# Classification of Gene Expression Data using Gaussian Restricted Boltzmann Machine (GRBM) – An Application on Human Lung Adenocarcinoma Data

*(GRBM)*

Jit Gupta
Dept. of Computer Sc. & Engineering
NetajiSubhash Engineering College
Techno City, Garia, Kolkata
*Email: dotgupta@gmail.com*

Indranil Pradhan
Dept. of Computer Sc. & Engineering
NetajiSubhash Engineering College
Techno City, Garia, Kolkata
*Email: indranil.pradhan@yahoo.in*

Anupam Ghosh*
Dept. of Computer Sc. & Engineering
NetajiSubhash Engineering College
Techno City, Garia, Kolkata
*Email: anupam.ghosh@rediffmail.com*

***Abstract*—In t**his article, the work deals with the classification of gene expression data using a Gaussian Restricted Boltzmann Machine (a Machine Learning model concerning Neural Networks). An RBM is a generative stochastic artificial neural network that contains one single layer of visible units and another single layer of hidden units. It is usually used to reconstruct or classify image data using the contrastive divergence method but in our work, we have applied and used it on a binary classification problem to classify whether a certain human has been affected by lung adenocarcinoma or not depending on his or her gene expression values. To tackle the class imbalance problem, safe-level SMOTE algorithm was used to over sample the minority class and a Random Forest was used as a gene selector. On comparing the results produced by RBM with a k-NN classifier and a decision tree classifier, we found that the former over fit the data while the latter produced results comparable with the RBM, thus proving that our model learns the data efficiently and accurately. This proves that RBM can be used in future classification cases that deal with gene expression values irrespective of the number of data points and the number of genes.

***Keywords****-Artificial neural networks, Classification, Machine learning, Neural Networks, Predictive models*

*****

## I. INTRODUCTION

Lung Adenocarcinoma [1] attributes for almost 40% of lung cancer cases in western countries. This highlights its prevalence in today's world. It is highly associated with cases of cancer caused by smoking. However, it has been seen that there has been a gradual increase in its numbers even in non smokers.

In this study, we have dealt with a problem of classifying [2] whether a human has lung adenocarcinoma or not where the data is gene expression data. We have used a Restricted Boltzmann Machine as the classifier [3]. Instead of using the much acclaimed Bernoulli's RBM, we used the Gaussian RBM [4] which takes inputs from [0,1] and interprets the inputs as a probabilistic distribution. For training the model, we used the Contrastive Divergence [5] method. The dataset used for application was an NCBI dataset of Lung Adeno Carcinoma (Homo sapiens), having 7129 genes and 96 data points out of which 86 were diseased and 10 were normal. Out of 7129 genes, only 4966 were considered as only they were a part of the probeset. The entire dataset was normalized to a range of [0,1]. A Random Forest [6] was used to select 8 genes out of 4966 by applying a ranking based on their impurity score. However, a major obstacle in the dataset was the class imbalance [7] present. The minority class had 10 data points while the majority class had 86. To overcome this obstacle, we used three different techniques and compared the results – Arithmetic Mean, Geometric Mean, safe-level SMOTE [8]. The result produced by the safe-level SMOTE was much more efficient than AM and GM. The synthetic data produced by both AM and GM resulted in a classification accuracy percentage of 50 only while using safe-level SMOTE increased it drastically to 97.22%.k-NN [9] and decision tree[9] classifier were also used for comparison and it was observed that both models produced results comparable to our model. This proves effectively that a Restricted Boltzmann Machine can be taken into serious consideration for binary classification problems containing gene expression values.

## II. METHODOLOGY

In this section we describe the methodology of the entire process, breaking it down into the following steps:-

- Normalization
- Gene selection
- Oversampling of minority class
- Creation of dataset
- Training and testing Gaussian RBM

### *A. NORMALIZATION*

The given data contains gene expression values within different ranges for different genes. The following formula is used for normalizing[10][11] each values for each gene –

$$Z = (X_i – \min(X))/( \max(X) – \min(X))$$

where $X=(X_1,...,X_n)$ and $Z_i$ is the $i^{th}$ normalized data point.

### *B. GENE SELECTION*

Random Forest

Random forests [6] are among the most popular machine learning methods thanks to their relatively good accuracy, robustness and ease of use. They also provide two straightforward methods for feature selection: mean decrease impurity and mean decrease accuracy.

Random forest consists of a number of decision trees [9]. Every node in the decision trees is a condition on a single feature, designed to split the dataset into two so that similar response values end up in the same set. The measure, based on which the (locally) optimal condition is chosen, is called impurity. For classification, it is typically either Giniimpurity [9]or information gain/entropy and for regression trees it is variance. Thus when training a tree, it can be computed how much each feature decreases the weighted impurity in a tree. For a forest, the impurity decrease from each feature can be averaged and the features are ranked according to this measure.

Usage of the random forest resulted in the selection of the following genes (descending order of impurity score) which were used for classification:-

- **CUTL1**
- **GBA**
- **FEZ1**
- **FH**
- **CLN3**
- **PTPN1**
- **HRAS**
- **AFFX-ThrX-3**

### C. *OVERSAMPLING OF MINORITY CLASS*

Class Imbalance Problem[12][13] is a problem in machine learning where the total number of a class of data (positive) is far less than the total number of another class of data (negative). This problem is extremely common in practice and can be observed in various disciplines including fraud detection, anomaly detection, medical diagnosis, oil spillage detection, facial recognition, etc.

Safe-level SMOTE

Based on SMOTE[14], Safe-Level-SMOTE[8], Safe-Level-Synthetic Minority Oversampling Technique, assigns each positive instance its safe level before generating synthetic instances. Each synthetic instance is positioned closer to the largest safe level so all synthetic instances are generated only in safe regions. The safe level (sl) is defined as formula (1).

If the safe level of an instance is close to 0, the instance is nearly noise. If it is close to k, the instance is considered safe. The safe level ratio is defined as formula (2). It is used for selecting the safe positions to generate synthetic instances.

safe level (sl) = the number of a positive stances in k nearest neighbours . (1)

safe level ratio = sl of a positive instance /sl of a nearest neighbours . (2)

Pseudo Code for Safe-Level-SMOTE

*Input: a set of all original positive normalized (II.A) instances D*

*Output: a set of all synthetic positive instances D'*

```
D' = Ø
for each positive instance p in D {
    compute k nearest neighbours for p in D and randomly select one from the k nearest neighbours, call it n
    slp = the number of positive stances in k nearest neighbours for p in D
    sln = the number of positive stances in k nearest neighbours for n in D
    if (sln ≠ 0) { ; sl is safe level.
        sl_ratio = slp / sln ;sl_ratio is safe level ratio.
    }
    else {
        sl_ratio = ∞
    }
    if (sl_ratio = ∞ AND slp = 0) { ; the 1st case
        does not generate positive synthetic instance
    }
    else {
        for (atti = 1 to numattrs) { ; numattrs is the number of attributes.
            if (sl_ratio = ∞ AND slp ≠ 0) { ; the 2nd case
                gap = 0
            }
            else if (sl_ratio = 1) { ; the 3rd case
                random a number between 0 and 1, call it gap
            }
            else if (sl_ratio> 1) { ; the 4th case
                random a number between 0 and 1/sl_ratio, call it gap
            }
            else if (sl_ratio< 1) { ; the 5th case
                random a number between 1-sl_ratio and 1, call it gap
            }
            dif = n[atti] - p[atti]
            s[atti] = p[atti] + gap·dif
        }
        D' = D' ∪{s}
    }
}
return D' F
```

### D. *CREATION OF DATASET*

For creation and validation of the dataset, 10 fold Cross Validation[15] was used. In *10*-fold cross-validation, the original sample is randomly partitioned into *10* equal sized subsamples. Of the *10* subsamples, a single subsample is retained as the validation data for testing the model, and the remaining *9* subsamples are used as training data. The cross-validation process is then repeated *10* times (the *folds*), with each of the *9* subsamples used exactly once as the validation data. The *10* results from the folds can then be averaged to produce a single estimation. The advantage of this method over repeated random sub-sampling (see below) is that all observations are used for both training and validation, and each observation is used for validation exactly once.

After cross validation, the given dataset was divided into different subsets of training set $V_i$ and a testing set $Y_i$. A training set is associated with an allied training label set. Let us take an example of one such subset –

Out of 172 data points (86 diseased and 86 normal after oversampling of normal class), 100 were fed as training data

(50 as diseased labels and 50 as normal labels) and 72 were fed as testing data.

Algorithm:

*Input – Array of selected features (X) from II.B, Oversampled dataset D from II.C Output – Training dataset $V_i$, testing dataset Y, training labels $A_i$*

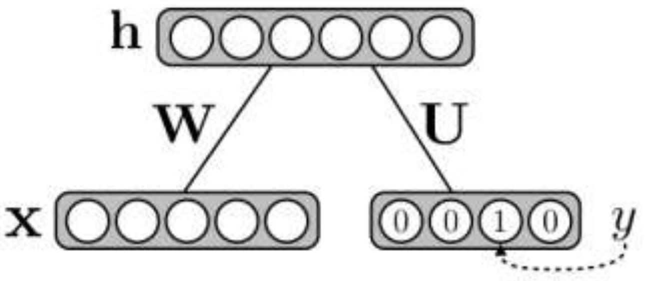
a) *Columns of data are chosen corresponding to the selected features to form matrix $(VY)_i$*
b) *Matrix $(VY)_i$ is broken into training dataset $V_i$ and testing dataset $Y_i$ based on the ratio 25:18*
c) *Training labels are created in the ratio of 1:1 for diseased and normal data in the array A*
d) *Array $A_i$, matrices $V_i$ and $Y_i$ are fed to the GRBM*

### E. TRAINING AND TESTING GRBM

A **restricted Boltzmann machine** (**RBM**)[16] is a generative stochastic artificial neuralnetwork that can learn a probability distribution over its set of inputs.

They can be trained in either supervised or unsupervised ways, depending on the task.As their name implies, RBMs are a variant of Boltzmann machines, with the restriction that their neurons must form a bipartite graph: a pair of nodes from each of the two groups of units (commonly referred to as the "visible" and "hidden" units respectively) may have a symmetric connection between them; and there are no connections between nodes within a group. By contrast, "unrestricted" Boltzmann machines may have connections between hidden units. This restriction allows for more efficient training algorithms than are available for the general class of Boltzmann machines, in particular the gradient-based **contrastivedivergence** [12] algorithm. However, we have used the Stochastic Maximum Likehood algorithm.

A Gaussian RBM was used as it takes real values between [0,1] as inputs and models the inputs as a probabilistic distribution. The hidden nodes however work on binary values just like its Bernoulli counterpart.



In the given figure, X is our training set and Y is our training labels.

Algorithm:

*Input: training pair* $(y^i, x^i)$ *and learning rate* $\lambda$
*Notation:* $a \leftarrow b$ *means a is set to value b*
$a \sim p$ *means a is sampled from p*
*Positive phase* $y^0 \leftarrow y^i,\ x^0 \leftarrow x^i,\ hb^0 \leftarrow sigm(c + Wx^0 + U \sim y^0)$
*Negative phase* $h^0 \sim p(h|y^0\ , x^0\ ),\ y^1 \sim p(y|h^0\ ),\ x^1 \sim p(x|h^0\ )\ hb^1 \leftarrow sigm(c + Wx^1 + U \sim y^1)$
*Update for* $\theta \in \Theta$ *do* $\theta \leftarrow \theta - \lambda$ “ $\partial\ \partial\theta\ E(y^0\ , x^0\ , hb^0\ ) - \partial\ \partial\theta\ E(y^1\ , x^1\ , hb^1\ )$ ” *end for*

As we have used the Stochastic Maximum Likelihood (also known as the Persistent Contrastive Divergence method)[12] we use mini batches for training and run a single Markov chain over it. Thus it is like a CD-n algorithm instead of CD-1 with the Markov chain parameters not being reset in each initialization.

### F. Equations

The standard type of RBM has binary-valued (Boolean/Bernoulli) hidden and visible units, and consists of a matrix of weights $W=W_{ij}$ (size $m \times n$) associated with the connection between hidden unit $h_j$ and visible unit $v_i$. Given these, the *energy* of a configuration (pair of boolean vectors) $(v,h)$ is defined as

$$E(v,h) = -\sum_i a_i v_i - \sum_j b_j h_j - \sum_i \sum_j v_i w_{i,j} h_j$$

or, in matrix notation,

$$E(v,h) = -a^{\mathrm{T}} v - b^{\mathrm{T}} h - v^{\mathrm{T}} W h$$

This energy function is analogous to that of a Hopfield network. As in general Boltzmann machines, probability distributions over hidden and/or visible vectors are defined in terms of the energy function:

$$P(v) = \frac{1}{Z} \sum_h e^{-E(v,h)}$$

where Z is a partition function (in other words, just a normalizing constant to ensure the probability distribution sums to 1). Similarly, the (marginal) probability of a visible (input) vector of booleans is the sum over all possible hidden layer configurations:

$$P(v,h) = \frac{1}{Z} e^{-E(v,h)}$$

Since the RBM has the shape of a bipartite graph, with no intra-layer connections, the hidden unit activations are mutually independent given the visible unit activations and conversely, the visible unit activations are mutually independent given the hidden unit activations. That is, for m visible units and n hidden units, the conditional probability of a configuration of the visible units*v*, given a configuration of the hidden units *h*, is

$$P(v|h) = \prod_{i=1}^{m} P(v_i|h)$$

Conversely, the conditional probability of *h* given *v* is

$$P(h|v) = \prod_{j=1}^{n} P(h_j|v)$$

The individual activation probabilities are given by

$$P(h_j = 1|v) = \sigma\left(b_j + \sum_{i=1}^{m} w_{i,j} v_i\right)$$

and

$$P(v_i = 1|h) = \sigma\left(a_i + \sum_{j=1}^{n} w_{i,j} h_j\right)$$

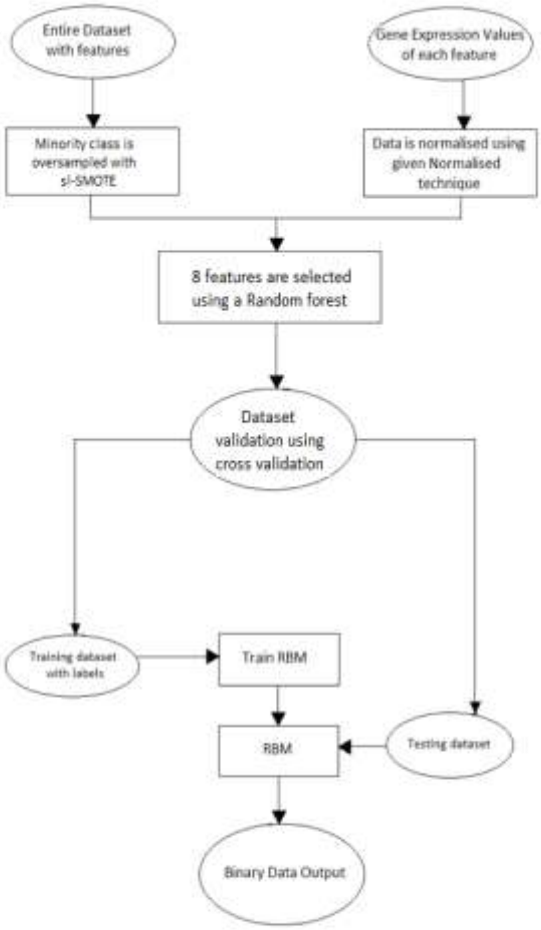
where sigma denotes the logistic sigmoid.

The visible units of RBM can be multinomial, although the hidden units are Bernoulli. In this case, the logistic function for visible units is replaced by the softmax[17] function

$$P(v_i^k = 1|h) = \frac{\exp(a_i^k + \Sigma_j W_{ij}^k h_j)}{\Sigma_{k'=1}^K \exp(a_i^{k'} + \Sigma_j W_{ij}^{k'} h_j)}$$

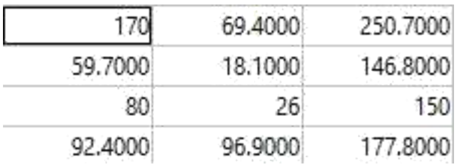
where$K$ is the number of discrete values that the visible values have.

Flowchart for process:


Entire Dataset with features
Gene Expression Values of each feature
Minority class is oversampled with sl-SMOTE
Data is normalised using given Normalised technique
8 features are selected using a Random forest
Dataset validation using cross validation
Training dataset with labels
Train RBM
RBM
Testing dataset
Binary Data Output

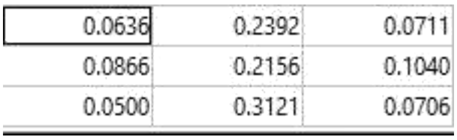
EXAMPLE:

Sample of dataset to be over sampled normalised:



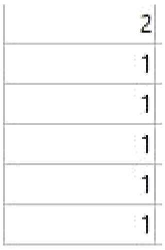
| | | |
|---|---|---|
| 170 | 69.4000 | 250.7000 |
| 59.7000 | 18.1000 | 146.8000 |
| 80 | 26 | 150 |
| 92.4000 | 96.9000 | 177.8000 |

Sample of normalised dataset:

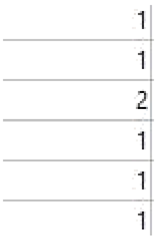


| | | |
|---|---|---|
| 0.0636 | 0.2392 | 0.0711 |
| 0.0866 | 0.2156 | 0.1040 |
| 0.0500 | 0.3121 | 0.0706 |

Sample of testing labels:Predicted labels:



| |
|---|
| 2 |
| 1 |
| 1 |
| 1 |
| 1 |
| 1 |



| |
|---|
| 1 |
| 1 |
| 2 |
| 1 |
| 1 |
| 1 |

## III. RESULTS AND DISCUSSIONS

### *A. DATASET DESCRIPTION:*

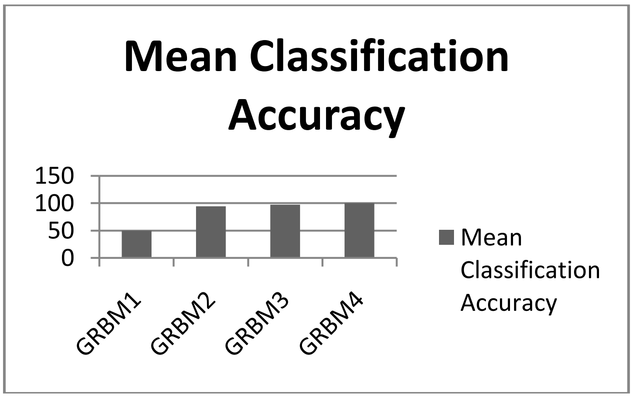
The dataset used for the application was a Lung Adeno Carcinoma dataset of Homo Sapiens having 99 columns and 7132 rows. Out of the 99 columns, 86 were diseased data points, 10 were normal data points, 2 had the probeset and genome names, and 1 identified the gene's presence in the probeset. Out of the 7132 rows, 2 had the Sample ID, 1 had the cluster ID and the remaining 7129 had the genes which were considered as features.The dataset was normalized using the formula given in II.A, and 8 features were selected using safe-level SMOTE. The entire implementation of the process was divided into parts. The feature selection and safe-level SMOTE on one platform; RBM, and data normalization on another platform. Python 3.5 was used for the gene scoring and gene selection using a Randomfores. It was also used to oversample the minority class using safe-level SMOTE. While the RBM Toolbox of MatLab R2013A was used for the RBM implementation.MatLab 2013A was also for the normalization of data.

### B. *ANALYSIS OF THE RESULTS*

***Parameters used in RBM:***

| | RBM1 | RBM2 | RBM3 | RBM4 |
|---|---|---|---|---|
| Learning rate | 0.1 | 0.5 | 0.1 | 0.5 |
| Momentum | 0.5 | 0.5 | 0.5 | 0.5 |
| Number of epochs | 1000 | 1000 | 1000 | 1000 |
| | | | | |
| Batch size | 100 | 10 | 10 | 100 |

Mean Classification Accuracy comparison:-



Therefore, RBM3 has been selected as our model. It has the following properties:

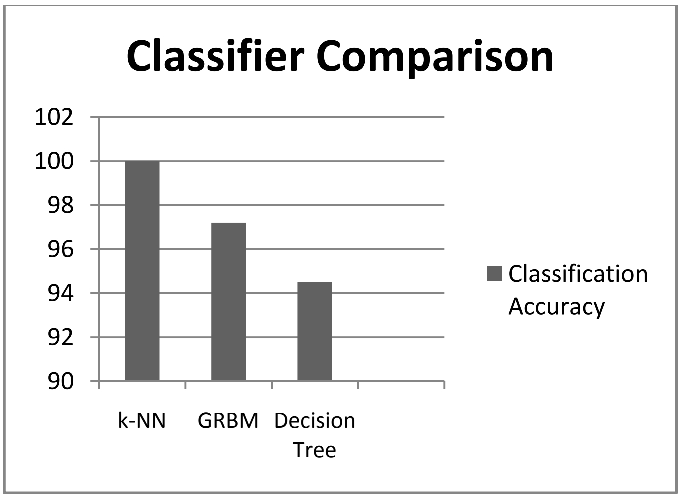
| **Method** | Stochastic Maximum Likelihood |
|---|---|
| **Learning rate** | 0.1 |
| **Momentum** | 0.5 |
| **Maximum Epochs** | 1000 |
| **Batch Size** | 10 |

### *C. VALIDATION OF RESULTS*

Confusion Matrix:

| | |
|---|---|
| **True Positive Rate** | 1.0 |
| **True Negative Rate** | 0.94 |
| **False Positive Rate** | 0.0 |
| **False Negative Rate** | 0.05 |

Comparison of different classification techniques with our model:



The chart above proves that our GRBM model can learn efficiently and can be taken into serious consideration for binary classification tasks involving gene expression data. A decision tree[9] that does not deal with Euclidean distances produces a classification accuracy of 94.55% while GRBM produced an accuracy of 98.62%. k-NN[9] on the other hand overfit the data and produced an accuracy of 100% which, even though ideal, is not possible in real life data.

## IV. CONCLUSION

In this work, the introduction takes us through a brief prologue to what the entire report is all about and talks about the project in bits and pieces, thus giving the reader a certain idea about the content of this compilation. This is followed by an entire detailed explanation of the process with the help of diagrams, flowcharts and formulae to describe each step in the process. The obstacles[18] faced are also described in this chapter along with the methods[19] adopted to tackle it. The activation function adopted to activate the hidden nodes is described in detail and the difference in methods from the usual Restricted Boltzmann Machine is acutely explained. The results which we are left with are compared in the next chapter depending on the different parameters used and the classification accuracy is compared with a k-NN classifier and a decision tree classifier.

However, there are certain limitations in this project which have not been addressed. An RBM has not been tested with multi class labels, thus gene expression data with multiple classes may or may not produce good results. Another limitation that the RBM does not take into account is the amount of synthetic data, most of which lies in an Euclidean space. Therefore classifiers that work in an Euclidean space shall tend to overfit the data, like the k-NN classifier. So, using safe-level SMOTE (or SMOTE itself) may produce biased results for a certain type of classifier.

In future the RBM can be used to classify multi class labels so that the level of progress of the disease can be mapped according to the gene expression values[20]. This would help in deciding the hazardous nature of the disease and also would help us assign a certain value to the patient’s condition, thus classifying the danger she or he is under.

## REFERENCES

[1] World Cancer Report 2014. World Health Organization. 2014. pp. Chapter 5.1. ISBN 9283204298.

[2] Domingos, P.: Metacost: A General Method for Making Classifiers Cost-sensitive. In: The 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 1999), pp. 155–164. ACM Press, San Diego (1999)

[3] Kamber, M., Han, J.: Data mining: Concepts and Techniques, 2nd edn., pp. 279–327. Morgan-Kaufman, NY, USA (2000)

[4] Melchior J, Wang N, Wiskott L (2017) Correction: Gaussian-binary restricted Boltzmann machines for modeling natural image statistics. PLOS ONE 12(3): e0174289. https://doi.org/10.1371/journal.pone.0174289

[5] Sutskever, Ilya; Tieleman, Tijmen (2010). "On the convergence properties of contrastivedivergence" . *Proc. 13th Int'l Conf. on AI and Statistics (AISTATS).*

[6] Ho, Tin Kam (1995). *Random Decision Forests*Proceedings of the 3rd International Conference on Document Analysis and Recognition, Montreal, QC, 14–16 August 1995. pp. 278–282

[7] Chawla, N., Japkowicz, N., Kolcz, A.: Editorial: Special Issue on Learning from Imbalanced Data Sets. SIGKDD Explorations 6(1), 1–6 (2004)

[8] ChumpholBunkhumpornpat, Krung Sinapiromsaran, ChidchanokLursinsap (2009) “Safe-Level-SMOTE: Safe-Level-Synthetic Minority Over-Sampling TEchnique for Handling the Class Imbalanced Problem” P478-479

[9] R. Quinlan, "Learning efficient classification procedures", *MachineLearning: an artificialintelligence approach*, Michalski, Carbonell& Mitchell (eds.), Morgan Kaufmann, 1983, p. 463-482 doi:10.1007/978-3-662-12405-5_15

[10] S. Tsakalidis, V. Doumpiotis& W. Byrne, "Discriminative Linear Transforms for Feature Normalization and Speaker Adaptation in HMM Estimation", Proc. ICSLP'02, Denver.

[11] Bin Mohamad, Ismail; DaudaUsman (2013). "Standardization and Its Effects on K-Means Clustering Algorithm" (PDF). Research Journal of Applied Sciences, Engineering and Technology.

[12] Chawla, Nitesh V. (2010) Data Mining for Imbalanced Datasets: An Overview doi:10.1007/978-0-387-09823-4_45 In: Maimon, Oded; Rokach, Lior (Eds) Data Mining and Knowledge Discovery Handbook, Springer ISBN 978-0-387-09823-4 (pages 875-886)

[13] Rahman,M.M. Davis,D.N. (2010) Addressing the Class Imbalance Problem in Medical Datasets, International Journal of Machine Learning and Computing vol. 3, no. 2, pp. 224–228, 2013.

[14] Chawla, N., Bowyer, K., Hall, L., Kegelmeyer, W.: SMOTE: Synthetic Minority OverSampling Technique. Journal of Artificial Intelligence Research 16, 321–357 (2002)

[15] Kohavi, Ron (1995). "A study of cross-validation and bootstrap for accuracy estimation and model selection". Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence. San Mateo, CA: Morgan Kaufmann. 2 (12): 1137–1143.

[16] Geoffrey Hinton (2010). A Practical Guide to Training Restricted Boltzmann Machines. UTML TR 2010–003, University of Toronto.

[17] RuslanSalakhutdinov and Geoffrey Hinton (2010). Replicated softmax: an undirected topic model. Neural Information Processing Systems 23.

[18] Japkowicz, N.: The Class Imbalance Problem: Significance and Strategies. In: the 2000 International Conference on Artificial Intelligence (IC-AI 2000), Las Vegas, NV, USA, pp. 111–117 (2000)

[19] Kubat, M., Matwin, S.: Addressing the Curse of Imbalanced Training Sets: One-Sided Selection. In: The 14th International Conference on Machine Learning (ICML 1997), pp. 179–186. Morgan Kaufmann, Nashville (1997)

Shi, T., Seligson D., Belldegrun AS., Palotie A, Horvath, S. (2005). "Tumor classification bytissue microarray profiling: random forest clustering applied to renal cell carcinoma". *ModernPathology*. 18 (4): 547–557. doi:10.1038/modpathol.3800322