Int J Pharm Bio Sci 2015 April; 6(2): (B) 1179 – 1186

**Research Article** **Bioinformatics**

**International Journal of Pharma and Bio Sciences** **ISSN 0975-6299**

# DESIGN OF NEAREST NEIGHBORHOOD ENSEMBLE FOR ENHANCED ACCURACIES TO DIABETES DATA SET

**T.LAVANYA[1] AND A.KUMARAVEL[2]**

*[1]Research Scholar, Department of Computer Science and Engineering Bharath University, Selaiyur, Chennai- 600073, lavi.sidd@gmail.com*
*[2]Professor and Dean, School of Computing, Bharath University, Selaiyur, Chennai-600073, drkumaravel@gmail.com*

## ABSTRACT

The design of an ensemble guarantees success, only when its base classifiers make both limited errors as well as high accuracy. We address the problem of achieving the possible enhancement of accuracy of the learning models for diabetes data set.In this paper we design an ensemble by four types of Meta level classifiers for this purpose. The base classifiers for feeding the ensemble we have proposed the pool of instance based learners uniformly and established the results of experiments by various size of neighborhood, selection of samples of training set and cross validation. Hence the introduction of such experiment we show the generation of an effective and diverse ensemble of NN classifiers.

**KEYWORDS**: **Ensemble, Unsupervised Learning, Bagging, Dagging, Multi boost, Ada boost, Instance based Learners, K-Nearest Neighbor Classifier, Classification accuracy.**

*Corresponding author 

**A.KUMARAVEL**
**Professor and Dean, School of Computing, Bharath University, Selaiyur, Chennai-600073,**

# INTRODUCTION

Generating learning models to predict the chronic disease based on diabetes occurrences is challenging and important. Authors [1,2] have shown the improvement of learners for this context. In this paper we consider by extending to the design of ensemble, specifically exploring with base classifiers KNN. We emphasize the aggregation of multiple learning models with the goal of improving overall accuracy. In this paper we address the problem of achieving the possible enhancement of accuracy of the learning models for diabetes data set. In this paper, we design an ensemble by four types of Meta level classifiers for this purpose. The base classifiers for feeding the ensemble we have proposed the pool of instance based learners uniformly and established the results of experiments by various size of neighborhood, selection of samples of training set and cross validation. Most of the algebraic methods like finding simple average, weighted average, simple or weighted sum, product, maximum, minimum, median and voting are applied to combine the learning models. Even though the accuracy is expected to increase, it is more difficult to characterize and explain predictions. An ensemble of classifiers succeeds in improving the accuracy of the whole when the component classifiers are both diverse and accurate. Diversity is required to ensure that the classifiers make uncorrelated errors. If each classifier makes the same error, the voting carries that error into the decision of the ensemble, there by gaining no improvement. In addition, accuracy is required to avoid poor classifiers to obtain the majority of votes. These requirements have been classifiers. Under simple voting and error independency conditions, if all classifiers have the same probability of error, and such probability is less than 50%, then the error of the ensemble decreases monotonically with an increasing number of classifiers. One way to generate an ensemble with the required properties is to train the classifiers on different sets of data, obtained by sampling from the original training set [3,4,5]. Breiman's bagging and Freund and Schapire's boosting are well known examples of successful iterative methods for improving the predictive power of classifiers learning systems. Bagging uses sampling with replacement. It generates multiple classifiers by producing replicated samples of the data. To classify an instance a vote for each class is recorded by every classifier that chooses it, and the class with the most votes is chosen by the aggregating scheme. Boosting uses adaptive sampling. It uses all instances at each repetition, but maintains a weight for each instance in the training set that reflects its importance as a function of the errors made by previously generated hypotheses. As for bagging, boosting combines the multiple classifiers by voting, but unlike bagging boosting assigns different voting strengths to component classifiers on the basis of their accuracy. Experimental evidence [3,4,5] proved that both bagging and boosting are quite effective in reducing generalization error, with boosting providing in general higher improvements. This behavior can be explained in terms of the bias-variance components of the generalization error [6]. The variance component measures the scatter in the predictions obtained from using different training sets, each one drawn from the same distribution. The effect of combination is to reduce the variance, is what both bagging and boosting achieve. In addition, boosting does something more. By concentrating the attention of the weak learner on the harder examples, it challenges the weak learner algorithm to perform well on these harder parts of the sample space, thereby reducing the bias of the learning algorithm. To gain some insights as to why this is the case, we need to take care in the design of ensemble of such weak classifiers like K-nearest neighborhood (KNN).

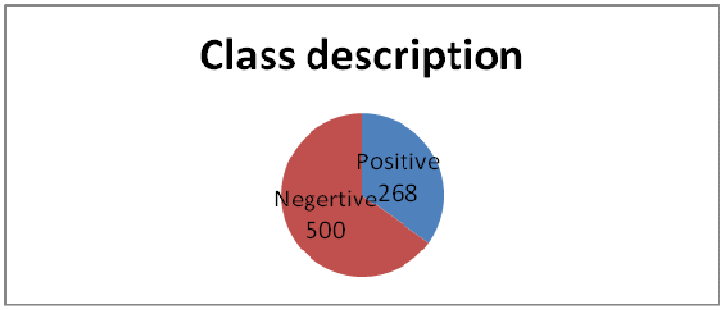
### *2. Dataset Collection and Data Preparation*

In this section, we dwell the collection of data and format in which the data has to be presented for mining experiments following the iterative steps in Figure 1. We use Java based implementation, namely Weka tool from University of Waikato, New Zealand. The

diagnostic, binary-valued variable investigated is whether the patient shows signs of diabetes according to World Health Organization criteria (i.e., if the 2 hour post-load plasma glucose was at least 200 mg/dl at any survey examination or if found during routine medical care).The population lives near Phoenix, Arizona, USA. Several constraints were placed on the selection of these instances from a larger database. In particular, all patients here are females at least 21 years old of Pima Indian heritage. The datasets for these experiments are from [15].The original data format has been slightly modified and extended in order to get relational format.

### *2.1 Dataset description*

The database of diabetes describes a set of eight attributes[10] as shown in the below list 2.2. The class attribute has binary values 'tested negative' and 'tested positive'. The number of instances in this database is 768.




### *2.2 List of attributes*

***For each attribute (all numeric-valued), and the units are shown***

| S.No | Attribute | Description with Units |
|---|---|---|
| 1. | Preg | Number of times pregnant |
| 2. | Plas | Plasma glucose concentration a 2 hours in an oral glucose tolerance |
| 3. | Pres | Diastolic blood pressure (mm Hg) |
| 4. | Skin | Triceps skin fold thickness (mm) |
| 5. | Insu | 2-Hour serum insulin (mu U/ml) |
| 6. | Mass | Body mass index (weight in kg/(height in m)^2) |
| 7. | Pedi | Diabetes pedigree function |
| 8. | Age | Age (years) |
| 9. | Class | Class variable (0 or 1) ' tested negative' or 'tested positive' |

### *2.3 Brief statistical analysis*

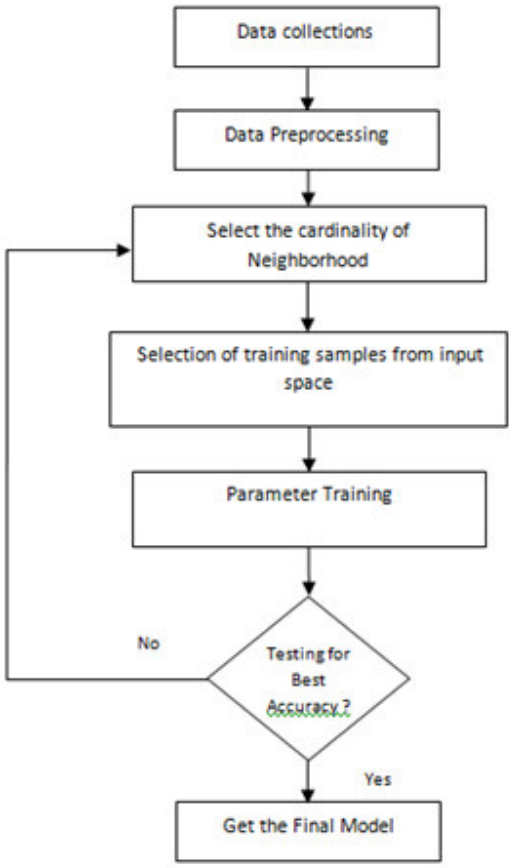
| Attribute number | Mean | Standard Deviation |
|---|---|---|
| 1. | 3.8 | 3.4 |
| 2. | 120 | 32.0 |
| 3. | 69.1 | 19.4 |
| 4. | 20.5 | 16.0 |
| 5. | 79.8 | 115.2 |
| 6. | 32.0 | 7.9 |
| 7. | 0.5 | 0.3 |
| 8. | 33.2 | 11.8 |

### *3.1 Methods and Terms for KNN Classifier*

In this section we discuss briefly the nature of the KNN method [7, 8]. One of the unsupervised learning is by KNN and it happens to be a non parametric lazy learning algorithm. Though there are no restrictions on the underlying data distribution, more memory is needed as we need to store all training data. KNN is free of any typical theoretical assumptions made like Gaussian mixtures and Linear seperability. To make it simple, we depend on Euclidean distance applied in the

feature space, an eight dimensional space. The significant parameter K influences the classification. In this paper we vary this from 1 to 10 based on the plateau seen at the performance most of the times. Kernel Density Estimation helps to determine the value of K by the density volume or coverage of neighbors. It is estimated as in the expression 'NV / K 'where V is the volume surrounding the data point, N is the total number of points in the data set and K is the points covered by this volume. KNN method is based on fixing the value of K and determining the minimum volume V that encompasses K points in the dataset. The proposed method is shown in the following figure 1.

**Figure 1**
***Iterative steps of Data Mining Procedure for the proposed Architecture***




The dataset is downloaded as mentioned in 2.1 and used in our experiment as shown in figure 1. The base classifier KNN is iterated as shown in the tables from 1 to 4.

### *3.2 Methods and Terms for Ensemble Design*

Bagging (bootstrap aggregating) builds a set of classifiers from successive bootstrap samples of the original training set. The final classifier combines individual KNN classifiers by a majority vote, a form of un weighted averaging [3]. The diversity of the ensemble combined with the final averaging is known to increase the robustness of the aggregated classifier. AdaBoost [12] can be attributed to its ability to enlarge the margin which could enhance the generalization capability of AdaBoost. It is also known that there is an accuracy/diversity dilemma in AdaBoost which means that the more accurate the two component classifiers become, the less they can disagree with each

other. Only when the accuracy and diversity are well balanced, can the AdaBoost demonstrate excellent generalization performance. However, the existing AdaBoost algorithms do not explicitly take sufficient measures to deal with this problem. Dagging [13] meta classifier creates many disjoint, stratified folds out of the training data and feeds each chunk of data to a copy of the supplied base classifier. Predictions are made out of averaging, since all the generated base classifiers are put into the Vote Meta classifier. They are useful for base classifiers that are worse in time behavior, regarding number of instances in the training data. MultiBoosting [12] is an extension to the highly successful AdaBoost technique for forming decision committees. MultiBoosting can be viewed as combining AdaBoost with wagging. It is able to harness both AdaBoost's high bias and variance reduction. Using C4.5 as the base learning algorithm, Multi-boosting is demonstrated to produce decision committees with lower error than either AdaBoost or wagging significantly more often than the reverse over a large representative cross-section of UCI data sets. It offers the further advantage over AdaBoost of suiting parallel execution. The design of proposed method is shown in figure 2. In the meta level we designate four types of meta classifiers Ada Boost, Bagging, Dagging and Multi boost. We perform 10-fold validation, a method of sampling without replacement and averaging the accuracies over 10 iterations. We stop at the training which takes time not less than the threshold duration to cover all possible ensembles. This is due to parameter training when the values chosen are not feasible to finishing the model building or testing the folds.

**Figure 2**
***Design of Ensembles***

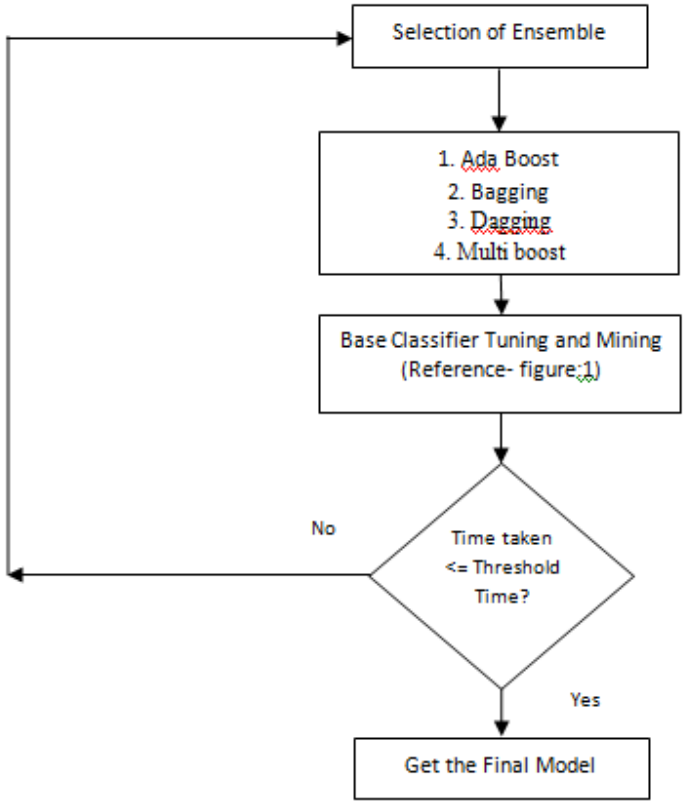


**Table 1**
***Iteration over K for the Ensemble : Adaboost***

| | K | Accuracy | Average Accuracy |
|---|---|---|---|
| Adaboost M1(Base Classifier: KNN) | 1 | 70.1823 | 71.679 |
| | 2 | 66.0156 | |
| | 3 | 72.6563 | |
| | 4 | 72.1354 | |
| | 5 | 73.1771 | |
| | 6 | 72.3958 | |
| | 7 | 74.7396 | |
| | 8 | 72.2656 | |
| | 9 | 72.1354 | |
| | 10 | 71.0938 | |

**Table 2**
***Iteration over K for the Ensemble : Bagging***

| | **K** | **Accuracy** | **Average Accuracy** |
|---|---|---|---|
| **Bagging (Base Classifier: KNN)** | 1 | 69.6615 | 71.84897 |
| | 2 | 70.1823 | |
| | 3 | 71.7448 | |
| | 4 | 71.6146 | |
| | 5 | 72.1354 | |
| | 6 | 72.1354 | |
| | 7 | 72.7865 | |
| | 8 | 72.3958 | |
| | 9 | 72.7865 | |
| | 10 | 73.0469 | |

**Table 3**
***Iteration over K for the Ensemble : Dagging***

| | **K** | **Accuracy** | **Average Accuracy** |
|---|---|---|---|
| **Dagging (Base Classifier: KNN)** | 1 | 74.8698 | 73.69792 |
| | 2 | 73.9583 | |
| | 3 | 74.2188 | |
| | 4 | 74.2188 | |
| | 5 | 74.2188 | |
| | 6 | 73.8281 | |
| | 7 | 73.6979 | |
| | 8 | 73.5677 | |
| | 9 | 72.3958 | |
| | 10 | 72.0052 | |

**Table 4**
***Iteration over K for the Ensemble : Multi-Boosting***

| | **K** | **Accuracy** | **Average Accuracy** |
|---|---|---|---|
| **MultiBoost (Base Classifier: KNN)** | 1 | 70.1823 | 71.02865 |
| | 2 | 66.6667 | |
| | 3 | 71.4844 | |
| | 4 | 70.0521 | |
| | 5 | 71.3542 | |
| | 6 | 70.3125 | |
| | 7 | 73.8281 | |
| | 8 | 72.0052 | |
| | 9 | 72.2656 | |
| | 10 | 72.1354 | |

**Figure 3**
***Comparison of Accuracies by Iterations over K***

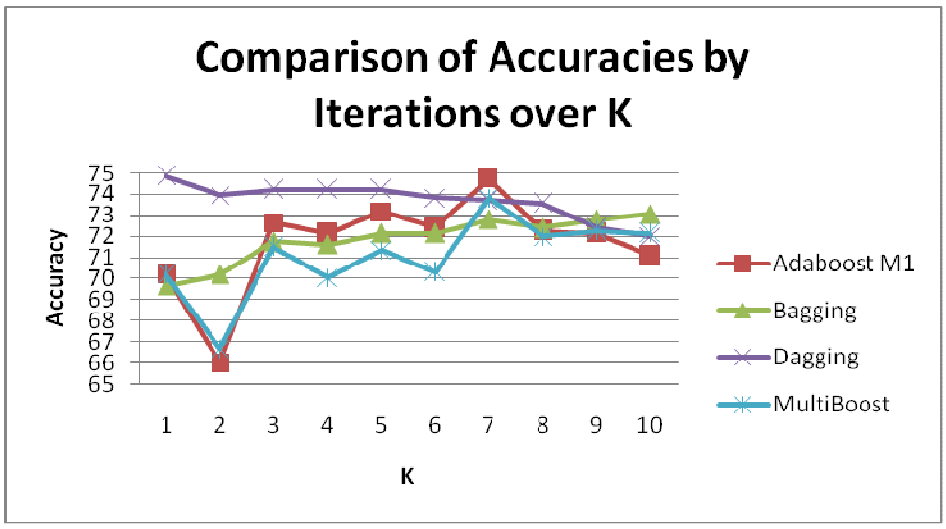
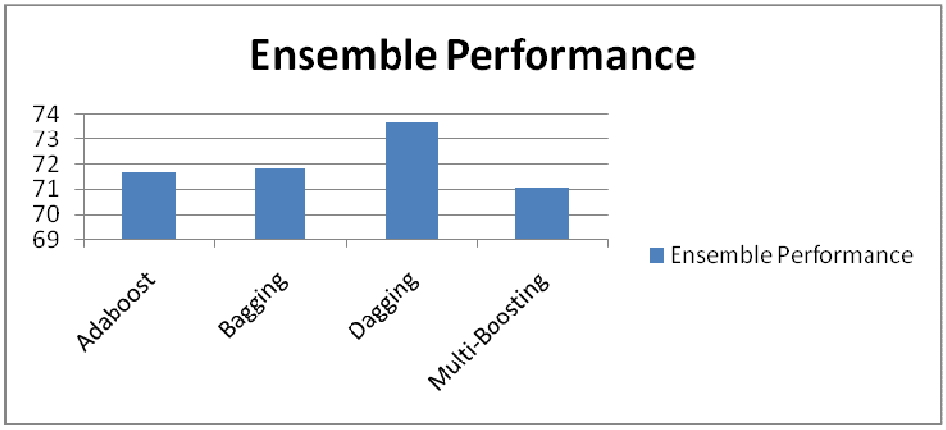



**Figure 4**
***Ensemble Performance***




## 4. RESULTS

The four tables illustrate the capacity of our design of ensemble purely made up of KNN classifiers. The average accuracy of the Adaboost, bagging , dagging and Multi-Boosting are 71.679, 71.84897, 73.69792 and 71.02865 respectively. Here we see Dagging for a neighborhood of size K<6, out performs the other classifiers as shown in the figure 3, Multiboost classifier performs low relatively most of the time. This frame work for design of ensemble may be extended to other set of uniform classifiers like SVM, Decision Trees, Bayes and Neural Networks.

## 5.CONLUSION

We observe the ensemble implemented by 'Dagging' achieves maximum accuracy for the diabetes dataset chosen.

## ACKNOWLEDGEMENT

Authors would like to thank management and research division of Bharath University, India for their support and encouragement for this research work.

## REFERENCES


1. D.Udhayakumarapandian., RM.Chandrasekaran., and A.Kumaravel "Enhancing The Accuracy 0f SVM Classifiers With Kernel And Parameter Training" Int J Pharm Bio Sci , 6(2): (B) 204 - 217, April(2015) (In press)
2. D.Udhayakumarapandian., RM.Chandrasekaran and A.Kumaravel "A Novel Subset Selection For Classification Of Diabetes Dataset By Iterative Methods" Int J Pharm Bio Sci ,5 (3) : (B) 1 – 8, July(2014)
3. L. Breiman., Bagging predictors. Machine Learning 24:123-140, 1996.
4. A.Kumaravel., Pradeepa.R., Efficient molecule reduction for drug design by intelligent search methods.Int J Pharm Bio Sci, 4(2): (B) 1023 – 1029,Apr (2013)
5. L. Breiman., Bagging predictors. Machine Learning 24:123-140, 1996.
6. L. Breiman., Prediction games and arcing algorithms. Neural Computation 11:1493-1517, 1999.
7. D. W. Aha., D. Kibler and M. K. Albert, Instance-based learning algorithms, Machine Learning, 6, 37–66 , 1991.
8. Brighton.H., Mellish.C, Advances in instance selection for instance-based learning algorithms, Data Min. Knowl. Discov. 6 **,** 153–172 , 2002.
9. H.Dunham., Data Mining, Introductory and Advanced Topics, Prentice Hall, ISBN-10: 0130888923 Published 08/22/2002.
10. Website for weka software http://www.cs.waikato.ac.nz/ml/weka/ downloaded on 3rd august 2014
11. Breiman.L,"RandomForestsin Machine Learning", vol. 45, pp. 5-32, 2001
12. Geoffrey I. Webb MultiBoosting: A Technique for Combining Boosting and Wagging. Machine Learning. Vol.40(No.2), 2000.
13. Ting K. M., Witten, I. H.: Stacking Bagged and Dagged Models. In: Fourteenth international Conference on Machine Learning, San Francisco, CA, 367-375, 1997.
14. A.Stensvand., T. Amundsen, L. Semb, D.M. Gadoury, and R.C. Seem.. Ascospore release and infection of apple leaves by conidia and ascospores of Venturia inaequalis at low temperatures. Phytopathology 87:1046-1053, 1997.
15. Website for attribute description http://archive.ics.uci.edu/ml/machine-learningdatabases/pima-indians diabetes., accessed on 3rd august 2014
16. Chan.P.,Stolfo.S. A comparative evaluation of voting and meta-learning on partitioned data. Twelfth International Conference on Machine Learning,(P)15-17, 1995.
17. Freund.Y ., Schapire.R Experiments with a new boosting algorithm. Thirteenth International Conference on Machine Learning, 1996.