Modified Stacked Generalization with Sequential learning

Ms. Bhoomi Trivedi
INDUS institute of Eng. & Tech
Ahmedabad

Ms. Neha Kapadia
TCET, Kandivali(E)
Mumbai

ABSTRACT
Nowadays machine learning techniques can be successfully applied to data mining tasks. In inductive machine learning, combination of several classifiers is very lively field and has shown favorable results compare to those of single expert systems for variety of scenarios. In this paper one of the ensemble learning method, i.e stacked generalization is modified to get better predictive accuracy. In stacking, by knowing its area of expertise, different diverse base classifiers are combined by a learnable combiner. So error can be generalized by the combiner. As diversity is the important aspect of the ensemble learning, in this paper sequential learning of the base classifier is experimented for that. To evaluate the performance of the proposed method different data sets like, IONOSPHERE, HYPOTHYROID, WAVEFORM are used. The experiments demonstrate the efficiency of the proposed model in terms of accuracy and time by yielding higher accuracy and lesser time relative to conventional staked generalization method.

General Terms
Stacked Generalization, Classification, Training phase, Application phase

Keyword
Stacked Generalization, sequential stacked generalization, ensemble learning, multiple classifier system.

1. INTRODUCTION
Varieties of learning algorithms are available. These algorithms can be clustered in four natural groups,

- ✓ Decision tree learner such as C4.5.
- ✓ Simple learners such as Naïve Bayes.
- ✓ Linear discriminants such as MLR.
- ✓ Instance based learner such as Ibk or kstar.

Both empirical observations and specific machine learning applications confirm that a given learning algorithm outperforms all others for a specific problem or for a specific subset of the input data, but it is unusual to find a single expert achieving the best results on the overall problem domain. Leading approach for choosing the classifiers empirically is by estimating the candidate accuracy via cross validation and selects the one that is most accurate. This method is on and average shows same or somewhat better result compared to single base classifier but if we increase the no of base classifiers, crossvalidation[1] method often picks the wrong base algorithm. As a consequence multiple learner systems (an ensemble of classifiers) try to exploit the local different behaviour of the base learners to enhance the accuracy and the reliability of the overall inductive learning system. Numerous methods have been suggested for the creation of ensemble of classifiers. Multiple classifier system (Ensembles of classifiers) are build by different mechanisms. Some of them are like, 1) by using single learning algorithm and different subset of training dataset (bagging) 2) by single learning algorithm with different training parameters (boosting) 3) by using different learning algorithm for each classifier (stacking).

As crossvalidation essentially computes a prediction for each example in the training set, it was soon realized that this information could be used in more elaborate ways than simply counting the number of correct and incorrect predictions. In Stacking generalization accuracy is achieved by using two phases of processing: one by reducing biases employing a mixture of algorithms, and the other by learning from meta-data the regularities inherent in base-level classifiers. Stacking introduces the concept of a meta-learner, which replaces the voting procedure. Stacking tries to learn which base-level classifiers are the reliable ones, using another learner, meta learner, to discover how best to combine the output of the base learners.

For building good ensemble, most important key is whether the classifiers in a system are diverse enough from each other, or in other words, that the individual classifiers have a minimum of failures in common. If one classifier makes a mistake then the others should not be likely to make the same mistake. To achieve diversity in the base classifier training, sequential learning method is used in conventional stacked generalization method in the proposed algorithm. We applied proposed model to the different datasets, taken from the UCI repository. The experimental results show that proposed model indeed improve the classification accuracy compared with the original stacking and boosting methods.

This paper is organized as follows. In Section 2 stacked generalization is explained. Section 3 and 4 dimensions of the stacking and proposed method respectively. Experimental results are given in Section 5 to show the effectiveness of the proposed method. Conclusions are given in Section 6.

In figure 1.1 whole path from the data mining to stacked generalization is shown.

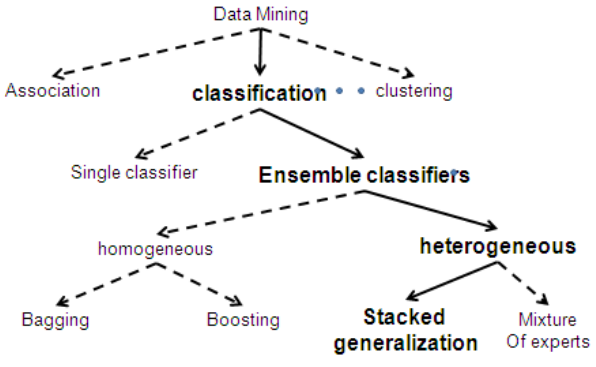


Fig.1.1 path to the stacked generalization

2. STACKED GENERALIZATION

Stacked generalization is the layered architecture. The classifiers at the layer-0 (Level-0) receive the original data as their input, and each classifier outputs a prediction for its own sub problem. Successive layers receive the predictions of the layer immediately preceding it as an input and finally a single classifier at the top level outputs the final prediction. Stacked generalization attempts to minimize the generalization error by using classifiers at higher layers to learn the type of errors made by the classifiers immediately below.

Robi Polikar [12] has explained basic concept of stacking with two layers (level) in his ensemble learning paper. Figure 1.1 illustrates the stacked generalization approach, where C_1… Ct are the base classifiers, located at the level-0. These base classifiers are trained using instances of the training data set with parameters θ_1 through θ_T (where may include different training datasets, classifier architectural parameters, etc.) to output hypotheses h_1 through h_t . The outputs of these classifiers are used in the training of the meta classifier, which is at the next level. True classes of the training data set are also used in the training of meta classifier, C_{T+1}. Traditionally, the *k*-fold selection process described in above is used to obtain the training data for classifier C_{T+1}. Specifically, the entire training dataset is divided into T blocks, and each first-level classifier C_1. . . C_T is first trained on (a different set of) $T - 1$ blocks of the training data. Therefore, there is one block of data not seen by each of the classifiers C_1through C_T. The outputs of each

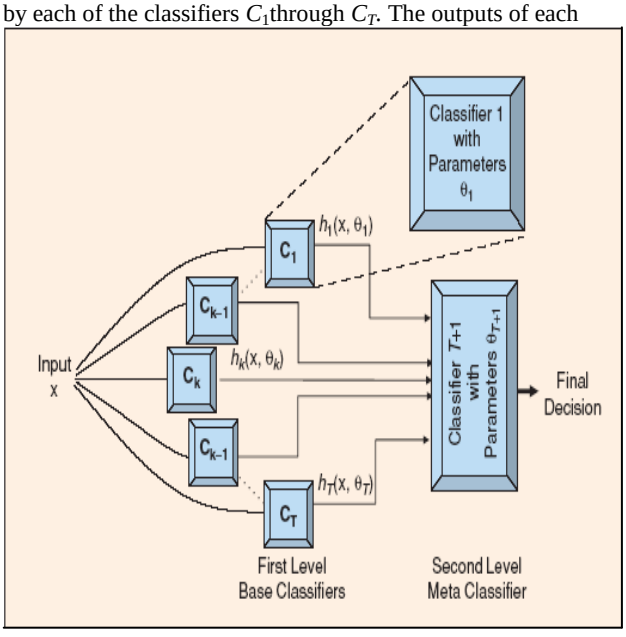


Fig 2.1 Stacked generalization

classifier for the block of instances, on which it was not trained, along with the correct labels of those instances, constitute the training data for the second level meta-classifier C_{T+1}. Once C_{T+1} is trained, all data are pooled, and individual classifiers C_1, …, C_T are retrained on the entire database, using a suitable resampling method.

So for the additional cost of running an appropriate meta classifier it is possible to utilize all the output generated by a crossvalidation. Furthermore, the dimensionality of the meta dataset is equal to the number of classes multiplied by the number of base classifiers and thus fairly independent of the dimensionality of the original dataset. The additional training cost for the meta classifier is usually much smaller than the training costs for the base Classifiers, especially for large, high-dimensional datasets.

2.1 Algorithm of stacked generalization

Training Phase:

1. Train the component classifiers using leave-one-out cross validation as follows. For each instance in the data set, train each of the level-0 classifiers using the remaining instances. After training, classify the held-out instance using each of the trained level-0 classifiers. Form a vector from the predictions of each of the level-0 classifiers and the actual class of that instance.

2. Train the level-1 classifier, using as the level-1 training set the collection of vectors of the level-0 classifier predictions and the actual classes. This collection has cardinality j to j, since there is one level-1 training instance corresponding to each level-0 training instance.

3. Since the level-0 classifiers have not been trained on the entire training set, re-train the level-0 classifiers on the entire training set.

Application phase:

When presented with a new instance whose class is unknown, classify the instance using each of the level-0 classifiers, deriving an input vector for the level-1 classifier. The derived vector is then classified by the level-1 classifier, which outputs a prediction for the new instance.

3. STACKING DIMENSIONS

From the above algorithm, we can say basic stacking algorithm contains three independent dimensions.

3.1 Base classifier choice:

Base classifier can be any arbitrary machine learning algorithms. But if probability of classifiers is going to be used then only those classifiers can be base classifiers, which give the probability of class. From different research it seems that stacking works well with 3 or 5 or 7 classifiers. From researches it is observed that C4.5,nearest neighbor and naïve bayes is the good choice for the base classifier combination.

3.2 Meta classifier choice:

From the group described above, any classifier can be chosen as a meta classifier. Which meta classifier to use for combining the prediction of base classifiers is dependent on the meta data also. Whether meta data is class prediction or probability of the class. According to Wolpert, relatively global and smooth classifiers should perform well [1] at the meta level.

3.3 Meta data:

Which type of meta data should be produced i.e. class label or probability distribution from the base classifiers is also very important parameter for stacking. From these two types of meta data class probability has given better results comparably. Class probability meta data with MLR meta learner has given significant improved stacking method. If class predictions are going to be used as a meta data Naïve Bayes is good choice as a meta learner.

3.4 Current approaches of stacked generalization

Moreover, different ensemble approaches have also been combined with the stacked generalization to achieve higher accuracy or diversity. Some approaches are bagging with the stacked generalization, stacking with dagging. Figure 3.1 abstracts different approaches.

4. PROPOSED MODEL

Proposed algorithm, "Sequential stacking" is based on the diversity criteria in sacked generalization [stacking]. New technique, sequential learning, is used to train the classifiers. With this concept base classifiers can get more diversity and base classifiers covers different area in error surface. So in

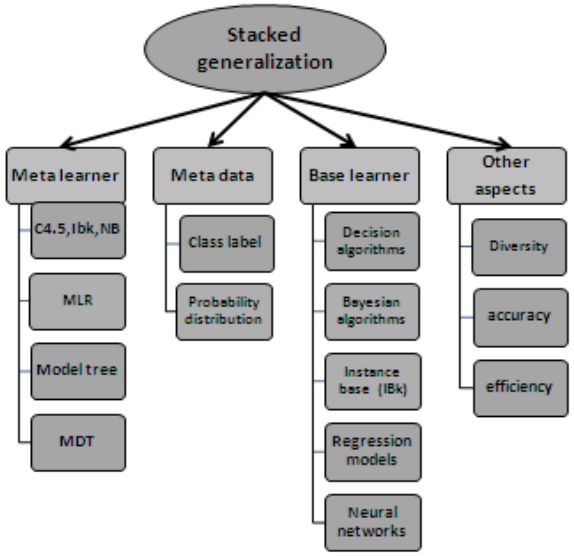


Fig 3.1 Approaches of stacked generalization

combining stage error surface will be covered in such a way that the classification accuracy is efficiently improved.

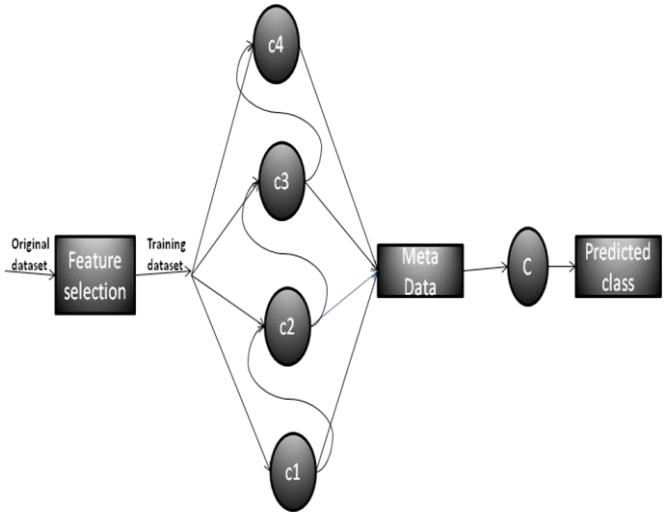


Fig 4.1 proposed model

Figure 4.1 depicts the way Sequential stacking is conceptualized. In the place of creating no of folds for cross validation for training of base classifiers, sequential learning is used to create diverse base classifiers. Different area of error surface is created by this and by the combining the prediction of these diverse classifier we can get better accuracy.

In figure , for training level 0 classifiers, version of boosting algorithm is used for sequential learning, After training the first classifier, a copy of the "difficult" training samples is added to the next training set which is used to train the second classifier. This procedure is repeated for all base classifiers. After the training the level-0 networks, they are run with the training set to provide a new training set for the level-1 network. This generates a single pattern for a new data set which will be used to train the level-1 network. The inputs of this pattern consist of the outputs of all the level-0 networks, and the target value is the corresponding target value from the original full data set. In figure 4.2 pseudo code of the proposed model is given.

Input: 1) dataset St = $\{(x_1,y_1),(x_2,y_2),\ldots,(x_m,y_m)\}$;

First level learning algorithms $L_1,L_2,\ldots,L_T$;

Second level learning algorithm L

Output: $H(x)=h'(h_1(x),\ldots,h_T(x))$

Method:

Step 1: Apply the feature selection method to the original dataset to generate the training dataset.

Step 2: Train the base level classifiers with the sequential learning.

- initialize $D_1(i)= 1/N$, i=1,2,…,N

  Do for t=1,2,…,T:
- Select a training data subset St, drawn from the distribution Dt.
- Train weak learner Lt with St, receive hypothesis ht.
- Calculate the error of

$$h_t : \varepsilon_t = \sum_{i:h_t(x_i)\neq y_i} D_t(i)$$

- if $\varepsilon_t > ½$, abort.

  Set βt = εt/(1-εt).
- Update distribution

$$\text{Dt:Dt+1(i)=} \frac{D_t(i)}{Z_t} \times \{\beta_t \quad \text{if } h_t$$

  $(X_i)=y_i\}$,1 otherwise
- Where Zt = ΣiDt(i) is a normalization constant chosen so that Dt+1 becomes a proper distribution function.

Step 3: Generate new data set (meta data)

St'=NULL

For i=1,2,…,m

For i=1 ,2,…,T

Zit =ht(xi)

End;

St'=St'U{(zi1,zi2,…,ziT),yi)}

End;

St'=St' U { (zi1,zi2,…,ziT),yi)}

End;

Step 4: train the second level learner h' by applying the second level learning algorithm L to the new data set D'

h' = L(St')

Figure 4.2 Pseudo code of proposed model

5. EXPERIMENTAL RESULTS

Here proposed model is experimented and results are shown below are the data after experiment. For experiment different data set are used from the UCI repository. The algorithms are tested to see the accuracy performance. Other factors which can affect the classification methods are like no of instances, no of attributes, type of classes, type of classifier etc. in this chapter most affected factors are included in performance study of the new algorithm. Moreover proposed method is the multiple classifier system, so it is also compared with the different types of MCS. Proposed method is also tested on different types of data set. Proposed method is also tested with single learning algorithm as a base algorithm for base classifiers and multiple learning algorithms for different classifiers.

5.1 Performance study of different ensemble methods

In the Table 5.1 comparison between different ensemble method is shown. Comparison is done based on different data sets and it is done between basic known ensemble methods like, bagging, boosting and stacking. The results show that for some data set boosting is giving higher accuracy and for some data set stacking is giving higher accuracy. Figure 5.1 shows the graphical representation of this result.

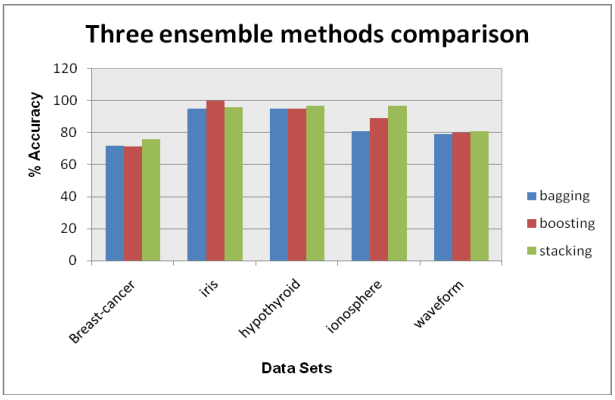


Fig 5.1 different ensemble method accuracy

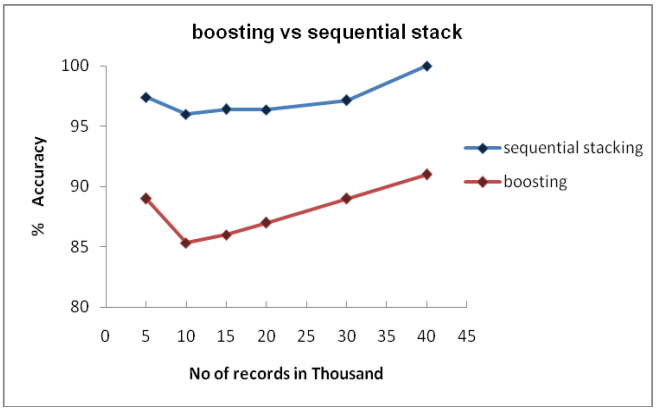
5.2 Performance Study on Classifier Accuracy

Performance comparison of accuracy between ensemble methods and proposed approach is given in the table 5.2**.** The performance result is derived on "ionosphere" dataset of different size. . The accuracy of the model has been tested for three different methods. Figure 5.2 shows its graphical representation.

The accuracy of the model has been tested for three different methods. Figure 5.2 shows its graphical representation.

Table 5.2 accuracy of proposed model with no of records

No. of records (In Thousands)	Boosting	stacking	Seq_stack
	Accuracy (%)	Accuracy (%)	Accuracy (%)
5	89	94	97.4
10	85.31	92.5	96
15	86	92	96.4
20	87	93.7	96.36
30	89	94	97.14
40	91	95.3	100



Data set Name	Bagging	Boosting	Stacking
	Accuracy (%)	Accuracy (%)	Accuracy (%)
Breast-cancer	72	75.17	76
Iris	95	100	96
hypothyroid	95	95	97
ionosphere	81	89	97
waveform	79	80	81

Fig 5.2 Accuracy of proposed model with number of records

From Table 5.2 and Figure 5.2 it is shown that sequential_stacking is giving greater accuracy then compare to boosting. Stacking method is not giving higher accuracy then the sequential learning but it is taking too much time to train the model even. It is taking 20 seconds in compare to 1.40 seconds of boosting or sequential stacking. From the figure it is also noticeable that after some number of records the accuracy is increasing. As in our figure after 10000 records the accuracy of the classifier is increasing in both the method.

5.3 Performance Study on different data sets

In Table 5.3 comparison between three methods is given based on different data sets. Figure 5.3 is giving its graphical representation. With most of the data set sequential learning higher accuracy than boosting or basic stacking method. But in the case of breast cancer data set it is giving lower accuracy compared to other.

Table 5.3 accuracy with different type of data

Data set Name	Boosting Time (sec)	stacking Time (sec)	Seq_stacking Time (sec)
ionosphere	0.99	13.95	1.46
iris	0.05	0.08	0.06
Breast-cancer	0.03	0.11	0.06
hypothyroid	0.6	1.23	0.5
waveform	2.35	15.9	2.19
adult	2.57	77.93	7.8

Table 5.4 time required in sec by ensembles

Data set Name	Boosting Accuracy (%)	stacking Accuracy (%)	Seq_stackingg Accuracy (%)
Ionosphere	89	97	97.4
Iris	100	96	97
Breast-cancer	75	75.52	63
Hypothyroid	95	97	99.81
waveform	80	81	86

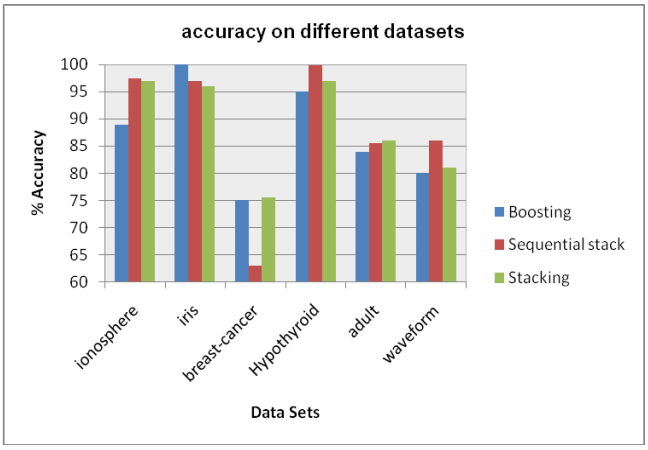


Figure 5.3 accuracy with different type of data

5.4 Time required by different ensembles

In table 5.4, it is shown that sequential stacking is giving higher result compared boosting and stacking but the time required for the stacking method compared to sequential stacking and boosting is much larger. The comparison table for the time taken by all methods for these data set is given in the Table 5.4. In the above table it is explicitly shown that the time required for the adult data set having 32000 records, for stacking is unpredictable. It is very very high in compare to other methods. Moreover the comparison for the different meta classifier is also done. Naïve Bayes , J48, simple logistic classifiers were used as the meta learner and the conclusion is that J48 is giving good performance and simple logistic also giving good performance but the time required by the J48 is very large compared to simple logistic. And as a meta learner no need to choose meta leaner that consumes larger time in training but in that place the smooth function can be beneficial.

6. CONCLUSION

we proposed new model which used sequential learning as a diversifying approach in the base classifiers, instead of cross-validation for more diverse and independent base classifiers. To combine the predictions of base classifiers conventional stacked generalization method is used. Combiner combines the results by knowing the classifiers expertise, so errors are generalized and efficiency has been improved with proposed model. By analyzing the result it is concluded that stacking with sequential learning is giving better accuracy than the stacked generalization with cross validation. Proposed method gives approximately 4 to 7 % increase in accuracy. In terms of time, proposed method reduces time taken to build the model to the greater extent.

7. FUTURE ENHANCEMENT

In future we intend to extend this research in following ways. Research can be done in the base classifier selection also. Select the no of classifiers based on some criteria like, accuracy, diversity etc. with this we can do pruning of the classifiers. We can use genetic algorithm for the base classifier selection also. Dynamic classifier selection can be used in the classifier selection also. Moreover, advantage of bagging, boosting and stacking can be combined as a hybridization of ensemble methods. Bootstrap generation with weighted classification can be done after base level classification, meta learner can be used to generalize the error. Multinomial logistic regression can be used as a meta learner.

8. REFERENCES

[1] DAVID H. WOLPERT, Ph.D., *"Stacked Generalization"*, Complex Systems Group, Theoretical Division, and Center for Non-linear Studies, Los Alamos.

[2] TING, K. M., & WITTEN, I. H. "Issues in stacked generalization." *Journal of Artificial Intelligence Research.*

[3] TODOROVSKI, L., & DˇZEROSKI, S" *Combining multiple models with* meta *decision trees".* In *Proceedings of the Fourth European Conference on Principles of Data Mining and Knowledge Discovery,* Berlin, Springer.

[4] ZENKO, B., & DˇZEROSKI, "*Stacking with an extended set of meta-level attributes and MLR.*" In *Proceedings of the Thirteenth European Conference on Machine Learning*, Berlin: Springer.

[5] ZENKO, B., TODOROVSKI, L., & DˇZEROSKI,"*A comparison of stacking with MDTs to bagging, boosting, and other stacking methods"*. In p*roceedings of the First IEEE International Conference on Data Mining,* Los Alamitos, IEEE Computer Society.

[6] AGAPITO LEDEZMA, RICARDO ALER AND DANIEL BORRAJO" *Empirical Study of a Stacking State-space* **",** Universidad Carlos III de Madrid Avda. de la Universidad, 3028911 Legan´es. Madrid (Spain)

[7] SASO DˇZEROSKI, BERNARD ˇZENKO "*Is Combining Classifiers with Stacking Better than Selecting the Best One?***"** *Department of Knowledge Technologies, Joˇzef Stefan Institute, Jamova 39, SI-1000 Ljubljana, Slovenia*

[8] ELEXANDER K SEEWALD "*exploring the parameter stacking state space",Australian research institute of artificial intelligence,schottengasse 3, A-1010 Wien,Austria.*

[9] CROUX C.JOOSSENS K. AND LEMMENS A. "Bagging a stacked classifier".

[10] S.B. KOTSIANTI AND D. KANELLOPOULOS, "*Combining Bagging, Boosting and Dagging for Classification Problems***"** *Educational Software Development Laboratory Department of Mathematics University of Patras.*

[11] MARTIN SEWELL "*Ensemble Learning" Department of Computer Science University College London* April 2007

[12] ROBI POLIKAR, "Ensemble based system in decision making".

[13] METE OZAY AND FATOS TUNAY YARMAN VURAL, **"***On the Performance of Stacked Generalization Classifiers"(2008)***.**

[14] CHRISTOPHER J. MERZ "*Using Correspondence Analysis to Combine Classifiers"***,** *Department of Information and Computer Science, University of California, Irvine*

[15] ALEXANDER K. SEEWALD "*Towards Understanding Stacking "* Studies of a General Ensemble Learning scheme , Phd thesis.

[16] David B. Skalak, "*Prototype Selection for Composite Nearest Neighbor Classifiers*", *Department of Computer Science University of Massachusetts Amherst, Massachusetts 01003-4160*