Vol 8. No. 3 Issue 2 – October, 2015
African Journal of Computing & ICT

© 2015 Afr J Comp & ICT – All Rights Reserved - ISSN 2006-1781
www.ajocict.net

Reverse Probability Weight (RPW): An Optimization Technique for k-Nearest Neighbours Algorithm for Imbalanced Dataset

R.S. Ogunakin & E. Fubara
Department of Computer Science
University of Port Harcourt
Choba, Port Harcourt, Nigeria.
Emails: rotimi.ogunsakin@uniport.edu.ng, fubara.egbono@uniport.edu.ng

ABSTRACT

K-Nearest Neighbors Classifier experiences performance drawback when dealing with imbalanced dataset due to the majority vote technique used for classification. This research work examined and analyzed the effect of imbalanced dataset on k-NN, proposed and implemented a Reverse Probability Weight (RPW) technique as an optimization technique for dealing with imbalanced dataset in k-NN. All implementations and experiments were done using MATLAB and the result of optimized k-NN using Reverse Probability Weight (RPW) technique compared with k-NN, Logistic Regression and Support Vector Machine (SVM) shows that (1) The Reverse Probability Weight (RPW) optimizes k-NN in the presence of imbalance dataset and behave exactly as k-NN in the presence of balanced dataset (2) The Reverse Probability Weight (RPW) Technique for k-NN Outperforms k-NN, Logistic Regression and Support Vector Machine (SVM) in the presence of imbalanced dataset.

Keywords- k-Nearest Neighbors, Reverse Probability Weight (RPW), Optimization Technique

African Journal of Computing & ICT Reference Format:
R.S. Ogunakin & E. Fubara (2015): Reverse Probability Weight (RPW): An Optimization Technique for k-Nearest Neighbours Algorithm for Imbalanced Dataset. Afri J Comp & ICTs Vol 8, No.3 Issue 2 Pp 141-152

1. INTRODUCTION

K-Nearest Neighbour is a non parametric method utilized for classification and regression, a type of instance based or lazy learning algorithm where the function is only approximated locally and all computation is differed until classification [1][2]. K-NN is arguably one of the simplest and yet effective classification algorithms in the domain of Machine Learning (ML) [3]. When dealing with imbalanced dataset, k-NN algorithm tends to have a performance drawback due to the suboptimal classification of the minority class in the dataset as a result of the majority vote technique used in classification. There have been several recommended approaches to dealing with imbalance datasets in k-NN where some focused on the “data space" [4] while others focused only on “feature space”[5].

The data space approach is based on sampling strategies, oversampling the minority class and under sampling the majority class. This approach is prone to over fitting because the feature space is not taken into consideration, as a result, the over sampling features clusters around the minority class in the feature space or the under sampling removing important samples [4]. The SMOTE approach and its optimized variants is the most popular in the “feature space” approach, where over-sampling data are chosen based on the minority class distribution in the feature space to avoid over fitting and under fitting [5].

Over the years, as the ever-evolving areas of technology application increases, data sizes also increases both in volume, variety and dimension. Classification of data becomes more difficult due to unboundedness of data size and the resulting imbalanced nature of datasets. Most of the data samples used in classification are naturally imbalanced, a good example is credit card fraud in which majority of the recorded transaction are non-fraudulent. The breast cancer data used in this research is also a very good example of such natural occurrence, where majority of the classes are negative i.e. “patient do not have breast cancer”.

Majority of the proposed optimization techniques to dealing with class imbalance dataset in k-NN either considers only the “data space” or the “feature space”. The proposed Reverse Probability Weight (RPW) technique for dealing with class imbalance dataset in k-NN considers both the class distribution in the “data space” and “feature space”. The result of experiment shows that the Reverse Probability Weight (RPW) technique for k-NN is optimized for dealing with imbalanced dataset compared to Support Vector Machine (SVM) and Logistic Regression.

2. REVIEW OF RELATED WORK

K-Nearest Neighbor Optimization

Due to the simplicity and efficiency of the k-NN Algorithm, it has attracted attention in the Machine Learning domain. Several optimization techniques has been proposed to compensate for the computational draw back as a result of neighbor search and a more efficient algorithm to selecting optimal neighbor.

The Weighted Voting Technique as an alternative to the Majority Voting Technique in k-NN is an optimization technique to dealing with imbalanced dataset, where the distance between the sample to be classified and its k-Nearest Neighbors are used as a weighted-measure for voting decision rather than the class majority [6].

K-NN suffers from the problem of high variance in the case of limited sampling, however, SVM does not suffer same but SVM involves time-consuming optimization and computation of parametric distance and thus a hybrid of SVM and k-NN is proposed in this case [7]. The use of Outlier Detection Using Indegree Number (ODIN) Algorithm that utilizes k-NN graph is another improvement on the k-NN algorithm but the performance is only benchmarked using small number of observations and its performance on huge dataset is not explicit in the literature [8].

Fuzzy Sets Theory was introduced in k-NN as Fuzzy k-Nearest Neighbor Algorithm as an optimization technique to address the resulting difficulties in utilizing k-NN technique in pattern recognition - where instead of each labeled samples given equal importance in determining class membership of patterns to be classified regardless of the typicalness, a fuzzy optimization is introduced to fuzzify the typicalness of each labeled samples [9].

Dempster-Shafer theory is used to address the problem of classifying an unseen pattern on the basis of its nearest neighbors in pattern recognition; where the degree of support of the membership of a pattern is defined as a distance function between the sample to be classified and its nearest neighbor and the resulting k-Nearest Neighbors is pooled by the means of Dempster-Shafer rule [10]. The branch and bond algorithm is an efficient algorithm used to reduce the number of neighbor search in large datasets, it facilitate rapid computation of the k-Nearest Neighbor by totally eliminating the need for computation of many distances such that only 60 neighbors-distance computation out of 1000 samples suffices to gives optimal classification [11].

Multi-step query processing strategy in k-NN as a nearest neighbor search algorithm is used to address the efficiency requirements of high-dimensional and adaptable distance function, which occur as a result of increasing databases application, and complexity of objects such as images and videos in multimedia databases [12]. A new version of Approximating and Eliminating Search Algorithm (AESA) is used for finding nearest neighbor in metric space where distance computation is highly expensive. This algorithm only requires an inexpensive linear-programming approach instead of the usual AESA high-memory-demanding and computationally expensive quadratic programming [13].

K-Nearest Neighbor Optimization For Imbalanced Dataset

The above different optimization techniques to k-NN algorithms does not solve the class imbalanced problem in kNN. For example, in an imbalanced data set, if a sample is to be classified at the boundary region in the feature space, all selected k-Nearest Neighbors are guaranteed to be at approximate equal distance, the class proportion $pt(c_1)$ for the minority class and $pt(c_2)$ for the majority class in the feature space $\Phi(xc)$ is such that $pt(c_2) \gg pt(c_1)$ at the boundary and thus the distance weighted k-Nearest Neighbor technique and its optimized variant will not be applicable [14].

There are different techniques available for classification of imbalanced data sets, which can be summarized as follows: [15]:

1. Data preprocessing approach: Over and under sampling of data in “data space” and “feature space”
2. Algorithmic approach: Applying cost in making decision
3. Feature selection approach: Dimensionality reduction

The over and under sampling strategies has attracted several attention with conflicting viewpoints on usefulness. The random over and under sampling have their short comings - the random under-sampling technique can potentially remove important samples from the datasets while the random over-sampling can lead to over-fitting by oversampling data points clustering around the minority data samples [16]. Several techniques have been proposed to tackle these shortcomings in data preprocessing approach to classification of imbalanced dataset, such as the use of one sided selection to selectively under-sample the original population [17]. The use of Condensed Nearest Neighbor (CNN) rule to remove examples from the majority class that are far away from the decision boundary [18]. The Neighborhood Cleaning Rule (NCR) to remove the majority class samples using the NCR technique [19].

The SMOTE (Synthetic Minority Oversampling TEchnique) is a technique used to generate synthetic examples by operating in the “feature space” rather than the “data space” where the minority class is oversampled by taking each minority class samples and introducing synthetic examples along the line segment joining any/all the k minority class nearest neighbors using the SMOTE Algorithm [20]. An improvement of the SMOTE Algorithm is the incorporation of Locally Linear Embedding (LLE) Algorithm. The LLE algorithm is first applied by mapping the high-dimensional data into a low-dimensional space (Dimensionality reduction) where the input data is separable and thus oversampled using the SMOTE algorithm, the resulting generated synthetic data points by SMOTE are mapped back to the original high-dimensional input space through the LLE Algorithm [21].

Vol 8. No. 3 Issue 2 – October, 2015
African Journal of Computing & ICT
© 2015 Afr J Comp & ICT – All Rights Reserved - ISSN 2006-1781
www.ajocict.net

The Fuzzy-rough k-Nearest Neighbor Algorithm is an algorithmic approach for dealing with imbalanced dataset in k-NN to eliminate the bias of traditional method to the majority class by producing poor detection rate of the minority class - this approach takes into consideration the fuzziness and roughness of the nearest neighbors before making classification decision [22].

The Reverse Probability Weight (RPW) technique is an algorithmic technique for dealing with imbalanced dataset in k-NN. A very simple but highly effective algorithm for dealing with problem of imbalance dataset in k-NN.

K-nn Optimization Technique

Reverse Probability Weight (RPW) Technique

The Reverse Probability Weight (RPW) technique is a technique whereby the independent prior probability of a minority and a majority sample being selected in the data space is computed and project into the feature space, with the assumption that the prior probability of a sample being selected in the "data space" is the same as the probability of the same sample being selected in the "feature space".

For example, an imbalanced dataset of 100 samples *S* with 80 majority class c_0 of class label "0" and 20 minority class c_1 of class label "1" as shown in Fig.1. The probability of a minority class c_1 and majority class c_0 being selected in the data space D_s and feature space F_s can be computed as follows:

a) $P(c_0|D_s) = \frac{\sum c_0}{\sum(c_0+c_1)} = 0.8$

b) $P(c_1|D_s) = \frac{\sum c_1}{\sum(c_0+c_1)} = 0.2$

Assume the same probability holds in the feature space:

$$P(c_0|D_s) = \frac{\sum c_0}{\sum(c_0+c_1)} \Rightarrow P(c_0|F_s) = \frac{\sum kc_0}{\sum(kc_0+kc_1)}$$

$$P(c_1|D_s) = \frac{\sum c_1}{\sum(c_0+c_1)} \Rightarrow P(c_1|F_s) = \frac{\sum kc_1}{\sum(kc_0+kc_1)}$$

Where kc_0 and kc_1 are c_0 and c_1 nearest neighbours in the feature space.

The Reverse Probability Weight (RPW) is estimated by reversing the probability in the feature space and this can be refers to as "weight of fairness". This is an intuitive way of compensating for the sparsity of the minority class in the data space in feature space during classification.

$$RPW\, c_0 = 1 - \frac{\sum c_0}{\sum(c_0+c_1)} \Rightarrow 1 - \frac{\sum kc_0}{\sum(kc_0+kc_1)} \qquad \text{III.1}$$

$$RPW\, c_1 = 1 - \frac{\sum c_1}{\sum(c_0+c_1)} \Rightarrow 1 - \frac{\sum kc_1}{\sum(kc_0+kc_1)} \qquad \text{III.2}$$

The RPW c_0 and RPW c_1 will be used to weigh each c_0 and c_1 nearest Neighbors kc_0 and kc_1 before class assignment and thus:

a)
$If\ RPW\, c_0 * kc_0 > RPW\, c_1 * kc_1;\ Then\ Sample\ S \Rightarrow\ c_0$

b)
$If\ RPW\, c_0 * kc_0 < RPW\, c_1 * kc_1;\ Then\ Sample\ S \Rightarrow\ c_1$

c)
$f\ RPW\, c_0 * kc_0 = RPW\, c_1 * kc_1;\ Then\ Sample\ S \Rightarrow\ c_1$

$(Giving\ priority\ to\ the\ minority\ class)$

In Figure 2.1 below, k-NN classification with k = 9 as shown, using the majority vote will misclassify the sample as belonging to the majority class "0" instead of the minority class "1".

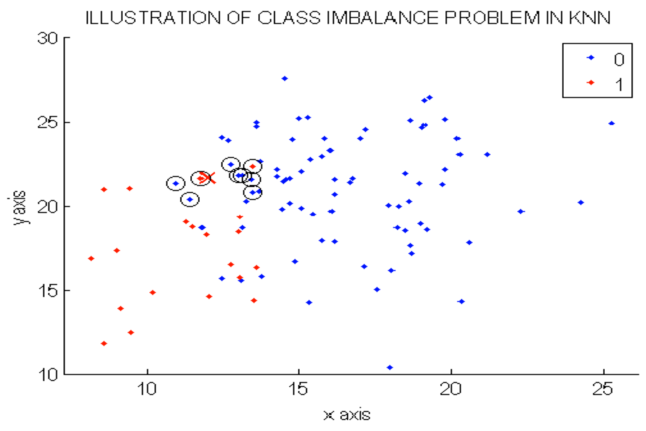


Figure III.1: Illustration Of Class Imbalanced Dataset in k-NN

Applying the RPW technique such that:

1) $P(c_0|D_s) = \frac{\sum c_0}{\sum(c_0+c_1)} = 0.8$

2) $P(c_1|D_s) = \frac{\sum c_1}{\sum(c_0+c_1)} = 0.2$

This implies:

a) $RPW\, c_0 = 1 - 0.8 = 0.2\ from\ equation\ (3.1)$

b) $RPW\, c_1 = 1 - 0.2 = 0.8\ from\ equation\ (3.2)$

$kc_0 = 7$ and $kc_1 = 2$ as shown in Figure 2.1 above.

Applying the RPW we obtain:

4) $RPW\, c_0 * kc_0 = 0.2 * 7 = 1.4$

2) $RPW\, c_1 * kc_1 = 0.8 * 2 = 1.6$

$RPW\, c_0 * kc_0 < RPW\, c_1 * kc_1;\ Then\ Sample\ S \Rightarrow c_1$

With the RPW technique the sample is thus classified correctly as belonging to the minority class c_1. The Optimized k-NN is able to deal effectively with imbalanced dataset using the reverse probability weight technique. The performance of RPW technique is first benchmarked with k-NN using three datasets prepared by increasing the minority class to analyse the sensitivity of the algorithm to the degree of imbalanceness in a dataset. The result of the experiment shows that RPW is a highly effective technique for dealing with imbalanced dataset in k-NN. The RPW Algorithm is presented below.

RPW Algorithm (Optimization of k-NN Algorithm Using RPW Technique)

```
1: Input: Data space $D_s$, Random number $k$,
   Sample $c$; Output: Class of sample $c$

2: Compute $P(c_1|D_s)$ and $P(c_0|D_s)$

    2a: Compute $RPW c_1 = 1 - P(c_1|D_s)$
    2b: Compute $RPW c_0 = 1 - P(c_0|D_s)$

3: Seek $kc_1$ and $kc_0$ in $F_s$ using Euclidean
   Distance

4: If $RPW c_0 * kc_0 > RPW c_1 * kc_1$ Then $c = c_0$

   ElseIf $RPW c_1 * kc_1 > RPW c_0 * kc_0$ Then $c = c_1$

   ElseIf $(RPW c_1 * kc_1 = RPW c_0 * kc_0)$ &&
  $(P(c_1|D_s) < P(c_0|D_s))$ Then $c = c_1$

   Else $c = c_0$
```

Performance Measure (f-value, ROC and Box-plot)
The overrepresentation of the negative class in the imbalanced dataset poses problems in accurately evaluating the performances of the classifiers, however, since error rate may not be a very good metric for skewed datasets, the classification performance of algorithms in an imbalanced dataset is measured by precision and recall [20].

$$Precision = \frac{TP}{(TP + FP)} \qquad (3)$$

$$Recall = \frac{TP}{(TP + FN)} \qquad (4)$$

Where *TP; TN; FP and FN* represents True Positive, True Negative, False Positive, and False Negative respectively.

The accuracy of a classifier is also important, it shows the overall performance of the classifier in correctly classifying both the positive and the negative classes.

$$Accuracy = \frac{(TP + TN)}{(TP + FP + TN + FN)} \qquad (5)$$

The main goal of learning from imbalanced datasets is mainly to improve the recall without affecting the precision, but since there is a trade-off between the Precision and Recall, we use the f-value which combines the trade-offs and output a single value that reflect the "goodness" of the classifier in the presence of imbalanced dataset [2].

$$F - value = \frac{(1 + \beta^2) * Recall * Precision}{\beta^2 * Recall + Precision} \qquad (6)$$

The box-plot is used as a tool to visualize and measure the performance of the Optimized k-NN when benchmarked with k-NN, Logistic Regression and Support Vector Machine (SVM)

3. EXPERIMENTS

All experiments are performed on MATLAB using a "breast cancer diagnostic" dataset with 30 features and two class labels. Class "0" is the majority class (The negative class) which implies that patient have no cancer of the breast and "1" which is the minority class (The positive class) and implies that patient have cancer of the breast. The breast dataset is divided into four categories for training and testing as follows:

Table 1: Breast Cancer Dataset

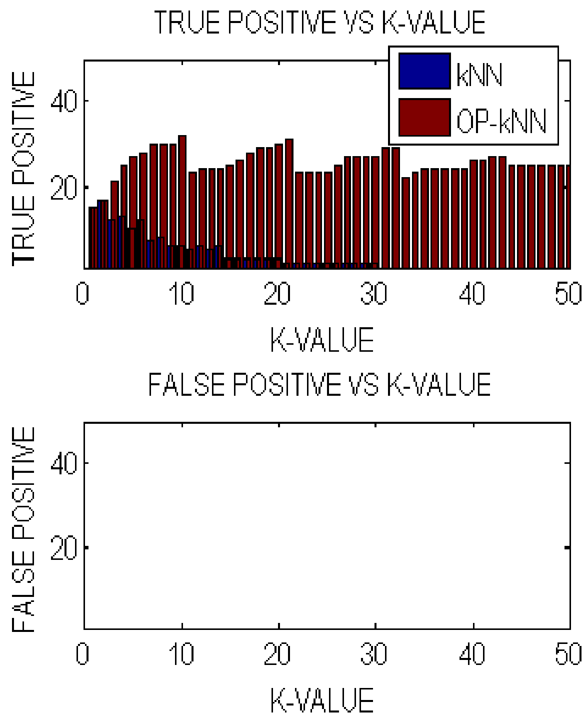
Dataset	Features	Total Samples	Class Zero	Class One
Imbalanced	30	200	180	20
Slightly Imbalanced	30	200	150	50
Balanced	30	200	100	100
Test	30	100	50	50

A balanced dataset of 100 samples (50 for class zero "0" and 50 for class one "1") is used as test dataset to be able to visualize how the different classifiers classify the minority class and majority class in the presence of imbalanced dataset. We did not use cross validation so we can be able to monitor the test data and keep track of the True Positive (TP) and False Negative (FN) of the minority data class. The Three training datasets are divided into "Imbalanced", "Slightly imbalanced" and "Balanced" such that we will be able to monitor the performance of the optimized k-NN on different datasets to affirm that while optimizing for imbalance dataset we are not recording any compromise on the balanced datasets.

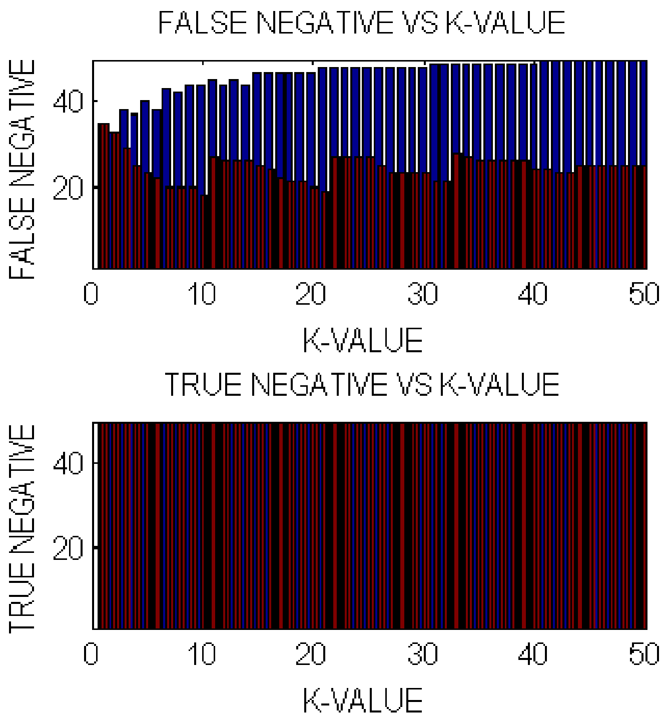
The sensitivity of the optimized k-NN algorithm to imbalanced dataset is observed as we move from imbalanced to slightly imbalanced and to balanced dataset.

Imbalanced Dataset (k-NN and Optimized k-NN using RPW Algorithm)

The confusion matrix obtained from the first experiment as shown in Figure 4.1(a) shows that the Optimized k-NN has higher True Positive (TP) while the False Positive (FP) remain constant as we varied k from 1 to 50. While the k-NN generally have a very low TP rate and the TP rate decreases as k increases which is as a result of the biasness of the k-NN algorithm to the minority class. This shows an improvement on the classifiers precision in classifying the minority class in the presence of an imbalanced dataset as compared to k-NN.



(a) True Positive and False Positive Vs K-Value

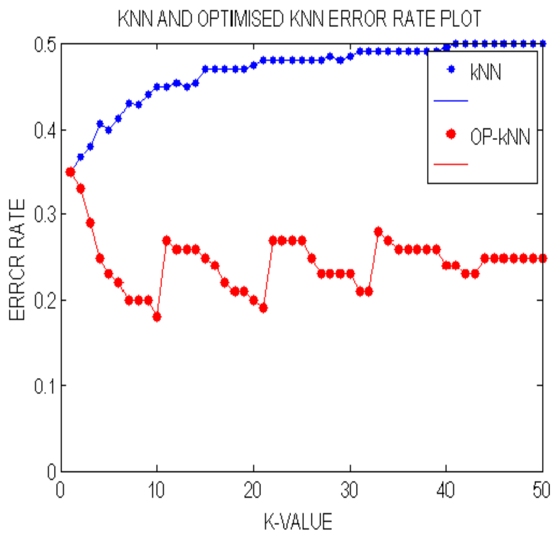


(b) False Negative and True Negative Vs K-Value

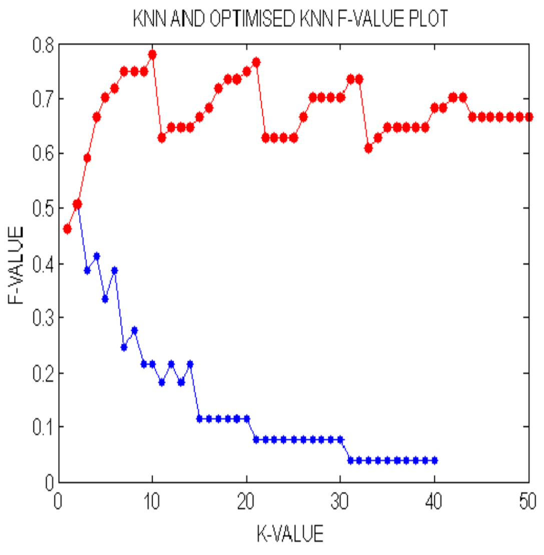
Figure 4.1: Imbalanced Dataset (k-NN Vs Optimized k-NN)

Figure 4.1(b) shows that the FN for the Optimized k-NN is also lower with lower variance, which shows an improvement in the recall, minimizing the number of incorrectly classified minority class while the FN of k-NN increases and TP decreases as we increase the value of k. TN remains constant for both k-NN and optimized k-NN as we increase the value of k. This is expected, because the negative class (class “0”) is the majority class and thus, both classifiers have no bias on the majority class.

Figure 4.2 shows that the f-value of the Optimized k-NN is remarkably higher and recorded a lower error rate with less variance compare to the f-value and the error rate of k-NN Algorithm.

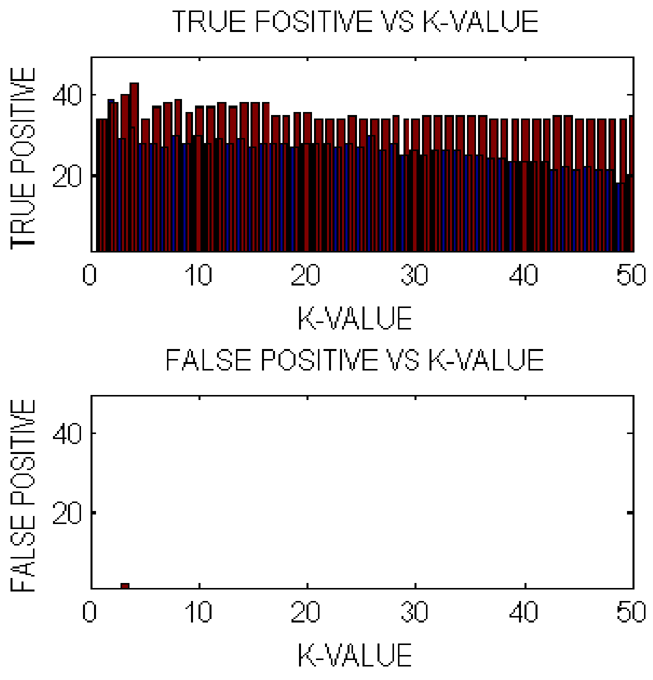


(a) k-NN and Optimised k-NN Error Rate

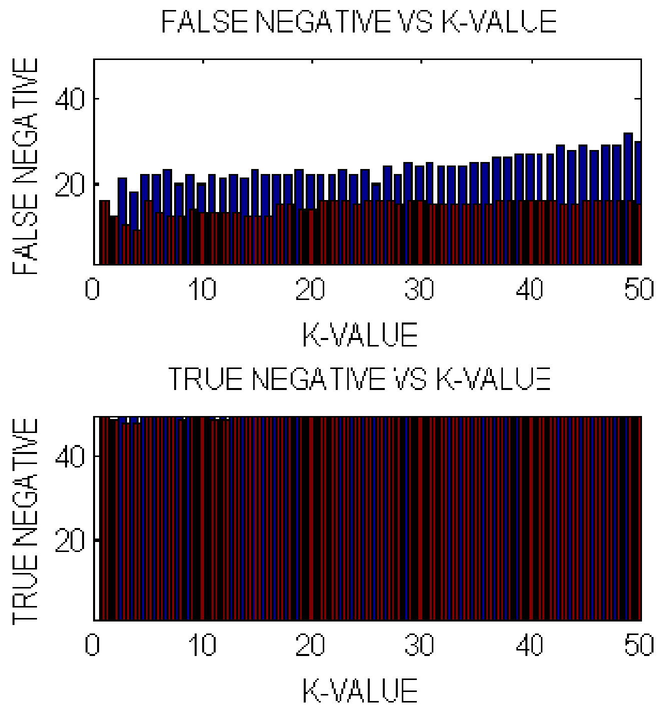


(b) k-NN And Optimized k-NN F-value plot
Figure 4.2: Error Rate and f-Value for k-NN and Optimized k-NN

Slightly Imbalanced Dataset (k-NN and Optimized k-NN using RPW Algorithm)
Both k-NN and the Optimized k-NN recorded an improvement on both precision and recall as we reduce the degree of imbalanceness in the training dataset as shown in Figure 4.3(a) and (b) below. But overall, the RPWk-NN outperforms k-NN.



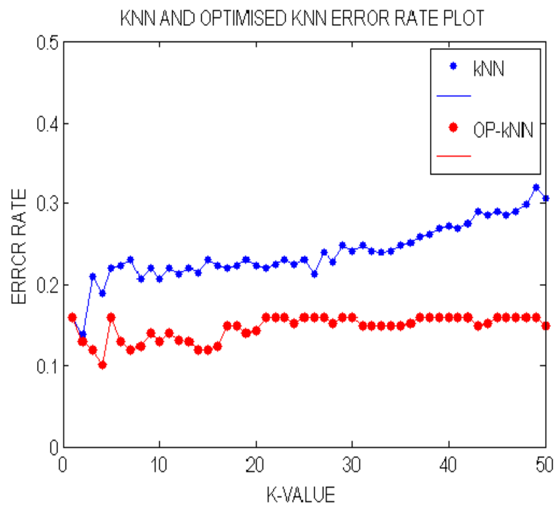
(a) True Positive and False Positive Vs K-Value



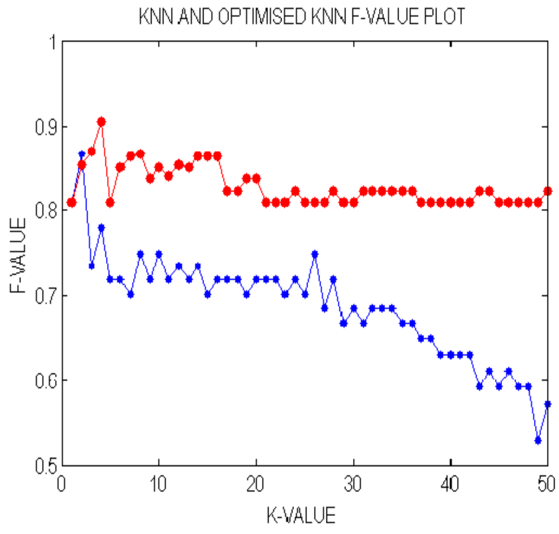
(b) False Negative and True Negative Vs K-Value
Figure 4.3: Slightly Imbalanced Dataset (k-NN Vs Optimized k-NN)

Vol 8. No. 3 Issue 2 – October, 2015
African Journal of Computing & ICT
© 2015 Afr J Comp & ICT – All Rights Reserved - ISSN 2006-1781
www.ajocict.net

Figure 4.4 shows that the f-value of the k-NN and optimized k-NN are higher than the one obtained from the imbalanced dataset confirming the sensitivity of the classifiers to data imbalanceness. Overall, the f-value of the Optimized k-NN is remarkably higher than that of k-NN and recorded a lower error rate with less variance compare to the error rate of k-NN Algorithm.



(a) k-NN And Optimized k-NN Error Rate

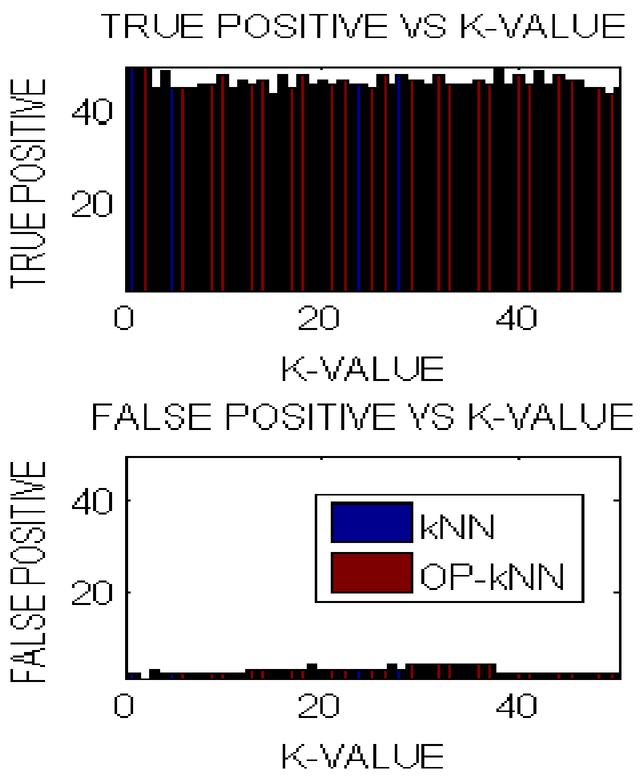


(b) k-NN And Optimized k-NN F-value plot

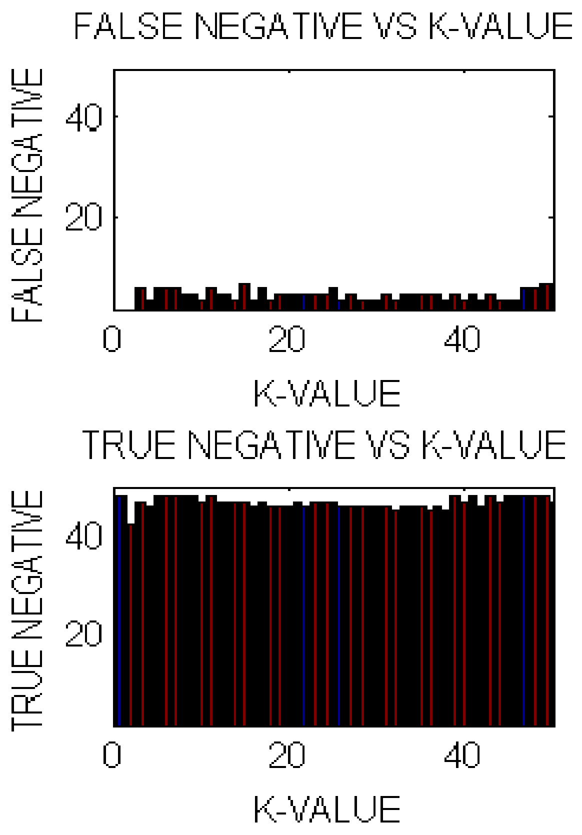
Figure 4.4: Error Rate and f-Value for k-NN and Optimized k-NN

Balanced Dataset (k-NN and Optimized k-NN using RPW)

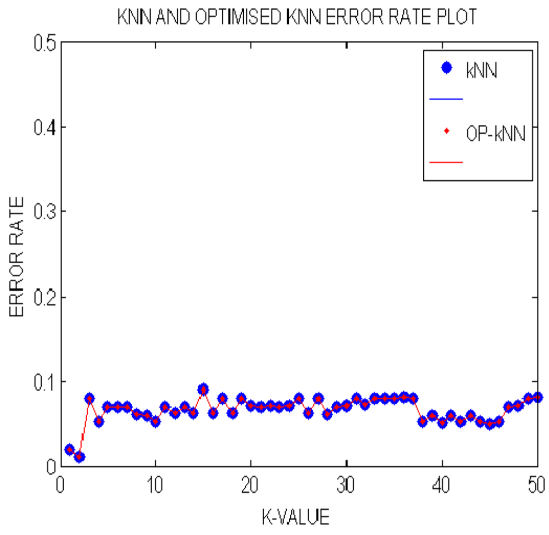
As seen from Figure 4.5(a), the result of both k-NN and Optimized k-NN are exactly the same which shows that the Reverse Probability Weight (RPW) Algorithm only optimizes k-NN in the presence of imbalanced dataset but behaves exactly as k-NN when the dataset is balanced. Figure 4.6(b) also shows that the same f-value and error rate were obtained for both k-NN and optimised k-NN in the presence of a balanced dataset.



(a) True Positive and False Positive Vs K-Value

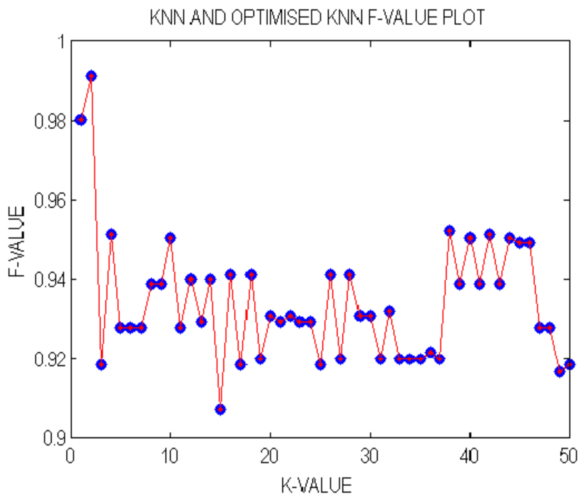


(b) False Negative and True Negative Vs K-Value
Figure 4.5: Balanced Dataset (k-NN Vs Optimized k-NN)



(a)

k-NN And Optimized k-NN Error Rate

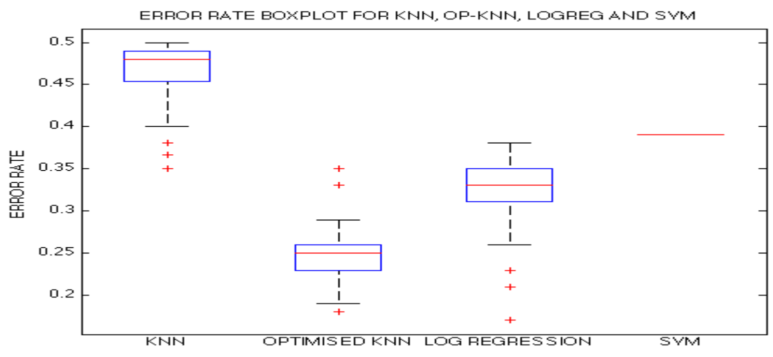


(b) k-NN And Optimized k-NN F-value plot
Figure 4.6: Error Rate and f-Value for k-NN and Optimized k-NN

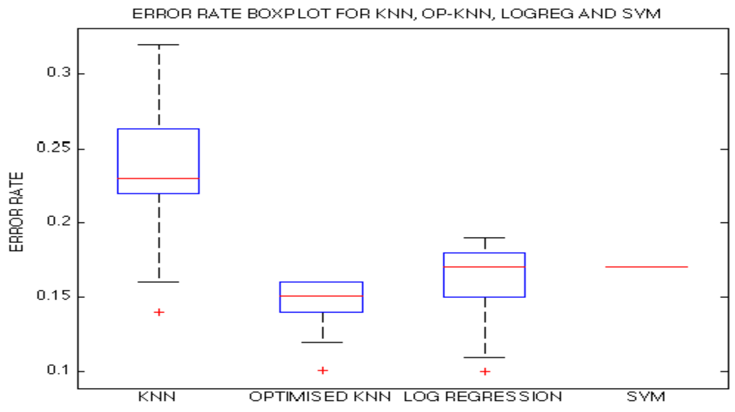
Vol 8. No. 3 Issue 2 – October, 2015
African Journal of Computing & ICT

The four classifiers were tested on the Three datasets and the results were analyzed using box-plot as shown in Figure 4.7(a), (b) and (c) below. We varied k from 1 to 50 for both k-NN and optimized k-NN. For logistic regression and SVM we varied the number of iterations from 1 to 50 and compute the Average Error Rate. We observed that the optimized k-NN performs best on the imbalanced dataset as shown in Figure 4.7(a) with the lowest error rate, followed by Logistic Regression, SVM and finally k-NN.

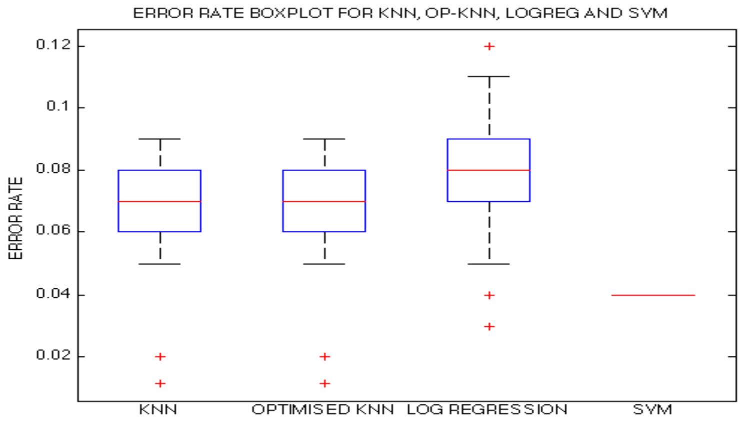
Also, the Optimized k-NN performs best on the slightly imbalanced dataset as shown in Figure 4.7(b) which shows that the Optimized k-NN using Reverse Probability Weight (RPW) technique is a better classifier to deal with imbalanced dataset compared to Logistic Regression, SVM and k-NN. The SVM performed best on the balanced dataset as expected, followed by the Optimized k-NN and k-NN.



(a) Imbalanced Dataset



(b) Slightly-Imbalanced Dataset



(c) Balanced Dataset

Figure 4.6: Error Rate Boxplot for k-NN, Optimized k-NN, Logistic Regression and SVM

© 2015 Afr J Comp & ICT – All Rights Reserved - ISSN 2006-1781
www.ajocict.net

4. ANALYSIS

The results of the experiments show that using Reverse Probability Weight (RPW) Algorithm optimizes k-NN in the presence of imbalanced dataset. The optimization is majorly visible in the increased True Positive (TP) Rate by correctly classifying the minority class and reduction in the False Negative (FN) rate by reducing the number of misclassified minority class - this is the major goal of the optimization. The RPW for the minority class is usually higher than the RPW for the majority class with a factor of their respective prior probabilities.

Analysis of the imbalanced dataset used shows that for a sample to be classified as belonging to the majority class, the number of the majority class kc_0 nearest Neighbors must be at least 9 times the number of the minority class kc_1 as shown below:

The total number of samples in the data space D_s = 200 and the number of positive minority class is 20 while the majority class has 180.

$$RPWc_0 = 1 - \frac{\sum c_0}{\sum(c_0 + c_1)} \Rightarrow 1 - \frac{\sum kc_0}{\sum(kc_0 + kc_1)} = 1 - 0.1 = 0.9$$

$$RPWc_1 = 1 - \frac{\sum c_1}{\sum(c_0 + c_1)} \Rightarrow 1 - \frac{\sum kc_1}{\sum(kc_0 + kc_1)} = 1 - 0.9 = 0.1$$

The ratio between sample kc_1 and sample $kc_0 = \frac{0.9}{0.1} = 9$

This means for any value of k in the imbalanced dataset, for a sample c to be classified as c_0 there must be at least Nine (9) times the number of kc_0 as there are kc_1 in the k nearest Neighbors. This account for the reason why increase in k-values lead to increase in True Positive (TP) because for every sample of the minority class c_1 in the k-nearest Neighbors there must exist a minimum of 9 majority class c_0 and this becomes more difficult to arrive at as we increase k, since the probability of having c_1 in the k-nearest Neighbors increases as we increase k and the probability of having 9-times kc_0 reduces as we increase k. If k is very large, there is a higher chance of classifying correctly the minority class since k-nearest neighbors will extend beyond the boundary in both directions.

A better performance is observed in both k-NN and the Optimized k-NN when we reduce the level of “imbalanceness” in the dataset because the Reverse Probability Weight (RPW) has a linear relation with the degree of imbalance in the dataset – $RPWc_1$ decreases as the number of sample c_1 increases and $RPWc_0$ increases as the number of sample c_0 decrease thus converging the RPW at the optimal.

$$RPW_{optimum} = \left[\frac{\sum kc_0}{\sum(kc_0 + kc_1)} + \frac{\sum kc_0}{\sum(kc_0 + kc_1)}\right] * 1/2$$

The performance of k-NN and Optimized k-NN are the same in the presence of a balanced dataset because the prior probability of both classes are the same in the data space, meaning all samples in the k-nearest Neighbors have same Reverse Probability Weight (RPW) which is equal to $RPW_{optimum}$

5. CONCLUSIONS

We have been able to show that the Reverse Probability Weight (RPW) Algorithm optimizes k-NN in the presence of imbalance dataset and also behave exactly as k-NN in the presence of balanced dataset. We have also been able to justify empirically through experiment that the prior probability of selecting a sample in the data space D_s is approximately equal to the probability of selecting the same sample in the feature space F_s.

$$P(c_0|D_s) = P(c_0|F_s) \; and \; P(c_1|D_s) = P(c_1|F_s)$$

In the experiment, we optimized k-NN based on majority vote technique, another approach would be to optimize based on weighted voting and compare the results. Since we observed increased performance in the slightly imbalanced dataset, another approach would be to use the **SMOTE** algorithm to reduce the level of imbalanceness in the dataset before applying RPW Algorithm and compare the result. The performance of $RPWk - NN$ (Reverse Probability Weight k-NN) could also be benchmarked with other novel techniques of dealing with imbalanced datasets in both k-NN and other classification algorithms.

REFERENCES

[1] Altman, N. S. (1992). "An introduction to kernel and nearest-neighbor nonparametric regression". The American Statistician, vol. 46 pp.175–185.

[2] Coomans. D, Massart D.L. (1982). "Alternative k-nearest neighbor rules in supervised pattern recognition". Analytica Chimica Acta, vol. 136 pp. 15-27.

[3] Zhenghui, M., and Ata, K. K-Nearest-Neighbors with a Novel Similarity Measure for Intrusion Detection [Online]. Available: http://www.cs.bham.ac.uk/~axk/ Zenghui_ukci13.pdf

[4] Nitesh, V.C. Data Mining for Imbalanced Dataset: An Overview [Online]. Available: https://www3.nd.edu/~dial /papers/SPRINGER05.pdf

[5] Nitesh, V.C., Kevin, W.B, Lawrence, O.H., and Philip, K. W. SMOTE: Synthetic Minority Over-sampling Technique [Online]. Available: https://www.jair.org/ media/953/live-953-2037-jair.pdf

[6] Jianping, G., Lan, D., Yuhong, Z. and Taisong, X. A New Distance-weighted k-nearest Neighbor Classifier. Available from: http://www.joics.com/publishedpapers/ 2012_9_6_ 1429_1436.pdf

[7] Hao, Z., Berg, A.C., Maire, M., Malik, J., "SVM-KNN: Discriminative Nearest Neighbor Classification for Visual Category Recognition," presented at Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference. Vol. 2 p. 2126-2136

[8] Hautamaki, V., Karkkainen, I., and Franti, P., ?Outlier detection using k-nearest neighbor graph,? *Proceedings of the 17th International Conference on Pattern Recognition*, *2004*. Vol.3 pp. 430?433.

[9] Keller, J. M., Gray, M. R., and Givens, J. A. (1985) "A fuzzy K-nearest neighbor algorithm," IEEE Trans. Syst. Man. Cybern., vol. 4 pp. 580?585.

[10] Denoeux, T. (1995) "A k-nearest neighbor classification rule based on Dempster-Shafer theory," IEEE Trans. Syst. Man. Cybern., vol. 25 pp. 804-813.

[11] Fukunaga, K., and Narendra, P. M. (1975) "A Branch and Bound Algorithm for Computing k-Nearest Neighbors," IEEE Trans. Comput., vol. 7 pp. 750?753.

[12] Seidl, T., and Kriegel, H. P. (1998) "Optimal multi-step k-nearest neighbor search," ACM SIGMOD Rec., vol. 27, no. 2, pp. 154-165.

[13] Mico, M. L.,Oncina, J., and Vidal, E. (1994) "A new version of the nearest-neighbor approximating and eliminating search algorithm (AESA) with linear preprocessing time and memory requirements". Pattern Recognit. Lett., vol. 15, no. 1, pp. 9-17.

[14] Wei, L., and Sanjay, C. (2004) Class Confidence Weighted k-NN Algorithms for Imbalanced Data Sets [Online]. Available: http://sydney.edu.au/engineering /it/~weiliu/Webpage/k-NN_pakdd11.pdf

[15] Longadge, R., Dongre, S. S., and Malik, L. (2013) "Class imbalance problem in data mining: review," in Int. J. Comput. Sci. Netw., vol. 2, no. 1, pp. 83?87.

[16] Chawla, N. V. (2005) "Data Mining for Imbalanced Datasets: An Overview". Data Min. Knowl. Discov. Handb., pp. 853?867.

[17] Kubat, M. and Matwin, S., "Addressing the Curse of Imbalanced Training Sets: One Sided Selection," in Proceedings of the Fourteenth Intemational Conference on Machine Learning, 1997, pp. 179-186

[18] Hart, P. E. (1968) "The Condensed Nearest Neighbor Rule". IEEE Transactions on Information Theory. Vol 14 pp. 515-516.

[19] Laurikkala, J. (2001) "Improving Identification of Difficult Small Classes by Balancing Class Distribution". Technical Report A-2001-2, University of Tampere.

[20] Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P. (2002). "SMOTE: Synthetic Minority Oversampling TEchnique". Journal of Artijcial Intelligence Research, vol 16 pp. 321-357.

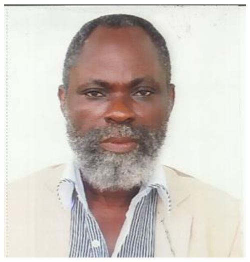
[21] Juanjuan, W., Mantao, X., Hui, W., and Jiwu, Z. "Classification of imbalanced data by using the SMOTE algorithm and locally linear embedding," in *proc,* ICSP, 2007, vol. 3 pp. 1-4.

[22] Han, H., and Mao, B. "Fuzzy-rough k-nearest neighbor algorithm for imbalanced data sets learning," in *proc* FSKD, 2010, vol. 3, pp. 1286?1290.

© 2015 Afr J Comp & ICT – All Rights Reserved - ISSN 2006-1781
www.ajocict.net

Author's Biographies

Mr. Rotimi Ogunsakin is a Lecturer at the Department of Computer Science, University of Port Harcourt. He specializes in Big Data Analytics and Financial Intelligence. He has BSc in Computing Science from the University of Port Harcourt and a Master degree in Advance Computer Science and Information Technology Management from The University of Manchester, United Kingdom. His present research interests are in Real-time Big Data Analytics and Dynamic Ecosystem Model in The Internet of Things (IoT).



Dr. Fubara Egbono is a Lecturer at the Department of Computer Science, University of Port Harcourt. He specializes on Distributed Databases and Machine Architecture. His research interest is in solving real life challenges using modern data design principles. He has a BSc and MEng in Computer Engineering at Vinnitsa Poly Technic Institute (Vinnitsa State Universit) Ukraine, USSR, Europe in 1993 and a PhD in Computer Science at Ebonyi State University in 2013. His present research interests are in Distributed Database Modeling and Optimization.