A Novel Class Imbalance Learning Using Intelligent Under-Sampling

Naganjaneyulu Satuluri[1] and Mrithyumjaya Rao Kuppa[2]

[1]*Associate Professor, Lakireddy BaliReddy College of Engg, Mylavaram, India*
[2]*Professor, Vaagdevi College of Engineering, Warangal, India*
svna2198@gmail.com

Abstract

Class imbalance is a problem that is very much critical in many real-world application domains of machine learning. When examples of one class in a training data set vastly outnumber examples of the other class(es), traditional data mining algorithms tend to create suboptimal classification models. Researchers have rigorously studied several techniques to alleviate the problem of class imbalance, including resampling algorithms, and feature selection approaches to this problem. In this paper, we present a new hybrid feature selection algorithm dubbed as Class Imbalance Learning using Intelligent Under Sampling (CILIUS), for learning from skewed training data. This algorithm provides a simpler and faster alternative by using C4.5 as base algorithm. We conduct experiments using four UCI data sets from various application domains using five learning algorithms for comparison and five evaluation metrics. Experimental results show that our method has higher Area under the ROC Curve, F-measure, Precision, TP rate and low TN rate values than many existing class imbalance learning methods.

Keywords: *Classification, class imbalance, filter, intelligent sampling, CILIUS*

1. Introduction

A dataset is class imbalanced if the classification categories are not approximately equally represented. The level of imbalance (ratio of size of the majority class to minority class) can be as huge as 1:99 [1]. It is noteworthy that class imbalance is emerging as an important issue in designing classifiers [2, 3, 4]. Furthermore, the class with the lowest number of instances is usually the class of interest from the point of view of the learning task [5]. This problem is of great interest because it turns up in many real-world classification problems, such as remote-sensing [6], pollution detection [7], risk management [8], fraud detection [9], and especially medical diagnosis [10–13].

There exist techniques to develop better performing classifiers with imbalanced datasets**,** which are generally called Class Imbalance Learning (CIL) methods. These methods can be broadly divided into two categories, namely, external methods and internal methods. External methods involve preprocessing of training datasets in order to make them balanced, while internal methods deal with modifications of the learning algorithms in order to reduce their sensitiveness to class imbalance [14]. The main advantage of external methods as previously pointed out, is that they are independent of the underlying classifier. In this paper, we are laying more stress to propose an external CIL method for solving the class imbalance problem.

This paper is organized as follows. Section II briefly reviews the Data Balancing problems and its measures. and in Section III, the proposed method of using the IUS (Intelligent Under Sampling) technique for CIL is described. Section IV presents the

imbalanced datasets used to validate the proposed method, while In Section V, the experimental setting are presented and In Section VI discuss, in detail, the classification results obtained by the proposed method and compare them with the results obtained by different existing methods and finally, in Section VII the paper is concluded.

2. Data Balancing

Whenever a class in a classification task is underrepresented (i.e., has a lower prior probability) compared to other classes, we consider the data as imbalanced [15, 16]. The main problem in imbalanced data is that the majority classes that are represented by large numbers of patterns rule the classifier decision boundaries at the expense of the minority classes that are represented by small numbers of patterns. This leads to high and low accuracies in classifying the majority and minority classes, respectively, which do not necessarily reflect the true difficulty in classifying these classes. Most common solutions to this problem balance the number of patterns in the minority or majority classes.

Either way, balancing the data has been found to alleviate the problem of imbalanced data and enhance accuracy [15, 16, 17]. Data balancing is performed by, e.g., oversampling patterns of minority classes either randomly or from areas close to the decision boundaries. Interestingly, random oversampling is found comparable to more sophisticated oversampling methods [17]. Alternatively, undersampling is performed on majority classes either randomly or from areas far away from the decision boundaries. We note that random undersampling may remove significant patterns and random oversampling may lead to overfitting, so random sampling should be performed with care. We also note that, usually, oversampling of minority classes is more accurate than undersampling of majority classes [17].

Resampling techniques can be categorized into three groups. Undersampling methods, which create a subset of the original data-set by eliminating instances (usually majority class instances); oversampling methods, which create a superset of the original data-set by replicating some instances or creating new instances from existing ones; and finally, hybrids methods that combine both sampling methods. Among these categories, there exist several different proposals; from this point, we only center our attention in those that have been used in under sampling.

- *Random undersampling*: It is a nonheuristic method that aims to balance class distribution through the random elimination of majority class examples. Its major drawback is that it can discard potentially useful data, which could be important for the induction process.

- *Random oversampling*: In the same way as random oversampling, it tries to balance class distribution, but in this case, randomly replicating minority class instances. Several authors agree that this method can increase the likelihood of occurring overfitting, since it makes exact copies of existing instances.

- *Hybrid Methods:*In this hybrid method both undersampling and oversampling will be applied for the datasets so as to make it a balance dataset.

The bottom line is that when studying problems with imbalanced data, using the classifiers produced by standard machine learning algorithms without adjusting the output threshold may well be a critical mistake. This skewness towards minority class (positive) generally

causes the generation of a high number of false-negative predictions, which lower the model's performance on the positive class compared with the performance on the negative (majority) class.A comprehensive review of different CIL methods can be found in [18]. The following section briefly discuss the external-imbalance and internal-imbalance learning methods.

The external methods are independent from the learning algorithm being used, and they involve preprocessing of the training datasets to balance them before training the classifiers. Different resampling methods, such as random and focused oversampling and undersampling, fall into to this category. In random undersampling, the majority-class examples are removed randomly, until a particular class ratio is met [19]. In random oversampling, the minority-class examples are randomly duplicated, until a particular class ratio is met [18]. Synthetic minority oversampling technique (SMOTE) [20] is an oversampling method, where new synthetic examples are generated in the neighborhood of the existing minority-class examples rather than directly duplicating them. In addition, several informed sampling methods have been introduced in [21]. A clustering-based sampling method has been proposed in [22], while a genetic algorithm based sampling method has been proposed in [23].

3. Class Imbalance Learning using Intelligent Under-Sampling

In this section, we follow a design decomposition approach to systematically analyze the different unbalanced domains. We first briefly introduce the framework design for our proposed algorithm.

The working style of under-sampling tries to decrease the number of weak or noise examples. Here, the weak instances related to the specific features are to be eliminated, which is identified according to a well-established filter and intelligent technique. The number of instances eliminated will belong to the '*k'*feature selected by filter and intelligent technique. Here, the above said routine is employed, which removes examples suffering from feature to class label noises at first and then removes borderline examples and examples of outlier category.

Feature to Class label noises are the examples whose influence is not seen for the decision of the class for that particular feature. Here, they are identified by the limited range categories, using the above said technique. In detail, at first some examples are deleted temporary from **N*strong*,** a dataset created with strong instances. Then, for a class to be shrank, all its examples inside of **N*strong*** are classified. If the classification is correct, and the accuracy is increased then the examples deleted temporary are regarded as being feature class label noises.

Borderline examples are the examples close to the boundaries between different classes for a specific feature. They are unreliable because even a small amount of attribute noise can send the example to the wrong side of the boundary. The outliers are those examples which are very rare in nature from the remaining set of examples. These are examples are of very rare use to the classification and thus to be removed for better performance.

The presented under-sampling algorithm is summarized In Algorithm 1.

Algorithm 1 CILIUS

1: { Input: A set of minor class examples *P*, a set of major class examples *N*, *jPj<jNj*, and *Fj*, the feature set, j> 0. *}*
2: *k* ← 0.
3: Apply CFS on subset *N*,
4: Find *Fj* for N, k= number of features extracted in CFS
5: repeat
6: k=k+1
7: Select the range for weak or noises instances of *Fj*.
8: Remove ranges of weak attributes and form a set of major class examples N*strong*
9: Until j = k
10: Train and Learn A Base Classifier (C4.5) Using *P* and *Nstrong*.
11: Obtain the values of AUC,TP,FP,F-Measure
12: Output: Average Measure;

The different components of our new proposed framework are elaborated in the next subsections.

3.1. Preparation of the Subsets

The datasets is partitioned into majority and minority subsets. As we are concentrating on under sampling, we will take majority data subset for further analysis and reduction.

3.2. Influential Feature Subset Detection

Majority subset can be further analyzed to find the weak or noisy instances so that we can eliminate those. For finding the weak instances one of the ways is that find most influencing attributes or features and then remove ranges of the noisy or weak attributes relating to that feature. How to find the most influencing attribute is by using an attribute selecting filter, in this case we have used Correlation based Feature Subset (CFS) evaluation [24]. The percentage of the weak instances removed may depend upon the properties of the dataset. The number of most influencing features in the dataset may also depend upon the unique properties of the dataset.

3.3. Choosing Feature Class Label Noise Ranges

How to choose the weak instances relating to that feature from the dataset set. We can find a range where the number of samples are less can give you a simple hint that those instances coming in that range or very rare or noise. We will intelligently detect and remove those instances which are in narrow ranges of that particular feature, borderline and noise instances.

3.4. Forming the Balance Dataset

The minority subset and the stronger majority subset is combined to form a strong and balance dataset, which is used for learning of a base algorithm. In this case we have used C4.5 as the base algorithm.

4. Dataset Details

We considered four benchmark real-world imbalanced dataset from the UCI machine learning repository [25] to validate the proposed new algorithm.

4.1. Datasets

Table 1 summarizes the details of these datasets in the ascending order of the positive-to-negative dataset ratio. This contains the name of the dataset, the total number of examples (Total), attribute, the number of target classes for each dataset, number of minority class examples (#min.), the number of majority class examples (#maj.). These datasets represent a whole variety of domains, complexities, and imbalance ratios.

We evaluate the proposed CILIUS algorithm on four real-world datasets including Breast_w, Diabetes, Hepatitis and Credit-g datasets. The four datasets are obtained from the University of California at Irvine machine learning repository [25]. The detailed information about the datasets is described in Table 1.

Table 1. The 4 UCI Datasets and their Properties

Dataset	Total	Attribute	Class	#min/#maj
Breast-w	699	9C	2	241/458
Diabeties	768	8C	2	268/500
Hepatatisis	155	13B, 6C	2	32/123
Credit-g	1000	14C ,7B	2	300/700

For every data set, we perform a tenfold stratified cross validation. Within each fold, the classification method is repeated ten times considering that the sampling of subsets introduces randomness. The AUC, Precision, F-measure, TP rate and TN rate of this cross-validation process are averaged from these ten runs. The whole cross-validation process is repeated for ten times, and the final values from this method are the averages of these ten cross-validation runs.

4.2. Performance Evaluation Criteria's

To assess the classification results we count the number of true positive (TP), true negative (TN), false positive (FP) (actually negative, but classified as positive) and false negative (FN) (actually positive, but classified as negative) examples. It is now well known that error rate is not an appropriate evaluation criterion when there is class imbalance or unequal costs. In this paper, we use AUC, Precision, F-measure, TP Rate and TN Rate as performance evaluation measures.

Let us define a few well known and widely used measures:

The Area under Curve (AUC) measure is computed by using the formula given in equation (i),

$$AUC = \frac{1 + TP_{RATE} - FP_{RATE}}{2} \quad \text{----------} \quad \text{(i)}$$

The Precision measure is computed by using the formula given in equation (ii),

$$\Pr ecision = \frac{TP}{(TP)+(FP)} \quad \text{----------} \quad \text{(ii)}$$

The F-measure Value is computed by using the formula given in equation (iii),

$$F-measure = \frac{2 \times \Pr ecision \times \mathrm{Re} call}{\Pr ecision + \mathrm{Re} call} \quad \text{----------} \quad \text{(iii)}$$

The True Positive Rate measure is computed by using the formula given in equation (iv),

$$TruePositiveRate = \frac{TP}{(TP)+(FN)} \quad \text{----------} \quad \text{(iv)}$$

The True Negative Rate measure is computed by using the formula given in equation (v),

$$TrueNegativeRate = \frac{TN}{(TN)+(FP)} \quad \text{----------} \quad \text{(v)}$$

5. Experimental Settings

5.1. Algorithms and Parameters

In first place, we need to define a baseline classifier which we use in our proposed algorithm implementation. With this goal, we have used C4.5 decision tree generating algorithm [24]. Furthermore, it has been widely used to deal with imbalanced data-sets [27]–[29], and C4.5 has also been included as one of the top-ten data-mining algorithms [30].

Because of these facts, we have chosen it as the most appropriate base learner. C4.5 learning algorithm constructs the decision tree top-down by the usage of the normalized information gain (difference in entropy) that results from choosing an attribute for splitting the data. The attribute with the highest normalized information gain is the one used to make the decision.

To validate the proposed CILIUS algorithm, we compared it with the traditional C4.5, CART (Classification and Regression trees), BPN (Back Propagation Neural Networks), REP (Reduced Error Pruning Tree) and SMOTE (Synthetic Minority Oversampling TEchnique). Four real world benchmark datasets taken from the UCI Machine Learning Repository are used throughout the experiments (see Table 1). We performed the implementation using Weka on Windows XP with 2Duo CPU running on 3.16 GHz PC with 3.25 GB RAM.

6. Results

We evaluated the performance of the proposed CILIUS approach on a number of real-world classification problems. The goal is to examine whether the new proposed learning framework achieve better classification performance than a number of existing learning algorithms.

We compared proposed method CILIUS with the C4.5, CART, BPN, REP and SMOTE state-of -the-art learning algorithms. In all the experiments we estimate AUC, Precision, F-measure, TP rate and TN rate using 10-fold cross-validation. We experimented with 4 standard datasets for UCI repository (Breast_w, Diabetes, Hepatitis and Credit-g); these datasets are standard benchmarks used in the context of high-dimensional imbalance learning. Experiments on these datasets have 2 goals. First, we study the class imbalance properties of

the datasets using proposed CILIUS learning algorithms. Second, we compare the classification performance of our proposed CILIUS algorithm with the traditional and class imbalance learning methods based on all datasets.

Table 2. Tenfold Cross Validation Classification Performance for Breast_w Dataset

	C4.5	CART	BPN	REP	SMOTE	CILIUS
AUC	0.957±0.034●	0.950±0.031●	0.991 ±0.018○	0.964±0.038●	0.972±0.027●	0.987±0.016
Precision	0.965±0.026●	0.971±0.033●	0.976±0.032●	0.962±0.034●	0.976±0.034●	0.986±0.020
F-measure	0.962±0.021●	0.960±0.020●	0.973±0.021●	0.963±0.027●	0.961±0.025●	0.984±0.014
TP Rate	0.959±0.033●	0.954±0.032●	0.972±0.035●	0.961±0.036●	0.953±0.037●	0.984±0.022
TN Rate	0.932±0.052○	0.941±0.056○	0.944±0.062○	0.931±0.068○	0.985±0.028 ●	0.978±0.030

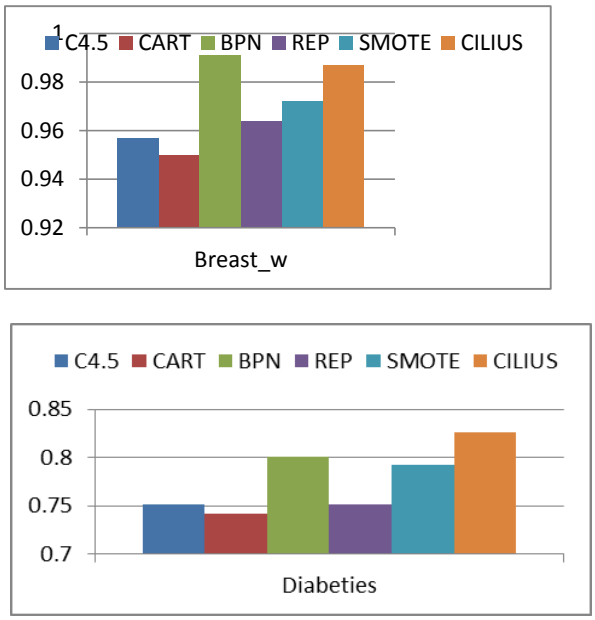

Figure 1. Test Results on AUC between the C4.5, CART, BPN, REP, SMOTE and CILIUS for Breast_w and Diabeties Datasets

Table 3. Tenfold Cross Validation Classification Performance for Pima Indian Diabetes Dataset

	C4.5	CART	BPN	REP	SMOTE	CILIUS
AUC	0.751±0.070●	0.742±0.078●	0.801 ±0.058●	0.751±0.068●	0.792±0.046●	0.826±0.056
Precision	0.797±0.045●	0.784±0.041●	0.791±0.053●	0.793±0.044●	0.781±0.062●	0.810±0.048
F-measure	0.806±0.044●	0.818±0.045●	0.812±0.420●	0.817±0.045●	0.743±0.058●	0.836±0.040
TP Rate	0.821±0.073●	0.852±0.075●	0.842±0.061●	0.841±0.076●	0.712±0.089●	0.869±0.064
TN Rate	0.603±0.111○	0.551±0.106○	0.581±0.015○	0.572±0.103○	0.814±0.087●	0.696±0.096

Table 4. Tenfold Cross Validation Classification Performance for Hepatitis Dataset

	C4.5	CART	BPN	REP	SMOTE	CILIUS
AUC	0.668±0.184●	0.561±0.130●	0.812 ±0.157○	0.624±0.158●	0.806±0.112○	0.714±0.166
Precision	0.510±0.371●	0.233±0.337●	0.561±0.308●	0.292±0.391●	0.712±0.175○	0.698±0.305
F-measure	0.409±0.272●	0.189±0.231●	0.512±0.257●	0.213±0.267●	0.682±0.149○	0.556±0.238
TP Rate	0.374±0.256●	0.172±0.246●	0.523±0.295○	0.192±0.249●	0.681±0.195○	0.499±0.525
TN Rate	0.900±0.097○	0.931±0.097●	0.891±0.094○	0.947±0.099●	0.848±0.112○	0.920±0.092

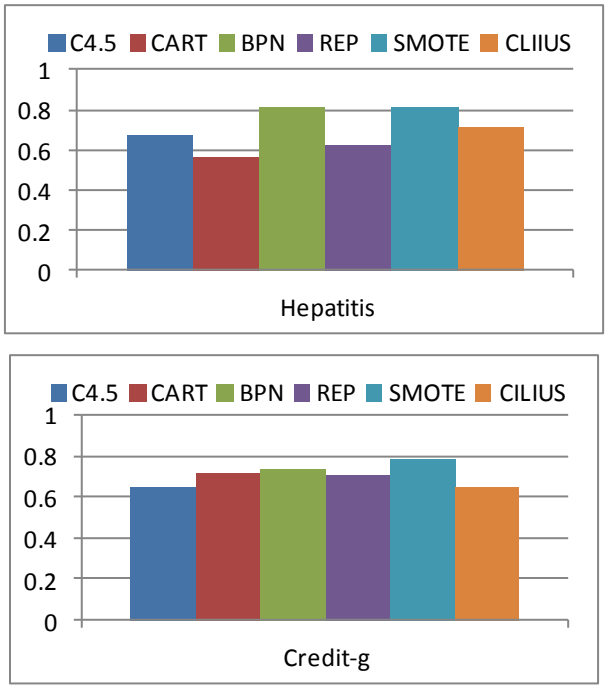

Figure 2. Test Results on AUC between the C4.5, CART, BPN, REP, SMOTE and CILIUSfor Hepatitis and Credit-g Datasets

Table 5. Tenfold Cross Validation Classification Performance for Credit-g Dataset

AUC	0.647±0.062○	0.716±0.055○	0.730 ±0.054○	0.705±0.057○	0.778±0.041○	0.648±0.070
Precision	0.767±0.025○	0.779±0.030○	0.792±0.031○	0.765±0.025○	0.768±0.034○	0.648±0.054
F-measure	0.805±0.022○	0.820±0.028○	0.801±0.033○	0.814±0.026○	0.787±0.034○	0.674±0.057
TP Rate	0.847±0.036○	0.869±0.047○	0.809±0.049○	0.872±0.057○	0.810±0.058○	0.707±0.084
TN Rate	0.398±0.085○	0.421±0.102○	0.503±0.085○	0.371±0.105○	0.713±0.056●	0.546±0.099

Results and Analysis

In Table 2,3,4 and 5, we present the results (with '●'indicating win of CILIUS Vs other algorithms and '○' indicating tie or loss of CILIUS Vs other algorithms) on breast_w, diabetes, hepatitis and credit-g datasets with comparison between C4.5, CART, BPN, REP and SMOTE versus CILIUS. Figure1 shows the mean AUC value of breast_w and diabetes dataset corresponding to each learning technique. Figure 2 shows the mean AUC value of hepatitis and credit-g dataset corresponding to each learning technique. Corresponding charts for other measures are omitted due to the space limitation. From these results we can make several observations. First, the developed new method generally outperform on all the methods C4.5, CART, BPN, REP and SMOTE on AUC, Precision, F-measure. TP rate and TN rate measures; the advantage of our method is most visible in the Breast_w, Diabeties and Hepatitis datasets Secondly, with the exception of the Credit-g datasets, there is at least one or other algorithm that outperforms the standard C4.5 algorithm. Finally, the method that most often win on almost all the datasets is our new Proposed Method.

7. Conclusion

In this paper we present the class imbalance problem paradigm, which exploits the weighted human learning strategy using filter and intelligent technique in the supervised learning research area, and implement it with C4.5 as its base learner. Experimental results show that CILIUS will perform well in the case of multi class imbalance datasets. Furthermore, CILIUS is much less volatile than C4.5. In our future work, we will apply CILIUS to more learning tasks, especially high dimensional feature learning tasks.

Acknowledgements

We thank Dr.L.S.S Reddy,Director Lakireddy Bali Reddy College of Engineering, Mylavaram, for his guidance and consistent support to complete this research work. We thank Professor Rao Vemuri for his comments for the improvement of the paper. We also thank anonymous reviewers for the efforts they will lay for the betterment of the paper.

References

[1] J. Wu, S. C. Brubaker, M. D. Mullin and J. M. Rehg, "Fast asymmetric learning for cascade face detection", IEEE Trans. Pattern Anal. Mach. Intell., vol. 30, no. 3, **(2008)** March, pp. 369–382.

[2] N. V. Chawla, N. Japkowicz and A. Kotcz, Eds., Proc. ICML Workshop Learn. Imbalanced Data Sets, **(2003)**.

[3] N. Japkowicz, Ed., Proc. AAAI Workshop Learn. Imbalanced Data Sets, **(2000)**.

[4] G. M. Weiss, "Mining with rarity: A unifying framework", ACM SIGKDD Explor. Newslett., vol. 6, no. 1, , **(2004)** June, pp. 7–19.

[5] N. V. Chawla, N. Japkowicz and A. Kolcz, Eds., Special Issue Learning Imbalanced Datasets, SIGKDD Explor. Newsl., vol. 6, no. 1, **(2004)**.

[6] W. -Z. Lu and D. Wang, "Ground-level ozone prediction by support vector machine approach with a cost-sensitive classification scheme", Sci. Total. Enviro., vol. 395, no. 2-3, **(2008)**, pp. 109–116.

[7] Y. -M. Huang, C. -M. Hung and H. C. Jiau, "Evaluation of neural networks and data mining methods on a credit assessment task for class imbalance problem", Nonlinear Anal. R. World Appl., vol. 7, no. 4, **(2006)**, pp. 720–747.

[8] D. Cieslak, N. Chawla and A. Striegel, "Combating imbalance in network intrusion datasets", in IEEE Int. Conf. Granular Comput., **(2006)**, pp. 732–737.
[9] M. A. Mazurowski, P. A. Habas, J. M. Zurada, J. Y. Lo, J. A. Baker and G. D. Tourassi, "Training neural network classifiers for medical decision making: The effects of imbalanced datasets on classification performance", Neural Netw., vol. 21, no. 2–3, **(2008)**, pp. 427–436.
[10] A. Freitas, A. Costa-Pereira and P. Brazdil, "Cost-sensitive decision trees applied to medical data", in Data Warehousing Knowl. Discov. (Lecture Notes Series in Computer Science), I. Song, J. Eder, and T. Nguyen, Eds., **(2008)**.
[11] K. Kilic, O¨ zgeUncu and I. B. Tu¨rksen, "Comparison of different strategies of utilizing fuzzy clustering in structure identification", Inf. Sci., vol. 177, no. 23, **(2007)**, pp. 5153–5162.
[12] M. E. Celebi, H. A. Kingravi, B. Uddin, H. Iyatomi, Y. A. Aslandogan, W. V. Stoecker and R. H. Moss, "A methodological approach to the classification of dermoscopy images", Comput.Med. Imag. Grap., vol. 31, no. 6, **(2007)**, pp. 362–373.
[13] X. Peng and I. King, "Robust BMPM training based on second-order cone programming and its application in medical diagnosis", Neural Netw., vol. 21, no. 2–3, pp. 450–457, 2008.Berlin/Heidelberg, Germany: Springer, vol. 4654, **(2007)**, pp. 303–312.
[14] RukshanBatuwita and Vasile Palade (2010) FSVM-CIL: Fuzzy Support Vector Machines for Class Imbalance Learning, IEEE Transactions on Fuzzy Systems, vol. 18, no. 3, **(2010)**, pp. 558-571.
[15] N. Japkowicz and S. Stephen, "The Class Imbalance Problem: A Systematic Study", Intelligent Data Analysis, vol. 6, **(2002)**, pp. 429-450.
[16] M. Kubat and S. Matwin, "Addressing the Curse of Imbalanced Training Sets: One-Sided Selection", Proc. 14th Int'l Conf. Machine Learning, **(1997)**, pp. 179-186.
[17] G. E. A. P. A. Batista, R. C. Prati and M. C. Monard, "A Study of the Behavior of Several Methods for Balancing Machine Learning Training Data", SIGKDD Explorations, vol. 6, **(2004)**, pp. 20-29.
[18] D. Cieslak and N. Chawla, "Learning decision trees for unbalanced data", in Machine Learning and Knowledge Discovery in Databases. Berlin, Germany: Springer-Verlag, **(2008)**, pp. 241–256.
[19] G.Weiss, "Mining with rarity: A unifying framework", SIGKDD Explor. Newslett., vol. 6, no. 1, **(2004)**, pp. 7–19.
[20] N. Chawla, K. Bowyer and P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique", J. Artif. Intell. Res., vol. 16, **(2002)**, pp. 321–357.
[21] J. Zhang and I. Mani, "KNN approach to unbalanced data distributions: A case study involving information extraction", in Proc. Int. Conf. Mach. Learning, Workshop: Learning Imbalanced Data Sets, Washington, DC, **(2003)**, pp. 42–48.
[22] T. Jo and N. Japkowicz, "Class imbalances versus small disjuncts", ACM SIGKDD Explor. Newslett., vol. 6, no. 1, **(2004)**, pp. 40–49.
[23] S. Zou, Y. Huang, Y. Wang, J. Wang and C. Zhou, "SVM learning from imbalanced data by GA sampling for protein domain prediction", in Proc. 9th Int. Conf. Young Comput. Sci., Hunan, China, **(2008)**, pp. 982– 987.
[24] M. A. Hall, "Correlation-based Feature Subset Selection for Machine Learning", PhD Thesis, Hamilton, New Zealand, **(1998)**.
[25] A. Asuncion and D. Newman, UCI Repository of Machine Learning Database (School of Information and Computer Science), Irvine, CA: Univ. of California, http://www.ics.uci.edu/~mlearn/MLRepository.html, **(2007)**.
[26] J. R. Quinlan, "C4.5: Programs for Machine Learning", 1st ed. San Mateo, CA: Morgan Kaufmann Publishers, **(1993)**.
[27] C. -T. Su and Y. -H. Hsiao, "An evaluation of the robustness of MTS for imbalanced data", IEEE Trans. Knowl. Data Eng., vol. 19, no. 10, **(2007)** October, pp. 1321– 1332.
[28] D. Drown, T. Khoshgoftaar and N. Seliya, "Evolutionary sampling and software quality modeling of high-assurance systems", IEEE Trans. Syst., Man, Cybern. A, Syst., Humans., vol. 39, no. 5, **(2009)** September, pp. 1097–1107.
[29] S. Garc´ıa, A. Fern´andez and F. Herrera, "Enhancing the effectiveness and interpretability of decision tree and rule induction classifiers with evolutionary training set selection over imbalanced problems", Appl. Soft Comput., vol. 9, no. 4, **(2009)**, pp. 1304–1314.
[30] X. Wu, V. Kumar, J. Ross Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. J. McLachlan, A. Ng, B. Liu, P. S. Yu, Z.-H. Zhou, M. Steinbach, D. J. Hand and D. Steinberg, "Top 10 algorithms in data mining", Knowl. Inf. Syst., vol. 14, **(2007)**, pp. 1–37.

Authors

S. Naganjaneyulu received his MCA degree from Acharya Nagarjuna University, Guntur, in 1999. M.Tech degree in Computer Science & Engineering from Dr. M.G.R. University, Chennai in 2007 and pursuing his Ph.D in Data Mining from Acharya Nagarjuna University, Guntur. Currently, he is working as a Associate Professor of IT in Lakireddy Bali Reddy College of Engineering, Mylavaram, India. He has got 11 years of teaching experience.

Mrithyumjaya Rao Kuppa received a Ph.D. degree from Kakatiya University in 1979. Now, he is a professor in Faculty of Computer Science and Engineering in Vaagdevi College of engineering, Warangal (India). His current research interest includes data mining techniques with applied to real world problems. He has published in total 16 papers in reputed journals and conferences such as IEEE, Elsevier, Springer and ACM. He also served as a Conference Chair for 2nd Vaagdevi International Conference on Information Technology for real world problems (VCON'10)-2010, http://www.vcon.net.in.