Communications on Applied Electronics (CAE) – ISSN : 2394-4714
Foundation of Computer Science FCS, New York, USA
Volume 5 – No.7, July 2016 – www.caeaccess.org

A Novel Approach for Handling Imbalanced Data in Medical Diagnosis using Undersampling Technique

Varsha Babar
ME Student,
Department of Computer Engineering
Dr. D. Y. Patil School of
Engineering and Technology
Savitribai Phule Pune University

Roshani Ade
Assistant Professor,
Department of Computer Engineering
Dr. D. Y. Patil School of
Engineering and Technology
Savitribai Phule Pune University

ABSTRACT
In many data mining applications the imbalanced learning problem is becoming ubiquitous nowadays. When the data sets have an unequal distribution of samples among classes, then these data sets are known as imbalanced data sets. When such highly imbalanced data sets are given to any classifier, then classifier may misclassify the rare samples from the minority class. To deal with such type of imbalance, several undersampling as well as oversampling methods were proposed. Many undersampling techniques do not consider distribution of information among the classes, similarly some oversampling techniques lead to the overfitting or may cause overgeneralization problem. This paper proposes an MLP-based undersampling technique (MLPUS) which will preserve the distribution of information while doing undersampling. This technique uses stochastic measure evaluation for identifying important samples from the majority as well as minority samples. Experiments are performed on 5 real world data sets for the evaluation of performance of proposed work.

General Terms
Machine Learning, Classification.

Keywords
Imbalanced Learning, Undersampling, Oversampling, Clustering.

1. INTRODUCTION
In various real time applications many of the data sets are very much imbalanced in nature. In such data sets majority class contains much more samples as compared to the minority class which contains very few samples. Because of this imbalance classifier may biased towards the majority samples and may misclassify the samples from the minority one. Standard classification algorithms also fail to classify such form of imbalanced data accurately with least misclassification error. The misclassification cost of minority sample is always much more than the misclassification cost of majority sample. Hence it is essential to resolve the imbalanced learning problem and classify the data more precisely. Consider the mammography data set which consist of 700 majority samples (non- cancerous) and 300 minority samples (cancerous). If such data set is given to any machine learning classifier, then as there is an imbalance in the data set, classifier may misclassify the samples from minority class into majority, i.e. cancer patient may be classified as a non-cancerous. Therefore, it is evident that in this domain, we need a classifier which provides high accuracy for the minority samples.

In order to deal with imbalance problem 4 major solutions are provided in the literature, namely sampling, active learning, cost sensitive learning and kernel based methods. Sampling based methods provide the solution at data level by balancing the number of samples among classes. Undersampling and oversampling are two key categories of sampling in which samples are either reduced from majority class or samples are added in the minority class. Both techniques have their own advantages as well as drawbacks. Active learning approaches focus mainly on acquiring labels to the unlabeled data. Another method is cost based method which provides solution to an imbalanced dataset at the algorithmic level. It uses cost matrix which represents costs associated with each representation. Besides of these methods, kernel based methods also work well in handling imbalanced datasets.

This paper proposes MLP-based undersampling technique which selects only important samples from the majority class for the training of MLP. Importance of samples is decided by computing a stochastic sensitivity measure (SM) value. To preserve the distribution of information, this technique divides the majority class into a number of clusters and from these clusters only most important samples are selected for SM evaluation. Section II provides a brief review on related works. The MLP based undersampling technique is presented in Section III. Section IV shows experimental comparisons between the proposed technique and current methods, and we conclude this paper in Section V.

2. BACKGROUND
There are four major categories that are useful in handling imbalanced data sets, which are nothing but sampling based methods, cost based methods, kernel based methods and active learning methods. This section provides a brief review on methods of imbalanced learning from sampling category only. Remaining methods can be found in [1] along with the nature of the imbalanced learning problem, approaches, various assessment metrics, major opportunities and challenges.

There are many undersampling techniques available in the literature. In random undersampling samples from majority class are randomly removed from the majority class to balance the data set. The Main deficiency of this method is that some important information may be loss. To overcome this problem, many researches proposed various undersampling techniques based on some statistical Knowledge. In [2] two methods, namely EasyEnsemble and BalanceCascade have been proposed. In EasyEnsemble technique, majority class is sampled into a number of subsets having size equals to the size of a minority class. Then for

each subset and entire minority class a learner is trained and output of those learners is then combined to get the final results. In case of BalanceCascade, learners are trained sequentially. The samples which are correctly classified from majority class are removed to avoid duplication. The KNN based approach has been proposed in [3]. This paper proposes four different methods for choosing majority training samples Near Miss 1, Near Miss 2, Near Miss 3 and most distant method. In [4] a new undersampling technique which is known as One Sided Selection (OSS) method has been proposed which keeps only important samples from the majority class to balance the data. This selection is done using minority class and one randomly selected majority sample along with KNN algorithm. The main drawback of this method is that, overall performance is dependent on the randomly selected majority sample. To solve this issue, a method CluterOSS has been proposed by researchers in [5] which is the adoption of OSS method. In this technique, samples of majority class are clustered using k-means algorithm and samples which are closer to the center are taken for undersampling process. Many undersampling techniques do not consider distribution of information among classes. To this end, one novel approach has been proposed as diversified sensitivity based undersampling (DSUS) which preserves the distribution of information using clustering and SM evaluation [6].

As undersampling techniques remove samples, there may be loss of information. Many researchers have worked on generating new samples in the minority class samples to get balanced data. In random oversampling original minority samples are randomly replicated, which may leads to the over fitting problem [7]. Synthetic Minority Oversampling Technique (SMOTE) has been proposed which generates the new synthetic sample for each minority sample. Initially it selects randomly a nearest neighbor of candidate sample. Then calculate the difference between candidate sample and its neighbor and multiply this difference by a random number in range of 0 and 1. This difference is added to the original candidate sample to get the new synthetic sample [8]. As SMOTE generates new samples for each minority, it may leads to overgeneralization. Also, it generates synthetic samples regardless of majority class, hence overlapping between classes increase. To overcome these deficiencies, many adaptations such as borderline-SMOTE [9], safe-level SMOTE [10], local neighborhood-based SMOTE [11], rough set theory based SMOTE [12], and Enhanced SMOTE [13] has been proposed. In borderline-SMOTE the samples located nearest to the decision boundary are identified first. These samples are also called as seed samples which are further used for synthetic sample generation. In case of safe level SMOTE, it generates the new samples along the same line as SMOTE does but with different safe levels. The safe level of any minority sample is nothing but a number of minority samples in its k nearest neighbors. This technique generates new synthetic sample closer to the larger safe level so that new instances will lie within the minority class. This will eliminate the problem of overlapping. A safe-level SMOTE gives better results than SMOTE and borderline-SMOTE.

In [14] research, the Adaptive Synthetic Sampling Technique has been proposed in which synthetic samples are generated according to the distribution of information. More synthetic samples are generated for those samples which are difficult for learning compared to those which are easy to learn. Another technique, namely, Ranked Minority Oversampling and Boosting (RAMOBoost) has been proposed in [15] which adaptively assigns a rank to each minority instance at every iteration and generates synthetic samples according to probability distribution which is based on the distribution of information. Furthermore, when data changes across time, incremental learning is required. Very few works have been done in dealing with imbalance problems in incremental learning [16-18]. Hence imbalanced data is a vital issue in incremental learning for many web-based real-world applications.

3. PROPOSED FRAMEWORK: MLPUS

The MLPUS involves three key mechanisms: a) clustering of majority class samples b) selection of important samples using SM evaluation c) training of MLP using selected samples in SM evaluation. Figure 1 shows the overall flow of MLPUS. When the imbalanced data is given, the proposed system initially cluster both classes separately into k clusters where, $k = \lfloor \sqrt{N_p} \rfloor$.

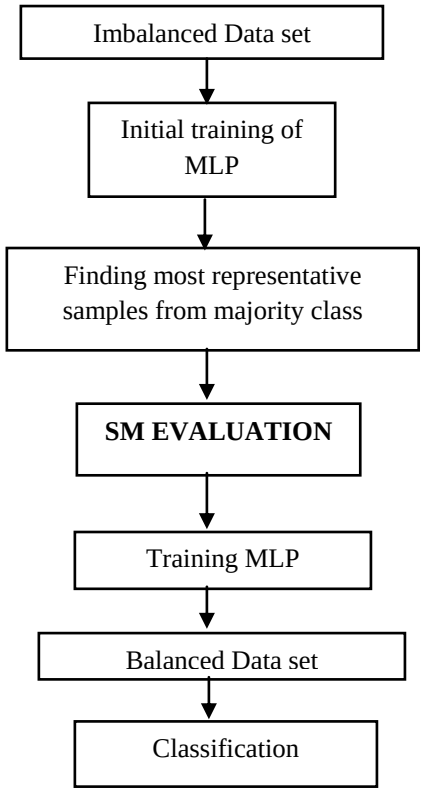

Figure 1: Work flow of MLPUS

Here k-means algorithm is used for clustering. Then from each cluster the sample located closest to the centroid of the cluster is taken and added into the training data set. In this way, in the initial training of MLP, we will get an equal number of samples from majority as well as minority class. The value of p remains constant at every iteration. These training samples are removed from the original data set. In 3.1 we present MLP training algorithm. The key step of undersampling i.e. SM evaluation is presented in 3.2.

As the number of majority class samples are much more than minority class samples, we cluster these majority samples into N_p number of clusters so that only important samples will take part in undersampling and distribution of information is also

Communications on Applied Electronics (CAE) – ISSN : 2394-4714
Foundation of Computer Science FCS, New York, USA
Volume 5 – No.7, July 2016 – www.caeaccess.org

preserved. The MLPUS chooses a sample near to the centroid of these N_p clusters and their SM values are then computed. The *k* samples having highest SM values will be selected. Similarly, SM values of all minority samples are computed and *k* samples are selected having largest SM. These 2*k* samples are then added to the training data set so that MLP will get balanced training data set. In every iteration training data set consists of *2tk* number of samples where *t* is the number of iterations and its value cannot be greater than *k*. The samples which are selected in this process are removed iteratively from the original data set and this process will repeat till minority samples are more than *k*.

Notations:

S_{maj} : Set of majority samples

S_{min} : Set of minority samples

N_p: Number of samples in minority class

Problem Description:

Let S be the system,

S= $\{S_{maj}, S_{min}\}$

Where, $S_{maj} > S_{min}$

Algorithm:

Step 1: Training the initial MLP

a. Cluster both S_{maj} and S_{min} into $k = \lfloor\sqrt{N_p}\rfloor$ clusters each.
b. Let A_0 and B_0 be the empty sets.
c. From each *k* cluster of the minority class, add the sample located closest to its center to A_0
d. From each *p* cluster of the majority class, add the sample located closest to its center to B_0
e. $S_{min} = S_{min} - A_0$, $S_{maj} = S_{maj} - B_0$, $S = A_0 \cup B_0$ and d = 0

Step 2: Train MLP using S

While N_p > k do

Step 3: Find most important samples from majority class

a. Cluster S_{maj} into N_p number of clusters.
b. Let C, A_d, B_d be the empty sets and d = d+1.
c. From each cluster of S_{maj} select the sample located closest to its center as an important sample and add this to set C.

Step 4: Compute the value of SM for each sample of C and S_{min} as

$$I(x) = \frac{1}{H}\sum_{h=1}^{H}(g(x+\Delta x_h) - g(x))^2$$

Step 5: Add p samples from C and S_{min} having largest SM value to set A_d and B_d respectively.

Step 6: $Smin = Smin - Ad, Smaj = Smaj - Bd,$

$$S = S \cup Ad \cup Bd$$

Step 7: Train a MLP using S.

End while.

Illustration:

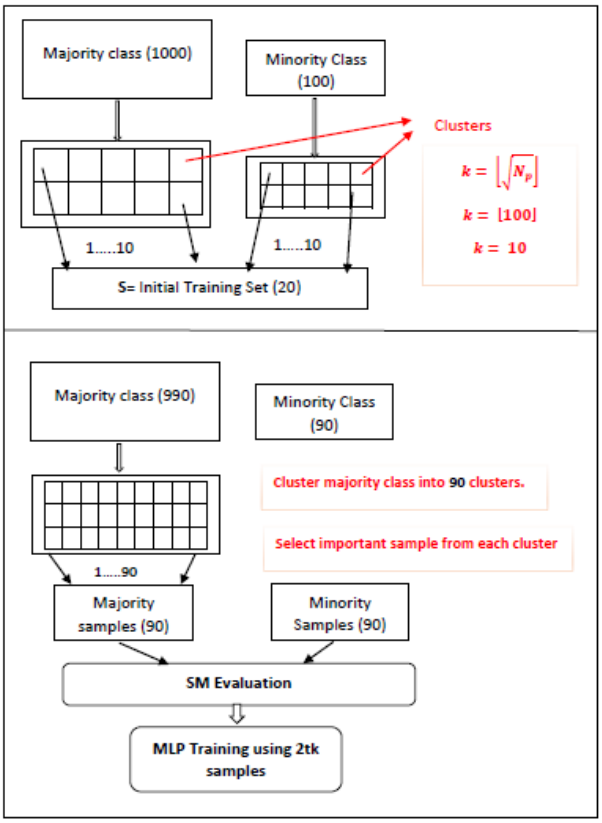

Figure 2: Illustration of MLPUS

Figure 2 shows one example of how MLPUS selects the samples for MLP training and SM evaluation. Upper partition represents an initial training of MLP. If there are 1000 samples in majority class and minority samples are only 100, then both classes are clustered into 10 clusters. From each cluster one sample is selected and added into the initial training data set. These samples are then removed from the original data set. Hence, in lower partition majority class contains 990 samples and minority class contains 90 samples. Then the only majority class is clustered into 90 clusters. Then from each of these clusters only representative samples are selected. Hence 90 samples from majority class and 90 samples from minority class are given for SM evaluation.

3.1 MLP Training

The utmost standard Neural Network is the Multi-Layer Perceptron (MLP) architecture, in which back propagation is used for training the model. It entails minimum three layers: an input layer, an output layer and one or more hidden layers. Firstly, connection weights are initialize randomly and learning rate (η) is chosen. If the learning rate is very less then learning will be too slow and if it is very high, then learning will not be done properly by MLP. Hence the value of η must be appropriate as it affects the performance of MLP. Then for each input sample (X) consider $x_1, x_2, x_3 \ldots x_n$ are the

input features and $w_1, w_2, w_3 \dots w_n$ are their corresponding weights. Then output at each neuron= $x_1 w_1 + x_2 w_2 + \cdots + x_n w_n$

This output is propagated at each layer and for output unit error is computed as

Error= Expected output – actual output

This error is then back propagated and weights are updated accordingly.

MLP can be defined as below:

$$g(x) = \sum_{j=1}^{M} w_{kj} f(\sum_{i=1}^{n} w_{ji} x_i)$$

Where,

w_{kj} = connection weight between output neuron (k) and hidden neuron (j),

w_{ji} = connection weight between hidden neuron (j) and input neuron (i),

M = number of hidden neurons

f(x) denotes the sigmoid function and can be defined as

$$f(x) = \frac{1}{1 + e^{-x}}$$

3.2 SM Evaluation

In [19] localized generalization error model has been proposed in which SM of a RBFNN was computed for the selection of RBFNN architecture. But this technique does not calculate SM value for individual instance. In [20] for the hyper-parameter selection sensitivity of each sample is measured by using SM computation for SVM. In [6] SM of RBFNN has been proposed which is used for evaluation of each sample for undersampling. This SM value will measure the output fluctuations of RBFNN. In this paper, we put forward computation of SM for MLP which will be used as main criterion for undersampling. The SM can be defined as squared difference between output of original sample and output of future unseen sample. If any small change is made to input features of sample then how its output is perturbed is measured using SM computation. The samples which are hard to learn will get largest value so that these samples will iteratively added in the training set and MLP will not misclassify them. Equation (3) shows the SM computation for each training sample x.

$$I(x) = \frac{1}{H} \sum_{h=1}^{H} (g(x + \Delta x_h) - g(x))^2$$

Where, Δx_h is halton point and function g(x) can be defined as follows

$$g(x) = \sum_{j=1}^{M} w_{kj} f(\sum_{i=1}^{n} w_{ji} x_i)$$

Where, M indicates the number of hidden neurons and function f(a) is a sigmoid function.

4. EXPERIMENTS AND RESULT ANALYSIS

As the proposed technique uses k-means algorithm for clustering of samples, in the initial training of MLP the value of k is equal to the square root of the number of samples in the minority class. To select the representative samples from the majority class, it is again clustered using k-means and here, k is equal to the number of samples in the minority class. Here, Multilayer Perceptron (MLP) is used for the classification. Performance of MLP is depends mainly on values of learning rate and epoch. In table1 different values of learning rate are taken and how these values affect the accuracy and other parameters is recorded. The value of learning rate should lie in between 0 and 1.

Table 1: Results on values on change in learning rate

Learning Rate	**0.1**	**0.3**	**0.5**
Accuracy	95.21	96.82	96.11
Kappa Statistics	0.910	0.936	0.921
Mean Absolute Error	0.072	0.054	0.063
Relative Absolute Error	10.97	10.95	9.23

From above results it is clear that if learning rate is 0.3 then MLP will give more accurate results. If learning rate is too high then accuracy may increase, but in that case MLP cannot learn properly. Similar experiment can be done to find out best value of epoch. Here epoch value is set to 300. The number of input features equals to the number of input neurons. The number of hidden neurons are set to 6 and output neurons is kept 2. For the activation function sigmoid function is chosen.

In this paper, in order to evaluate performance of MLPUS various experiments are carried out on 5 real world data sets which are taken from UCI repository. Table 2 shows characteristics of these data sets in the form of number of attributes, number of minority and majority samples and imbalance ratio. All data sets are in binary form.

Table 2: Description of real world data sets

Dataset	**Minority samples**	**Majority Samples**	**Imbalance Ratio**
Pima Diabetes	120	499	0.35:0.65
Breast Cancer	85	201	0.35:0.65
Hepatitis	32	123	0.21:0.79
Mammographic	445	516	0.46:0.54
Liver Disorder	145	200	0.42:0.57

Communications on Applied Electronics (CAE) – ISSN : 2394-4714
Foundation of Computer Science FCS, New York, USA
Volume 5 – No.7, July 2016 – www.caeaccess.org

To evaluate the performance of proposed work various performance measures can be derived from the confusion matrix such as precision, recall, overall accuracy and G-mean. There are other parameters such as kappa statistics, mean absolute error and relative absolute error which can be used for the evaluation. Table 3 shows all these measures with respect to above data sets.

Table 3: Performance measures of MLPUS

Dataset	Precision	Recall	G-mean	Overall Accuracy
Pima Diabetes	0.862	0.85	0.86	86.05
Breast Cancer	0.968	0.97	0.963	96.29
Hepatitis	0.98	0.867	0.931	97.05
Mammographic	0.85	0.86	0.85	85.18
Liver Disorder	0.83	0.84	0.84	84.25

As this undersampling technique uses MLP as a classifier, there are another efficient classifiers which may improve the classification results of the system. If MLP is replaced by CART or SVM then in some cases MLP gives good results but in most of the cases SVM gives better results in terms of accuracy.

Table 4: Performance of various classifiers

Dataset	MLP	CART	SVM
Pima Diabetes	86.05	94.42	**98.90**
Breast Cancer	**96.29**	93.12	88.35
Hepatitis	97.05	85.29	**98.75**
Mammographic	85.18	87.19	**89.31**
Liver Disorder	84.25	94.75	**99.39**

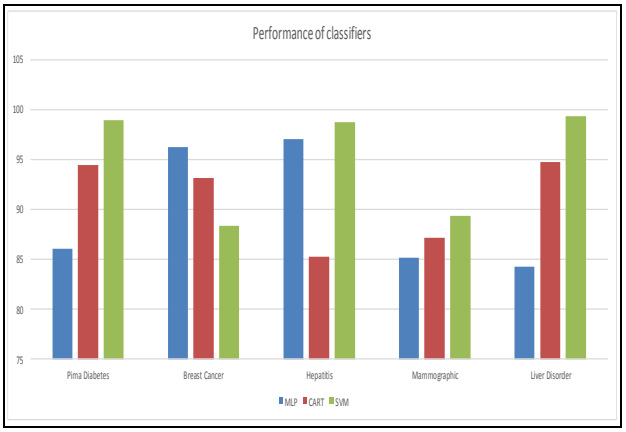

Figure 3: Performance of classifiers

The proposed undersampling technique can be compared with a popular technique known as SMOTE with 100% oversampling rate and 5 nearest neighbors. Also it can be compared with resampling technique. The performance of these techniques are compared in the form of accuracy. In table 5, second column shows results when no any sampling technique is applied to the data sets. In third column results of resampling technique are shown. While column 4 and 5 contains results of SMOTE and MLPUS respectively. All these results shows that MLPUS performs better than other techniques.

A= Without Sampling

B= Resampling

C= SMOTE

D= MLPUS

Table 5: Comparison of MLPUS with other techniques

Dataset	A	B	C	D
Pima Diabetes	81.90	**89.82**	81.46	86.05
Breast Cancer	64.68	87.06	71.43	**96.29**
Hepatitis	78.48	93.67	81.52	**97.05**
Mammographic	80.95	81.99	83.49	**85.18**
Liver Disorder	71.59	68.11	69.79	**84.25**

In this section another one experiment is performed on pima diabetes data set for SM evaluation. When this data set is given to the MLPUS, then SM values of certain instances are recorded. We collect these SM values of both majority as well as minority samples as shown in table 6. For particular samples SM values are very high as these samples are hard to learn for the classifier.

Table 6: SM values of minority and majority samples

No of samples	SM values of minority samples	SM values of majority samples
0-5	0.021	0.086
6-10	0.026	0.006
11-15	0.253	0.099
16-20	3.480	1.52
21-25	0.007	0.082
26-30	1.526	0.110
31-35	0.030	0.027
36-40	0.020	0.002
41-45	2.628	0.017
46-50	5.199	0.271

Following graph shows the representation of this SM evaluation. From this graph it is clear that SM values of particular majority samples are greater than that of the

minority samples. But for most of the minority samples SM values are high as these samples are difficult for learning.

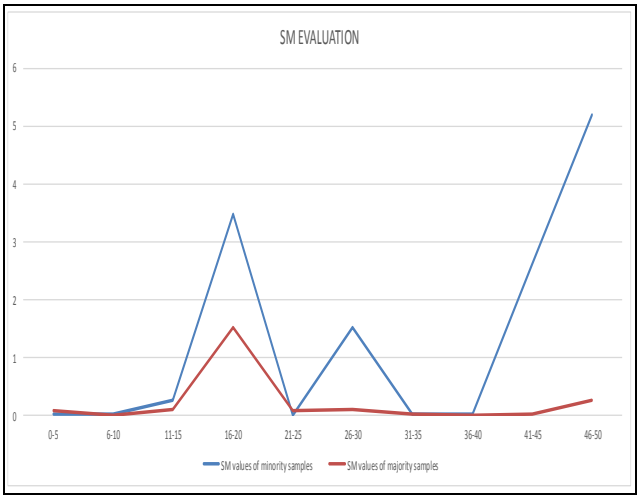

Figure 4: SM values of majority and minority samples

As shown in this graph initially SM values for majority samples are greater than minority, but at the end there is huge difference between values of both classes. Minority values are much more than majority.

5. CONCLUSION

As many undersampling techniques exists for imbalanced learning problem, this proposed MLPUS technique preserves the distribution of information among the classes. It selects the most important samples from the majority class. As SM evaluation is used for undersampling purpose, hard to learn samples are iteratively added in the training data sets and will not be misclassified by the classifier. Instead of using MLP if SVM is used for classification then more accurate results can be obtained. MLPUS also performs much better than other sampling techniques. SM evaluation of this method helps to identify important samples for undersampling. As here k-means algorithm is used for clustering, it can be replaced by other clustering mechanisms in order to improve the performance of MLPUS. Several future adaptions can be made to this technique. This proposed technique can be integrated with other oversampling techniques in order to investigate whether 2 techniques will give better results together. This undersampling technique can be extended for multiclass imbalance problem. This technique can be used to resolve the imbalanced problem occurring in the incremental learning.

6. REFERENCES

[1] H. He and E.A. Garcia, Learning from Imbalanced Data, IEEE Trans. Knowledge Data Eng., vol. 21, no. 9, pp. 1263-1284, Sept. 2009.

[2] X.Y. Liu, J.Wu, and Z.H. Zhou, Exploratory Under Sampling for Class Imbalance Learning, Proc. Intl Conf. Data Mining, pp. 965- 969, 2006.

[3] J. Zhang and I. Mani, KNN Approach to Unbalanced Data Distributions: A Case Study Involving Information Extraction, Proc. Intl Conf. Machine Learning, Workshop Learning from Imbalanced Data Sets, 2003.

[4] M. Kubat and S. Matwin, Addressing the Curse of Imbalanced Training Sets: One-Sided Selection, Proc. Intl Conf. Machine Learning, pp. 179-186, 1997.

[5] Victor H. Barella, Eduardo p. Costa, and Andre C P L F Carvalho, ClusterOSS: a new undersampling method for imbalanced learning

[6] Wing W. Y. Ng, Junjie Hu, Daniel S. Yeung, Shaohua Yin, and Fabio Roli, "Diversified Sensitivity-Based Undersampling for Imbalance Classification Problems", IEEE Trans. Cybernetics vol. 45, no. 11, Nov. 2015.

[7] H.He, Self-Adaptive Systems for Machine Intelligence,Wiley, Aug 2011

[8] N.V. Chawla, K.W. Bowyer, L.O. Hall, and W.P. Kegelmeyer, "SMOTE: Synthetic Minority oversampling Technique",J. Artificial Intelligence Research, vol. 16, pp. 321-357, 2002.

[9] H. Han, W.Y. Wang, and B.H. Mao, "Borderline-SMOTE: A New Oversampling Method in Imbalanced Data Sets Learning", Proc. Intl Conf. Intelligent Computing, pp. 878-887, 2005.

[10] C. Bunkhumpornpat, K. Sinapiromsaran, and C. Lursinsap, "Safe-level-SMOTE: Safe level-synthetic minority over-sampling technique for handling the class imbalanced problem," in Advances in Knowledge Discovery and Data Mining. Berlin, Germany: Springer, 2009, pp. 475 482, 2009.

[11] T. Maciejewski and J. Stefanowski, "Local neighbourhood extension of SMOTE for mining imbalanced data," in Proc. IEEE Symp. Comput. Intell. Data Min. (CIDM), Paris, France, pp. 104111, 2011.

[12] E. Ramentol, Y. Caballero, R. Bello, and F. Herrera, "SMOTE-RSB*: A hybrid preprocessing approach based on oversampling and undersampling for high imbalanced data-sets using SMOTE and rough sets theory," Knowl. Inf. Syst., vol. 33, no. 2, pp. 245265, 2012.

[13] Reshma C. Bhagat and Sachin S. Patil, "Enhanced SMOTE Algorithm for Classification of Imbalanced Big-Data using Random Forest", IEEE International Advance Computing Conference (IACC), 2015.

[14] H. He, Y. Bai, E.A. Garcia, and S. Li, "ADASYN: Adaptive Synthetic Sampling Approach for Imbalanced Learning", Proc. Intl Joint Conf. Neural Networks, pp. 1322-1328, 2008.

[15] S. Chen, H. He, and E.A. Garcia, "RAMOBoost: Ranked Minority Oversampling in Boosting", IEEE Trans. Neural Networks, vol. 21, no. 20, pp. 1624-1642, Oct. 2010.

[16] Ade, Roshani, and P. R. Deshmukh. "Instance-based vs Batch-based Incremental Learning Approach for Students Classification." International Journal of Computer Applications 106.3 (2014).

[17] Ade, Roshani, and Prashant Deshmukh. "Efficient knowledge transformation for incremental learning and detection of new concept class in student's classification system." Information Systems Design and Intelligent Applications. Springer India, 2015. 757-766.

[18] Kulkarni, Pallavi Digambarrao and Roshani Ade. "Learning from Unbalanced Stream Data in Non-Stationary Environments Using Logistic Regression Model: A Novel Approach Using Machine Learning for Assessment of Credit Card Frauds." Handbook of Research on Natural Computing for Optimization Problems. IGI Global, 2016. 561-582. Web. 9 Jun. 2016. doi:10.4018/978-1-5225-0058-2.ch023

[19] D. S. Yeung, W. W. Y. Ng, D. Wang, E. C. Tsang, and X.-Z. Wang, "Localized generalization error model and its application to architecture selection for radial basis function neural network," IEEE Trans. Neural Netw., vol. 18, no. 5, pp. 1294–1305, Sep. 2007.

[20] B. Sun, W. W. Y. Ng, D. S. Yeung, and P. P. K. Chan, "Hyper-parameter selection for sparse LS-SVM via minimization of its localized generalization error," Int. J. Wavelets Multiresolut. Inf. Process., vol. 11, no. 3, 2013, Art. ID 1350.