Evidence Combination for Baseline Accuracy Determination

Teodora Moldovan, Camelia Vidrighin, Ioana Giurgiu, Rodica Potolea
Technical University of Cluj-Napoca
Camelia.Vidrighin@cs.utcluj.ro

Abstract

Several classifier combination approaches have been proposed in machine learning literature in order to enhance the performance of simple learning schemes. This paper presents a new classifier fusion system based on the principles of the Dempster-Shafer theory of evidence combination. The system tackles the advantages of combining different sources of information to attain a high degree of stability across different problem domains. The uncertainty evaluation provided by the Dempster-Shafer theory also contributes to achieving this stability. System evaluation has confirmed the assumptions related to stability and allows us to formulate a method of establishing the baseline accuracy for any problem domain. Thus, the choice of a specific learning scheme for a certain problem is justified only if it's performance is better than that of the system proposed here.

1 Introduction

In the field of data mining, one of the main objectives is to achieve the highest possible classification accuracy. A classification algorithm used successfully with a specific set of features may not be appropriate with a different set of features. In addition, classification algorithms are different in their theories, and hence they achieve different degrees of success for different applications. These particularities have led to an increasing interest towards trying to combine the predictions of several algorithms, in order to obtain a scheme that performs well in several different areas.

The combination approach tries to overcome several drawbacks related to algorithms which use a single hypothesis: the statistical problem, the computational problem and the representation problem.

The statistical problem, or high variance, arises when the hypothesis space is too large for the available training data. When this occurs, the learning algorithm may be forced to choose as output one of several hypotheses that achieved the same accuracy on the training data. However, there is a risk it will fail to choose the best one, therefore compromising future predictions. A simple and elegant solution is provided by the use of ensemble learning methods: a vote from all equally good classifiers can reduce the risk.

The computational problem, or computational variance, occurs when the learning algorithm cannot guarantee to find the best hypothesis within the computational space. Such issues may appear when heuristic methods need to be used to address the computational complexity of the search problem (such as for decision trees, or artificial neural networks). A weighted combination of several local minima can reduce the risk of choosing the wrong output.

The representation problem, or high bias, arises when none of the hypotheses in the search space is a good enough approximation of the truth. In this case, a weighted sum of hypothesis can form a more accurate approximation.

While ensemble methods can reduce both the bias and the variance of learning algorithms, they do not solve the problem of failing to choose a classifier that will perform best. In addition, there is also a problem of establishing a lower bound to the accuracy on a certain problem. This is one of the issues tackled in this paper.

There are two main approaches to combining classifiers: ensemble methods and fusion methods. Ensemble methods combine several hypotheses obtained by the same base learner, and fusion methods are based on the data fusion principles.

The ensemble methods run a specific learning algorithm (single classifier) multiple times on the same dataset and form a hypothesis at each run. There are several ways to obtain the set of hypotheses. The most prominent are Breiman's *Bagging* and Schapire's *Boosting*. *Bagging* (Bootstrap Aggregating) [1]

constructs each hypothesis independently, by providing a different training set to each individual learner. The resulting group of hypotheses contains members that are accurate enough, and yet diverse enough, such that the accuracy of the ensemble is higher than the accuracy of any individual. A second method for constructing ensembles is *Boosting* [2]. In this additive approach, the set of hypotheses is obtained during several boosting phases. Each distinct model is built through the same learning mechanism, by varying the distribution of examples in the training set. After each boosting phase, the weights of the misclassified examples are increased, while those of the correctly classified examples are decreased.

A second approach to combining classifiers is represented by the classifier fusion techniques based on the data fusion principles. Data fusion allows a more informed decision about a particular phenomenon by extracting complementary pieces of information from different sources.

As different classifiers may offer complementary information about the features to be classified, combining classifiers in an efficient way can yield better classification results. The combination may or may not perform better than the best classifier in the system, but it certainly reduces the overall risk of making a particularly poor selection.

One of the noticeable differences between classifier fusion methods and ensemble learning methods is the training set. While ensemble learning uses classifiers that have been trained on different sampling of the original dataset, the data fusion technique can use data obtained from different sources.

Several classifier fusion approaches have been proposed for this purpose, including combining classifiers using ARTMAP [8], Learn++ [9], genetic algorithms [3] and other combinations of boosting/voting methods.

This paper presents a new system based on a new classifier combination methodology, which uses the advantages of the Dempster-Shafer theory of evidence. The Dempster-Shafer theory is a powerful method for combining evidence from different classifiers.

The classifier combination technique has been successfully applied in multi-sensor data fusion for domains like threat analysis in computer security [4], satellite image classification, colour image segmentation, etc. [5].

The system is intended to provide a reference accuracy value when choosing a classifier for any specific dataset. Due to the advantages provided by the fusion technique, the risk is minimized and the accuracy obtained by applying the combined classifier on any data is surely among the highest possible. Thus, the choice of a classifier on a specific dataset is justified only if the classifier shows higher accuracy than the combined classifier on that particular dataset.

A powerful argument in choosing the combination method proposed by the Dempster-Shafer theory is that it takes into consideration uncertainty. Ensemble methods treat uncertainty as failure to classify an instance. The uncertainty of a classifier involved in a Dempster-Shafer combination scheme doesn't qualify as misclassification, but demands for a more detailed investigation.

The rest of the paper if organized as follows: Section 2 is an overview on the Dempster-Shafer theory and the belief combination technique. Section 3 proposes a new system based on the formal model for the classifier combination methodology following the Dempster-Shafer theory. Section 4 is a description of the experimental setup, the datasets used in testing the system and the results obtained. Also, this section presents a comparison between our system and ensemble learning methods. Section 5 summarizes the results, draws the conclusions and notes the directions for future work in this area.

2 The Dempster-Shafer Theory

The Dempster-Shafer Theory (DST) is a mathematical theory of evidence, based on belief functions and plausible reasoning. Its main feature is that it combines several pieces of information in order to compute the probability of an event. Initial efforts for developing the theory were made by A. Dempster (1967), but the theory was completed by the seminal work performed by G. Shafer (1976) [10].

In a finite discrete space, DST can be interpreted as a generalization of the probability theory, where probabilities are assigned to sets, as opposed to mutually exclusive singletons (in probability theory, evidence is associated with only one possible event, while in DST evidence can be associated with sets of events). DST becomes the classical probability theory when there is enough evidence to allow the assignment of probabilities to individual events.

Maybe the most important feature of DST is that the model is designed to handle varying levels of information precision, without having to make any assumptions about how that information is "divided" further down. Moreover, it allows for directly representing the uncertainty of system responses: the imprecise input can be modeled by a set or an interval, and the output is a set or an interval.

2.1 Basic concepts

There are several basic concepts related to DST: a set, called the frame of discernment, and the following functions: basic probability assignment (*bpa* or *m*), belief function (*Bel*), commonality function (*Q*), doubt function (*Dou*), plausibility function, or upper probability function (*Pla*).

Frame of discernment

Also known as the *universe of discourse*, the frame of discernment θ is a set of mutually exclusive and exhaustive possibilities in a domain. If we take for example the roll of a dice, $\theta = \{1, 2, 3, 4, 5, 6\}$. The elements of 2^θ form the class of general propositions in the domain (just like in classical probability theory). The difference from the probability theory can be found in the fact that we may not know the probability assignments for all the elements in θ, and still be able to reason, without needing to assign probabilities to them using the principle of insufficient reasoning.

Basic probability assignment

Definition. A function $m : 2^\theta \rightarrow [0, 1]$ is called a *basic probability assignment* if it satisfies:

(i) $m(\Phi) = 0$

(ii) $\sum_{A \subseteq \theta} m(A) = 1$

The quantity $m(A)$ is defined as A's basic probability number, and it represents the proportion of all relevant and available evidence that supports the claim that a particular element of θ belongs to set A, but to no particular subset of A. The value $m(A)$ is relevant only to the set A, and makes no additional claims about any subset of A. Any further evidence on the subsets of A would be represented by another *bpa*, i.e. $m(B)$ would be the *bpa* for B, $B \subset A$.

Some researchers have found it useful to interpret the *bpa* as a classical probability. Although this interpretation has proven to be useful, it does not cover the full scope of the representational power of the *bpa*.

Belief function

Definition. A function $Bel : 2^\theta \rightarrow [0, 1]$ is called a *belief function* if it satisfies:

(i) $Bel(\emptyset) = 0$

(ii) $Bel(\theta) = 1$

(iii) For any collection $A_1 \ldots A_n$, $A_i \subset \theta$, we have

$$Bel(A_1 \cup \ldots \cup A_n) \geq \sum_{I \subseteq \{1,\ldots,n\}, I \neq \Phi} (-1)^{|I|+1} Bel(\bigcap_{i \in I} A_i)$$

A belief function assigns to each subset of θ a measure of our total belief in the proposition represented by the subset.

One, and only one *bpa* corresponds to each belief function. The relation is symmetric. They are related by the following two formulae:

$$Bel(A) = \sum_{B \subseteq A} m(B), \text{ for all } A \subseteq \theta \quad (1)$$

$$m(A) = \sum_{B \subseteq A} (-1)^{|A-B|} Bel(B) \quad (2)$$

Thus, a belief function and a basic probability assignment convey exactly the same information.

Commonality function

Definition. A function $Q : 2^\theta \rightarrow [0, 1]$ is called a commonality function, if there is a basic probability assignment, such that:

$$Q(A) = \sum_{A \subseteq B} m(B), \text{ for all } A \subseteq \theta \quad (3)$$

If Q is a commonality function, then the function defined by

$$Bel(A) = \sum_{B \subseteq \neg A} (-1)^{|B|} Q(B) \quad (4)$$

is a belief function. From this belief function, the basic probability assignment can be recovered using (2). If it is substituted in (3), the original Q results.

Therefore, the sets of belief functions, basic probability assignments, and commonality functions are in one-to-one correspondence, and each representation conveys the same information as any of the others.

Doubt function

Definition. Given a belief function *Bel,* the doubt function is defined by:

$$Dou(A) = Bel(\neg A)$$

Plausibility function

Definition. Given a belief function *Bel,* the upper probability function, or the plausibility function is defined by:

$$Pla(A) = 1 - Dou(A)$$

This expresses how much we should believe in A, if all currently unknown facts were to support A. Therefore, the true belief in *A* will be somewhere in the interval [*Bel(A)*, *Pla(A)*] .

2.2 The combination rule

Consider two basic probability assignments $m_1(.)$ and $m_2(.)$ for belief functions $Bel_1(.)$ and $Bel_2(.)$, respectively.

Let A_j and B_k be focal elements of Bel_1 and Bel_2, respectively. Then $m_1(.)$ and $m_2(.)$ can be combined to obtain the belief mass committed to *C*, where *C* is a set of all subsets produced by $A \cap B$, to the following combination or orthogonal sum formula (Shafer, 1976):

$$m(C) = m_1 \oplus m_2(C) = \frac{\sum_{j,k,A_j \cap B_k = C} m_1(A_j) m_2(B_k)}{1 - \sum_{j,k,A_j \cap B_k = \varnothing} m_1(A_j) m_2(B_k)},$$

$$C \neq \varnothing \qquad (5)$$

The denominator is a normalizing factor, which intuitively measures how much $m_1(.)$ and $m_2(.)$ are conflicting.

2.3 Combining several belief functions

The combination rule can be easily extended to several belief functions by adding new beliefs in cascade. Thus, the pair wise orthogonal sum of *n* belief functions $Bel_1, Bel_2, \ldots, Bel_n$, can be formed as:

$$((Bel_1 \oplus Bel_2) \oplus Bel_3) \ldots \oplus Bel_n = \overset{n}{\underset{i=1}{\oplus}} Bel_i \qquad (6)$$

3 The System

The system we propose uses the Dempster-Shafer theory of belief to obtain the fusion of three classifiers: the Naïve Bayes classifier, k-Nearest Neighbour (kNN) and the decision tree learner.

There are three main steps in designing the system, namely belief extraction from the three classifiers, uncertainty computation, and belief combination.

3.1 Extracting beliefs from classifier outputs

This step is done by taking into consideration the nature of each classifier. Therefore, the following approaches have been used:

1) For the Bayesian classifier, basic belief evaluation is done using the posterior probability function.

2) For k-Nearest Neighbour, a distance function is used to evaluate basic beliefs.

We have evaluated the distance by:

$$Distance\ measure = e^{-\frac{d_s}{d_{mean}}}$$

where $e^{-\frac{d_s}{d_{mean}}} = 1$, when $ds = 0$,

and $\lim_{ds \to \infty} e^{-\frac{d_s}{d_{mean}}} = 0$

Thus, the belief mass of a class is the average of all such distance measures voting for that class. Belief masses for the classes are then normalized, such that

$\sum_{i=1}^{K} m(i) = 1$, where *K* is the number of classes

3) For the decision tree, the confidence is the measure for evaluating beliefs.

The output of a decision tree learner is a decision tree. Association rules can be extracted from this tree, of the type $A \rightarrow B$. In addition, an association rule has support and confidence associated with it.

$$Support = \frac{Number\ of\ records\ with\ A\ and\ B}{Total\ number\ of\ records},$$

where the numerator indicates the number of records with *A* and *B* both true.

$$Confidence = \frac{Number\ of\ records\ with\ A\ and\ B}{Total\ number\ of\ records\ with\ A}$$

The confidence can also be written as

$$Confidence = \frac{P(A \cap B)}{P(A)},$$

where $P(A \cap B)$ is the probability of $A \cap B$.

In our context, the classification process $P(A \cap B)$ forms the probability of the occurrence of feature values with a given class. Here *A* indicates a feature value vector, and *B* indicates a class.

However,

$$P(A \cap B) = P(A \mid B)P(B).$$

If we observe that

$$Confidence = \frac{P(feature\ set \mid class)P(class)}{P(feature\ set)},$$

we note that the confidence can be used to form basic beliefs of the decision tree classifier.

3.2 Computing uncertainty for the classifiers

The closer the values of beliefs for K classes to each other, the more uncertain the classifier is about its decision. As the beliefs start spreading apart, the uncertainty starts to decrease.

The idea behind the uncertainty evaluation is if the number of classes is *K*, then the distance between the belief value and the value 1/*K* is evaluated. The ambiguity involved in the classifying decision is higher if two classes show very similar beliefs.

Uncertainty is computed using the formula:

$$H(U) = 1 - \frac{K}{K-1}\sum_{i=1}^{K}(m(i) - \frac{1}{K})^2$$

The belief is then calculated as

$$Bel(i) = \alpha\, m(i)$$

and the uncertainty is expressed as

$$Bel(\theta) = \beta\, H(U).$$

Previous studies [7] have shown that the predictions of the Bayesian classifier are biased towards the class having larger prior sample probability. In these cases, the uncertainty has to be calculated using a different expression.

Let p_1 be the prior sample probability of the class with maximum belief and p_2 be the prior sample probability of the class with the second highest belief.

If $p_1 > p_2$, then the uncertainty is evaluated using the following expression:

$$H(U) = 0.35\ e^{\frac{p_1}{p_2}}$$

The value 0.35 is the ratio of failures when highest belief has more prior class probability than the next highest belief to the total number of failures. This ratio indicates the percentage of uncertainty introduced in Bayesian classification due to such typical cases.

3.3 Combining evidence

The last step consists in combining the belief and the uncertainty obtained in the previous steps, such as to arrive at the final decision.

Let the combined belief mass be assigned to C_k, where C_k is a set of all subsets produced by A∩B. The mathematical representation of the combination rule is as follows:

$$Bel\ (C_k) = \frac{\sum_{Ai \cap Bi = Ck\ ;Ck \neq \varnothing} Bel\ (A_i) \times Bel\ (B_i)}{1 - \sum_{Aj \cap Bj = \varnothing} Bel\ (A_j) \times Bel\ (B_j)}$$

In our implementation, the Bayesian classifier is first combined with the kNN classifier, then the resulting classifier is combined with the decision tree learner. The combination takes advantage of the fact that one classifier may be more accurate in handling records corresponding to a certain class than the other.

Thus after combination the overall classification becomes more accurate.

4 Experimental work

Based on the model we proposed in section 3, we have implemented a new system that is intended to provide a reference accuracy value in choosing a classifier for any specific dataset especially if the dataset has not been previously tested.

We have implemented the system using as framework Weka [6].

The system has been tested on four datasets having different features and from different domains. The chosen datasets represent well known benchmarks and are available at the UCI Machine Learning Repository.

The implementation considers three different categories of classifiers. The decision tree learner was represented by the implementation of the C4.5 algorithm found in Weka, J4.8. For the kNN classifier, *k* was set to three, as in [7].

Testing was carried out on all the datasets for each of the three classifiers separately, then the combined classifier was tested on the datasets. The datasets have been randomly split into train set and test set, in a proportion of 80% for training and 20% for testing.

For a better validation of the system, comparisons with ensemble learning methods have been carried out. The algorithms tested are AdaBoost and bagging in combination with the three classifiers involved in the evaluation.

To ensure the accuracy of the test results, we have performed runs on 100 different splits of a dataset for each classifier and the averaging was then computed.

The combined classifier was expected to show a better accuracy than the average of the three classifiers separately. Also due to the theoretical advantages of the approach, the combined classifier was expected to show robustness, despite any possible single classifier accuracy gaps between different datasets.

The datasets are presented in Table 1.

Table1: Dataset features

Dataset	Number of instances	Number of attributes	Number of classes
Cars	1728	6	4
Cleveland	303	13	5
Pima	768	8	2
Wisconsin	699	9	2

The test results obtained by the three classifiers are presented in Table 2. The individual classifiers are not stable with respect to the datasets. While Naïve Bayes seems to classify the instances in the Wisconsin dataset most accurately, it performs poorly on the Cars dataset, where it produces the lowest accuracy among the three classifiers. Similar remarks can be done for the other two classifiers.

Table 2: Individual classifiers accuracy rates

Dataset	Bayes	kNN	J4.8
Cars	85.43%	92.30%	91.50%
Cleveland	55.73%	56.91%	52.60%
Pima	75.44%	73.38%	73.88%
Wisconsin	96.24%	95.35%	94.41%

Another remark is related to the fact that the differences in accuracy between the three classifiers are high especially in the case of Cars datasets, which proves the high value of the risk involved in choosing a certain classifier for a certain dataset.

Table 3: Comparison between the accuracy of the combined classifier and the average accuracy of the three classifiers

Dataset	Average	DST combined classifier
Cars	89.74%	91.55%
Cleveland	55.08%	55.81%
Pima	74.23%	74.85%
Wisconsin	95.33%	96.16%

As Table 3 shows, the combined classifier is more accurate than the average of the three classifiers. Even if there is a classifier that performs better than the combined classifier on a certain dataset, the same classifier will perform poorly on other datasets. For example, in the case of the Wisconsin dataset, the Bayesian classifier yields highest accuracy, while on the Cars dataset it achieves the poorest performance among the three classifiers. These results are in strong connection with the "No Free Lunch" theorem.

Results obtained with bagging and boosting for the three classifiers are shown in tables 4 and 5, respectively.

Table 4: Accuracy rates for bagging

Dataset	Bagging +Bayes	Bagging +kNN	Bagging +J4.8
Cars	85.14%	93.10%	92.71%
Cleveland	55.91%	58.01%	54.26%
Pima	75.38%	73.48%	75.11%
Wisconsin	97.40%	95.61%	95.33%

Table 5: Accuracy rates for boosting

Dataset	Boosting +Bayes	Boosting +kNN	Boosting +J4.8
Cars	90.35%	92.30%	95.21%
Cleveland	55.73%	53.55%	53.18%
Pima	75.61%	73.33%	72.31%
Wisconsin	95.68%	95.23%	96.24%

Table 6: Comparison between the accuracy of the combined classifier and the accuracy of ensemble learning methods

Dataset	Bagging	Boosting	Combined classifier
Cars	90.32%	92.62%	91.55%
Cleveland	56.06%	54.15%	55.81%
Pima	74.66%	73.75%	74.85%
Wisconsin	96.11%	95.72%	96.16%

It can be observed that even though the ensemble learning methods improve the accuracy, the problem of differences between the classifiers' predictions on a dataset still persists, especially in the case of bagging. The same observation applies for the problem of a classifier being the best predictor in one case and the worst on another dataset.

A comparison with bagging and boosting has been conducted, and the results are shown in Table 6:

The combined classifier outperforms Bagging on three datasets and Boosting other three datasets, being slightly less accurate than one of the three classifiers on one dataset. This proves yet again the risk minimization displayed by the combined classifier.

Starting from the results obtained we can formulate that the new system can be used to establish the baseline accuracy for a certain dataset, and help evaluate how well a specific classifier performs on that problem.

The following chart on Wisconsin datasets illustrates the idea of reference system:

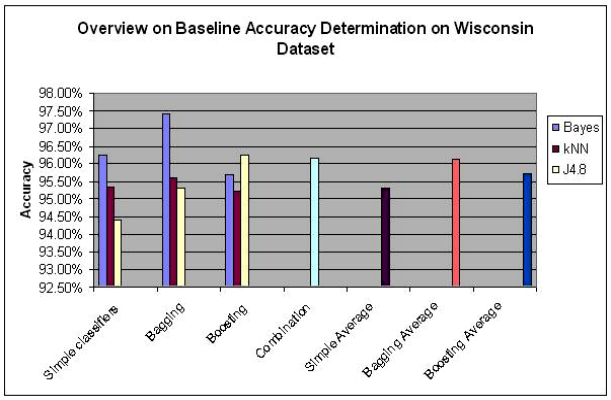

Figure 1: Overview on Baseline Accuracy Determination on Wisconsin Dataset

As the chart shows, among the 12 classifiers and average accuracy considered, only two can outperform the combined classifier: Bagging with Bayesian classifier and Boosting with Decision Tree classifier and the improvement is not always significant enough. Moreover, there is no guarantee that the same combination will perform equally well on another dataset, and the tests have proven exactly this –and the figure below illustrates the idea:

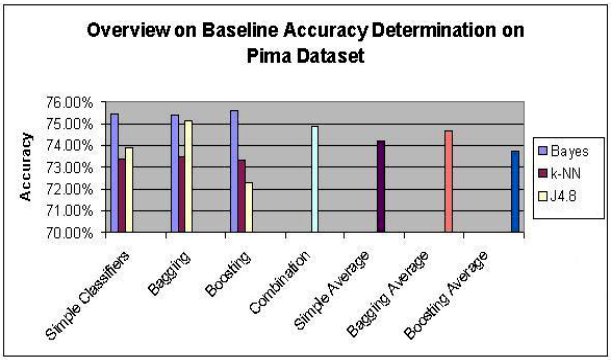

Figure 2: Overview on Baseline Accuracy Determination on Pima Dataset

From here the necessity of a reference high accuracy, a system that is stable with respect to any dataset, and the tested system has proven to be exactly that.

5 Conclusions

This paper presents a new system based on the classifier fusion technique proposed by the Dempster-Shafer theory. The starting point was represented by existing problems exhibited by individual classifiers when applied to different datasets. According to the "No Free Lunch" theorem, some classifiers may perform better on a certain dataset, but obtain disastrous results on a different dataset. Additional problems related to single classifiers include variance and bias.

In order to address the problems related to single learning schemes, methods of combination have been introduced in the machine learning community, the most prominent being ensemble approaches and classifier fusion. The latter was preferred due to its ability to offer a robust behaviour over several different datasets. This is a direct consequence of its capability of considering data coming from different sources. Another advantage of the Dempster-Shafer theory is that it considers uncertainty.

The new system was evaluated on several classical benchmarks, and its performance was compared to that of its component classifiers. We expected that the system perform better than the average of the single classifiers involved in the combination. In addition, comparative evaluations have been carried out with ensemble learning methods (bagging and boosting) in order to emphasize the advantages of the chosen methodology.

The results obtained have confirmed our expectations, the system's accuracy on all datasets being higher than the average of the three individual classifiers. Also, better accuracy has been observed in most of the cases when the system was compared with ensemble learning methods. Starting from the results obtained we can formulate that the new system can be used to establish the baseline accuracy for a certain problem, and help evaluate how well a specific classifier performs on that problem.

As further work we are considering testing the system with different data sources as inputs for the three classifier components. This would be possible due to the data fusion approach.

References

[1] L. Breiman, “Bagging predictors”, Machine Learning, 24, 1996, pp. 123-140.

[2] Y. Freund, R. E. Schapire, “Experiments with a New Boosting Algorithm”, International Conference on Machine Learning, 1996.

[3] L. Kuncheva, L. C. Jain, “Designing classifier fusion systems by genetic algorithms”, IEEE Trans. on Evolutionary Computation, 2000, pp. 327-336.

[4] G. Giacinto, F. Roli, L. Didaci, “Fusion of multiple classifiers for intrusion detection in computer networks”, Pattern Recognition Letters, 2003, pp. 1795 -1803.

[5] L. Roux, J. Desachy, “Information fusion for supervised classification in a satellite image”, Fourth IEEE International Conference on Fuzzy Systems, Yokohama, Japan, 1995, pp. 1119-1124.

[6] Witten, I., Frank, E., *Data Mining: Practical machine learning tools and techniques*, 2nd ed. Morgan Kaufmann, 2005.

[7] G Mahajani, Y Aslandogan, “Evidence Combination in Medical Data Mining”,University of Texas, Arlington, USA, 2003.

[8] G. Carpenter , S. Grossberg , J. Reynolds, “ARTMAP: Supervised Real-Time Learning and Classification of Nonstationary Data by a Self-Organizing Neural Network”, *Neural Networks,* vol. 4, 1991, pp. 565-588.

[9] R. Polikar, D. Parikh, S. Mandayam, “Multiple Classifier Systems for Multisensor Data Fusion”, IEEE Sensors Applications Symposium, Houston, Texas, USA, 2006.

[10] Shafer, G., *A Mathematical Theory of Evidence*, Princeton University Press, 1976.