9.48773. From the Table 6, χ^2 value is 11.3 and the p-value is 0.0233, so the null hypothesis is rejected. It concludes that all the

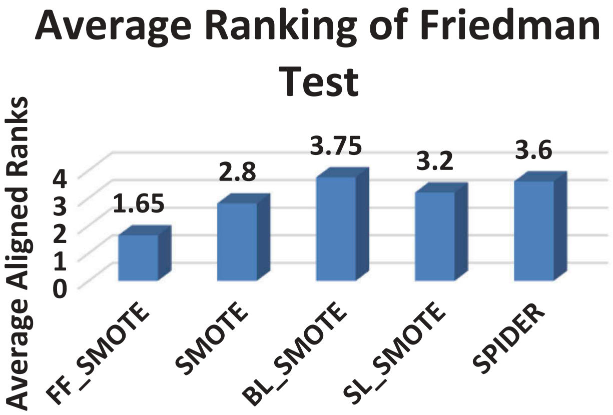

Figure 8. Average aligned rank comparison of all the algorithms.

Table 6. Test statistics of the Friedman aligned rank test.

N	10
Chi-Square	11.3
Degree of freedom	4
p-Value	0.0233

methods are not equal, there are certain differences among them. Then we used Holm post-hoc test to analyze the differences among the algorithms. The statistics of Holm test (Holm 1979) with FF_SMOTE as a control method is given in Table 7.

As per Holm statistic, FF_SMOTE outperformed all except SMOTE, despite getting the lowest *p*-value. To get the deep vision about the differences between FF_SMOTE and SMOTE, we did pairwise analysis using Wilcoxon Signed rank Test. It compares the ranks for the positive and negative differences of the two algorithms and is defined as follows:

$$R^{+} = \sum_{d_i > 0} rank(d_i) + \frac{1}{2}\sum_{d_j = 0} rank(d_i)$$

$$R^{-} = \sum_{d_i < 0} rank(d_i) + \frac{1}{2}\sum_{d_j = 0} rank(d_i)$$

Table 7. Holm test statistics for comparison among the algorithms.

Control method: FF_SMOTE (1.65)					
i	Algorithm (rank)	*Z*	*p*-Value	Holm	Hypothesis (α = 0.05)
4	BL_SMOTE (3.75)	2.969848	0.002979	0.0125	**Rejected for FF_SMOTE**
3	SPIDER (3.6)	2.757716	0.005821	0.016667	**Rejected for FF_SMOTE**
2	SL_SMOTE (3.2)	2.192031	0.028377	0.025	**Rejected for FF_SMOTE**
1	SMOTE (2.8)	1.626346	0.103876	0.05	Not rejected

Table 8. Wilcoxon test statistics for comparing FF_SMOTE and SMOTE.

Algorithms	R^+	R^-	Hypothesis (α = 0.05)	*p*-Value
FF_SMOTE vs. SMOTE	38.0	17.0	Accepted, no significant differences	0.23743

R^+ is the sum of ranks for the data set in which the first algorithm outperformed the second and R^- is the sum of ranks for the opposite. We analyzed FF_SMOTE with the SMOTE using Wilcoxon test. Test statistics are given in Table 8.

As per the statistic, although there are no significant differences between these algorithms because the *p*-value is more than 0.05 the higher ranks in favor of FF_SMOTE demonstrate the superiority of FF_SMOTE over SMOTE.

Conclusion

In this paper, we proposed an oversampling technique based upon firefly to tackle the class imbalance in classification. The proposed technique is able to generate optimized values in the smaller class region. It is applied on 10 real-time imbalanced data sets and compared with other state-of-the-art oversampling methods. The performance of the proposed method is best in case of highly imbalanced data set. Statistical validations using Friedman aligned rank test and Wilcoxon matched pair test also proved the superiority of proposed method over other techniques.

References

Alcalá-Fdez, J., A. Fernández, J. Luengo, J. Derrac, S. García, L. Sánchez, and F. Herrera. 2011. KEEL data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework. *Journal of Multiple- Valued Logic and Soft Computing* 172 (3):255–87.

Alcalá-Fdez, J., L. Sánchez, S. García, M. J. Del Jesus, S. Ventura, J. M. Garrell, J. Otero, C. Romero, J. Bacardit, V. M. Rivas, et al. 2008. KEEL: A software tool to assess evolutionary algorithms for data mining problems. *Soft Computing* 133:307–18.

Barandela, R., J. S. Sánchez, V. García, and E. Rangel. 2003. Strategies for learning in class imbalance problems. *Pattern Recognition* 363:849–51. doi:10.1016/S0031-3203(02)00257-1.

Batista, G. E. A. P. A., R. C. Prati, and M. C. Monard. 2004. A study of the behavior of several methods for balancing machine learning training data. *SIGKDD Explorations Newsletter* 6:20–29. doi:10.1145/1007730.1007735.

Batuwita, R., and V. Palade. 2013. *Class imbalance learning methods for support vector machines, imbalanced learning: Foundations, algorithms and applications*. John Wiley & Sons, New Jercy, United States.

Bezdek, J. C. 1981. *Pattern recognition with fuzzy objective function algorithm*. New York, NY: Plenum.

Bunkhumpornpat, C., K. Sinapiromsaran, and C. Lursinsap. 2009. *Safe-Level-SMOTE: Safe level- synthetic smaller over-sampling technique for handling the class imbalance problem. PADD2009, LNAI*, 5476: 475–82. Springer, Berlin, Heidelberg.

Chawla, N. V., K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. 2002. SMOTE: Synthetic smaller over sampling technique. *Journal of Artificial Intelligence Research* 16:321–57. doi:10.1613/jair.953.

Chawla, N. V., A. Lazarevic, L. O. Hall, and K. W. Bowyer 2003. SMOTBoost: Improving prediction of the smaller class in boosting. Proc. Knowledge Discovery databases, 107–19, Berlin, Heidelberg.

Chi, Z., H. Yan, and T. Pam. 1996. Fuzzy algorithms: With application to image processing and pattern recognition. In Tuan Pham (Ed.), *Advances in fuzzy systems-applications and theory: Volume 10*. Singapore: World Scientific.

Cristianini, N., Shawe-Taylor, J., Elisseeff, A., & Kandola, J. S. 2002. On kernel-target alignment. *Advances in Neural Information Processing System* 14:367–73.

Demšar, J. 2006. Statistical comparisons of classifiers over multiple datasets. *Journal of Machine Learning Research* 7:1–30.

Derrac, J., S. Garcia, D. Molina, and F. Herrera. 2011. A practical tutorial on the use of non-parametric statistical tests as a methodology for comparing evolutionary and Swarm intelligence algorithms. *Swarm and Evolutionary Computation* 1:3–18. doi:10.1016/j.swevo.2011.02.002.

Fernandez, A., S. García, M. J. Del Jesus, and F. Herrera. 2008. A study of the behaviour of linguistic fuzzy rule base classification systems in the framework of imbalanced data-sets. *Fuzzy Sets and Systems* 15918:2378–98. doi:10.1016/j.fss.2007.12.023.

Fister, I., I. Fister, X.-S. Yang, and J. Brest. 2013. A comprehensive review of firefly algorithm. *Swarm and Evolutionary Computation* 13:34–46. doi:10.1016/j.swevo.2013.06.001.

Galar, M., Fernandez, A., Barrenechea, E., Bustince, H., and Herrera, F. 2011. A review on ensembles for the class imbalance problem: Bagging, boosting and hybrid based approaches. *IEEE Transactions on System, Man and Cybernetics-Part C: Applications and Reviews* 42(4): 463–84.

Galar, M., A. Fernández, E. Barrenechea, and F. Herrera. 2013. EUSBoost: Enhancing ensembles for highly imbalanced data-sets by evolutionary undersampling. *Pattern Recognition* 46:3460–71. doi:10.1016/j.patcog.2013.05.006.

García, S., A. Fernández, J. Luengo, and F. Herrera. 2010. Advanced non parametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power. *Information Sciences* 180:2044–64. doi:10.1016/j.ins.2009.12.010.

Garcia, S., and F. Herrera. 2009. Evolutionary undersampling for classification with imbalanced data-sets: Proposals and taxonomy. *Evolutionary Computation* 17:275–306. doi:10.1162/evco.2009.17.3.275.

García, S., and F. Herrera. 2008. An extension on statistical comparisons of classifiers over multiple datasets for all pairwise comparisons. *Journal of Machine Learning Research* 9:2677–94.

Green, D. M., and J. A. Swets. 1966. *Signal detection theory and psychophysics*. New York: Wiley.

Guo, H., and H. L. Viktor. 2004. Learning from imbalanced data-sets with boosting and data generation: The Databoost-IM approach. *SIGKDD Explorations Newsletter* 6:30–39. doi:10.1145/1007730.1007736.

Han, H., W. Wang, and B. Mao 2005. Borderline-SMOTE: A new oversampling method in imbalanced data-sets learning. ICIC 2005. *LNCS* 3644, 878–87, Springer, Berlin, Heidelberg.

Hanley, J. A., and B. J. McNeil. 1983. A method of comparing the areas under receiver operating characteristic curves derived from the same cases. *Radiology* 1483:839–43. doi:10.1148/radiology.148.3.6878708.

Hodges, J. L., and E. L. Lehmann. 1962. Rank methods for combination of independent experiments in analysis of variance. *Annals of Mathematical Statistics* 33:482–97. doi:10.1214/aoms/1177704575.

Holm, S. 1979. A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics* 6:65–70.

Hong, X., S. Chen, and C. J. Harris. 2007. A kernel based two class classifier for imbalanced data-sets. *IEEE Transactions on Neural Networks* 181:28–41. doi:10.1109/TNN.2006.882812.

Hu, S., Liang, Y., Ma, L., and He, Y. 2009. MSMOTE: Improving classification performance when training data is imbalanced. *Proceedings of the Second International Workshop on Computer Science and Engineering* 2:13–17.

Iman, T., K. Ting, and J. Kamruzzaman 2006. z-SVM: An SVM for improved classification of imbalanced data. Proceedings of the 19th Australian joint conference on Artificial Intelligence, 264–73, Springer-verlag, Berlin, Heidelberg.

Kandola, J., and J. Shawe-Taylor 2003. Refining kernel for regression and uneven classification problems. Proceedings of International Conference on Artificial Intelligence and Statistics, InAISTATS, 2003.

Kim, M. J. 2013. Geometric mean based boosting algorithm to resolve data imbalance problem. DBKDA2013, The fifth International conf. on Advances in databases, knowledge and data applications, In PACIS, 15–20.

Lin, C.-F., and S.-D. Wang. 2002. Fuzzy support vector machines. *IEEE Transactions on Neural Networks* 13 (2):464–71. doi:10.1109/72.991432.

Mark, H., F. Eibe, H. Geoffrey, P. Bernhard, R. Peter, and H. Ian. 2009. The WEKA data mining software: An update. *SIGKDD Explorations* 11:1.

Massachusetts, N. 2013. *MATLAB version 8.1 2013*. The MathWorks Inc.

Metz, C. E. 1978. Basic principals of ROC analysis. *Seminars in Nuclear Medicine* 84:283–98. doi:10.1016/S0001-2998(78)80014-2.

Quinlan, J. R. 1993. *C4.5: Programs for machine learning*. 1st ed. San Mateo-California: Morgan Kaufmann Publishers.

Spacman, K. A. 1989. Signal detection theory: Valuable tools for evaluating inductive learning. Proc. Sixth International Workshop on Machine learning, Morgan Kaufman, San Mateo, CA, 160–63.

Stefanowski, J., and S. Wilk. 2008. Selective preprocessing of imbalanced data for improving classification performance. *Datawarehousing and Knowledge Discovery Lecture Notes in Computer Science* 5182:283–92.

Wang, B. X., and N. Japkowicz. 2010. Boosting support vector machines for imbalanced data-sets. *Knowledge Information Systems* 25 (1):1–20. doi:10.1007/s10115-009-0198-y.

Wilcoxon, F. 1945. Individual comparisons by ranking methods. *Biometrics Bulletin* 16:80–83. doi:10.2307/3001968.

Wilson, D. R., and T. Martinez. 2000. Reduction techniques for data point-based learning algorithms. *Machine Learning Journal* 38:257–86. doi:10.1023/A:1007626913721.

Wu, G., and E. Chang 2003a. Adaptive feature space conformal transformation for imbalanced data learning. Proceedings of the 20th International Conference on Machine Learning, Washington DC, 816–23.

Wu, G., and E. Chang 2003b. Class boundary alignment for imbalanced data-set learning. Proceedings of the International Conference on Machine Learning from Imbalanced data-sets, Washington DC, 49–56.

Wu, G., and E. Chang. 2005. Kba: Kernel boundary alignment considering imbalanced dataset distribution. *IEEE Transactions on Knowledge and Data Engineering* 176:786–95. doi:10.1109/TKDE.2005.95.

Wu, S., and S. I. Amari. 2002. Conformal transformation of kernel functions: A data-dependent way to improve the performance of support vector machine classifier. *Neural Networks Letter* 15(1):59-67.

Yang, X. S. 2008. *Nature-inspired metaheuristic algorithms*. Frome: Luniver Press. ISBN:1-905986-10-6.

Yang, X. S. 2009. Firefly algorithms for multimodal optimization. In *Proceedings of 5th symposium on stochastic algorithms, foundations and applications. Lecture notes in computer science*, ed. O. Watanabe and T. Zeugmann, Berlin, Heidelberg, vol. 5792, 169–78.

Yang, X. S. 2010. *Nature-inspired metaheuristic algorithms*. 2nd ed. Luniver Press, United Kingdom. ISBN: 1-905986-28-9.

Yang, X.-S., and X. He. 2013. Firefly algorithm: Recent advances and applications. *International Journal of Swarm Intelligence* 1:36–50. doi:10.1504/IJSI.2013.055801.

Yen, S.-J., and Y.-S. Lee. 2009. Cluster-based under-sampling approaches for imbalanced data distributions. *Expert Systems with Applications* 36:5718–27. doi:10.1016/j.eswa.2008.06.108.

Ying, M. 2013. Imbalanced classification based on active learning SMOTE. *Research Journal of Applied Sciences, Engineering and Technology* 53:944–49.