

Using an Ensemble of Classifiers to Audit a Production Classifier

Piero Bonissone, Neil Eklund, and Kai Goebel

GE Global Research, One Research Circle, Niskayuna, NY 12309, USA
{bonissone, eklund, goebelk}@research.ge.com

Abstract. After deploying a classifier in production it is essential to support its lifecycle. This paper describes the application of an ensemble of classifiers to support two stages of the lifecycle of an on-line classifier used to underwrite life insurance applications: the *monitoring* of its decisions quality and the *updating* of the production classifier over time. All combinations of five classification methods and seven fusion methods were assessed from the perspective of accuracy and pairwise diversity of the classifiers, and accuracy, precision, and coverage of the fused classifiers. The proposed architecture consists of three off-line classifiers and a fusion module.

1 Introduction

The automation of a decision-making processes requires addressing each step in the lifecycle of the underlying decision engine. The development and deployment of such engine represent the first stage of its lifecycle. Once an engine has been placed in production, it is equally important to monitor its performance and probe the quality of its decisions. In previous papers we have described the design and optimization of two decision engines for underwriting insurance applications. Both engines, based on fuzzy constraints and fuzzy case-based reasoning [1,4], used an evolutionary algorithm to optimize their underlying parameters and minimize the cost of misclassification [6]. The core of this optimization was the generation of a set of Standard Reference Decisions (SRD). This set represents the ground truth against which every classifier is evaluated. In [4,18] we discussed the lifecycle of a classifier with an emphasis on its validation, verification, and knowledge-base maintenance. In this paper we focus on the final design and its validation with production data.

As discussed in [5], the design of a successful classifier fusion system consists of two important parts: design of the individual classifiers, selection of a set of classifiers [13, 21], and design of the classifier fusion mechanism [20]. Key to effective classifier fusion is the diversity of the individual classifiers. Strategies for boosting diversity include: 1) using different types of classifiers; 2) training individual classifiers with different data set (bagging and boosting); and 3) using different subsets of features. In our approach we follow the first and third strategies directly, and capitalize indirectly on the second strategy by employing the random forest classification method.

2 The Underwriting Problem and the Production Classifier

Insurance underwriting (UW) is a complex decision making task traditionally performed by individuals. UW can be formulated as a classical classification problem, consisting in assigning a given insurance application, described by its medical and demographic records, to one of a small set of rate classes. We define an insurance application as an input vector \bar{X} containing discrete, continuous, and nominal (attribute) variables. These variables represent the applicant's medical and demographic information that has been identified by actuarial studies to be pertinent to the estimation of the applicant's claim risk. Similarly, we define the output space \bar{Y} , e.g. the underwriting decision space, as an ordered list of rate classes. Due to the intrinsic difficulty of representing risk as a real number on a scale, the output space \bar{Y} is subdivided into rate classes containing similar risks. The underwriting process can be summarized as a discrete classifier that maps an input vector \bar{X} into a decision space \bar{Y} , where: $|\bar{X}| = n$ and $|\bar{Y}| = T$.

The automation of this decision making process has strong accuracy, coverage and transparency requirements. The production classifier must satisfy several constraints: a) *high classification accuracy* in spite of highly non-linear boundaries of the rate classes; b) *consistency* in interpretation of actuarial guidelines; c) intrinsic *flexibility* to ensure a balance between *risk-tolerance*, necessary to maintain price competitiveness, and *risk-avoidance*, necessary to prevent overexposure to risk; d) *transparent* and *interpretable* decisions, to satisfy legal and compliance regulations.

Consequently, we face two main design tradeoffs: 1) *Accuracy versus coverage* - requiring low misclassification rates for high volume of applications; 2) *Accuracy versus interpretability* - requiring a transparent, traceable decision-making process. A fuzzy logic engine (FLE) was deployed for production. This classifier uses fuzzy rule sets to encode best underwriting standards. Each rule set represents a set of fuzzy constraints defining the boundaries between rate classes. These constraints were initialized from underwriting guidelines, refined through interviews with expert underwriters, and tuned by evolutionary algorithms. The goal of the FLE is to assign an applicant to the most competitive rate class, providing that the applicant's vital data meet all the constraints of that particular rate class to at least a minimum degree of satisfaction. The minimum degree of satisfaction of all relevant constraints determines the confidence measure in the decision. The FLE is described in [3] and [5].

3 Classification Ensembles and Fusion Approach Selection

The ensemble classification system was developed in three stages. First, five candidate classification methods were trained. Second, the diversity of the classifiers was assessed. Finally, the decision accuracy of all combinations of candidate classification methods under seven fusion methods plus single classifiers were evaluated using a leave-one-out approach. Although systems were developed for both smokers and non-smokers, only the results for non-smokers (for which there are both more cases and more rate classes) are presented here.

As discussed in [3], we created an indicator to encode the result of applying underwriters' domain knowledge. This indicator, referred to as TAG, defines the best available rate class for each applicant based on a set of hard-coded rules representing insurance standard policies. The use of this aggregated domain knowledge boosted most classifiers' performance, leading to an accuracy improvement of about 1-2% on average [3]. Moreover, it allowed us to drop nine of the 19 features, and use 10 features plus the indicator for training (except where noted in the random forest section).

3.1 Candidate Classification Methods

Feed-forward Neural Network Classifier Ensemble. An independent 12 input nodes, 5 hidden nodes, and 1 output node artificial neural network (NN) was trained for each class. For each network, the data was labeled one for the corresponding class, or zero for any of the other four classes. To classify an unknown case, each network evaluates the case, and the case is assigned the class corresponding to the network with the highest value of the activation function. This approach (rather than using a single NN with five output nodes) decomposes the complexity of the classification problem and reduces the overall training time. Activation functions for both hidden and output neurons are logistic sigmoid functions. The range of target values was scaled to [0.1 0.9] to prevent saturation during training process.

Multivariate Adaptive Regression Splines Classifier Ensemble. Multivariate Adaptive Regression Splines (MARS) [10] is an adaptive nonparametric regression technique, able to capture main and interaction effects in a hierarchical manner. Being a piecewise-linear adaptive regression procedure, MARS can approximate very well any non-linear structure, if present. However, global models cannot easily incorporate jumps in decision boundaries of a large number of variables in an extremely small bounded range. Two approaches were used to address this problem. First, the use of the TAG variable helps the MARS search algorithm in initializing spline knots in the right place. Second, we developed a *Parallel Network* arrangement of models. We created a collection of MARS models, each of which solves a two-class problem, and we collated their outputs in a manner similar to the one used for the NN classifiers.

Support Vector Machines Ensemble. Support Vector Machines (SVM) [22] are learning machines that non-linearly map an input feature space into a higher dimensional feature space. A linear classifier is then constructed in the higher dimensional feature space. Five SVM models (one for each class) were trained, and their outputs were resolved in a manner similar to the one used for the NN classifiers. The shape parameter for the radial basis function Kernel Gamma and the parameter for cost of constrain violation C were both set to 3. The overall SVM classification was resolved in the same as the NN classifiers.

Random Forests. Random Forests (RF) [8] is a classification method that applies bagging [7] to a variation of classification trees [9]. A standard classification tree is constructed by splitting the data on the best feature of all possible features at each node. For RF, only a randomly selected subset (chosen always from the full set) of

features are eligible to split each node. Moreover, in contrast to standard classification trees, the individual RF trees are not pruned; rather they are grown to 100% node purity. Although typically hundreds of trees are developed, RF's are very quick to train (e.g., much faster than neural networks for a given data set and computer). Within our application, the performance of RF was superior to that of the other classification algorithms, including NN and SVM. While RF's contained 500+ trees, NN's and SVM's were much smaller ensembles, containing only five binary classifiers each.

Two RF classifiers were developed. The first RF was trained on the regular set of features (including TAG). The second (referred to as RFA) was trained on a larger set of features, comprised of the regular set (*excluding* TAG) and the features used to create the TAG variable. Although the performance of the RFA suffered slightly (though much less than any other classification method tried on the RFA dataset), the results were not surprisingly quite diverse from the other classifiers (see below). The results presented here for both RF and RFA are based on 1000 trees per forest, and six variables eligible to be split at each node.

3.3 Classification Accuracy

The single classifier classification accuracy was evaluated using five fold cross validation (using the same folds for each method). Table 1 shows the performance of the individual classifiers and the production classifier (FLE) as expressed by the true positive rate. While their accuracies are roughly comparables, their pairwise diversities are not, as shown in the next subsection.

Table 1. Five fold cross validation classification accuracy

Method	Accuracy	Method	Accuracy
MARS	92.71%	RF	93.30%
NN	92.87%	RFA	91.26%
SVM	92.23%	FLE	93.41%

3.3 Classifier Diversity

We assessed pairwise classifier diversity using four measures described in [14], Q , ρ , $disagreement$, and $double\ fault$. Results were comparable across all four measures. Only results for Yule's Q are presented here because of its more transparent interpretation.

Q ranges from -1 to 1 . For statistically independent classifiers, Q is 0 . Correlated classifiers have positive values of Q , while uncorrelated classifiers (i.e., classifiers that make mistakes on different cases) have negative values of Q . Q is only used for two classifiers at a time (Table 2) and is calculated

$$Q = \frac{ad - bc}{ad + bc} . \quad (1)$$

The values of the Q statistic for all five classifiers and the production classifier (fuzzy logic engine or FLE) are reported in Table 3, and reveal several interesting things about the classifiers for these data. First, all classification methods that use the TAG variable are highly correlated (i.e., they tend to misclassify the same cases). Both MARS and SVM show a moderate degree of positive correlation with the FLE, while the RFA is the only classifier to have a negative correlation with the FLE.

Table 2. Probabilities for two classifiers, C1 and C2. Note that $a + b + c + d = 1$.

	C1 correct	C1 wrong
C2 correct	<i>a</i>	<i>b</i>
C2 wrong	<i>c</i>	<i>d</i>

Table 3. Q statistic for all five classifiers and the PC

	FLE	MARS	NN	SVM	RF	RFA
FLE	1.000	0.412	0.225	0.416	0.383	-0.308
MARS	0.412	1.000	0.873	0.915	0.925	0.021
NN	0.225	0.873	1.000	0.937	0.776	-0.142
SVM	0.416	0.915	0.937	1.000	0.820	-0.059
RF	0.383	0.925	0.776	0.820	1.000	0.411
RFA	-0.308	0.021	-0.142	-0.059	0.411	1.000

3.4 Fusion Methods

Seven fusion methods were evaluated using each of the 26 unique combinations of two or more of the five classifiers (excluding the FLE), along with the performance of each single classifier method. The fusion methods used are described below.

Majority Vote. Each classifier has one vote and the case is assigned the winning class. Ties are broken randomly.

Averaging. The normalized output of each classifier is averaged for each class, and the case is assigned the class with the highest average. Ties are broken randomly.

Borda Count. The case is assigned to the class with the maximum sum of the rank of the negative class weights (within each classifier). Ties are broken randomly.

N of All. Under the N of All (NOA) fusion scheme, if the number of classifiers voting for a particular class is greater than N (where N is > 1 and \leq the number of classifiers), a case is assigned that class; otherwise, the case is assigned “no decision”.

Behavior Knowledge Space (BKS). This fusion method [12] treats every possible combination of output from different classifiers as a cell in a lookup table (the BKS table). During training, training samples associated with a particular call are partitioned by actual class and the most representative class (the majority class) is selected

for each cell. This equates in essence to setting up the classifier output probability distribution. For test patterns, the classification is accomplished using the class label of the BKS cell indexed by the classifiers output. In our implementation, the BKS assigns “no decision” to novel patterns.

Naïve Bayes. The naïve Bayes (NB) fusion approach makes the assumption that the decisions of the individual classifiers are independent. While this assumption is almost certainly invalid, this approach works quite well in practice [15]. Using n features f (individual classifier decisions) that pick a particular class c of a set of classes C , the NB decision rule is

$$\text{classify}(f_1, \dots, f_n) = \operatorname{argmax}_c p(C = c) \prod_{i=1}^n p(F_i = f_i | C = c) \cdot \quad (2)$$

Meta-SVM. The Meta-SVM (MSVM) uses the normalized output of each classifier as a feature space to train a new SVM classifier. The output of the MSVM classifier assigns the five classes to some region of one dimension; thus, there is a further problem of dividing that dimension into five separate regions. To automate this process, a classification tree [9] was trained on the MSVM output, and the smallest tree with exactly five leaf classes was used to determine the class boundaries. The use of a classification tree to automate determining class boundaries is both faster and more accurate than hand tuning.

3.5 Fusion Accuracy

Of the 1866 cases, there were 188 where the result of at least one combination of fusion method and classifier method (FMC) differed from the ground truth, i.e., the standard reference decision (SRD) set. The leave-one-out accuracy (LA) for each FMC is plotted in the lower set of axes in Figure 1 (for the 188 cases). The upper set of axes is on a different scale, showing only the best performing combinations, and revealing there are three FMCs that have the same maximum accuracy. FMC details, Precision [$TP/(TP+FP)$] and Recall [$TP/(TP+FN)$] scores (with respect to the FLE) of the three most accurate FMCs are listed in Table 4.

Table 4. Classification method, fusion method, precision and recall scores of the three most accurate FMCs. One represents inclusion of a classification method, zero exclusion.

MARS	NN	RFA	RF	SVM	Method	Precision	Recall
0	1	1	1	0	Borda count	55.8%	71.7%
0	1	1	1	0	Majority vote	55.6%	75.0%
1	1	1	1	0	Naïve Bayes	55.6%	75.0%

Across all combinations of classification method, majority vote, averaging, Borda count, and NB do better than individual classifiers. However, NOA, BKS and MSVM do on average worse than a single classifier. The shortcomings of NOA and BKS arise from a common explanation: both methods allow “no decision”. While this option may be a desirable behavior in many circumstances, nonetheless it does not count

toward total correct decisions. The failure of MSVM is probably related to the size of the feature space: MSVM is the only classifier trained on the raw output of the other classifiers. This suggests that some intelligent preprocessing of the data beforehand might have improved performance.

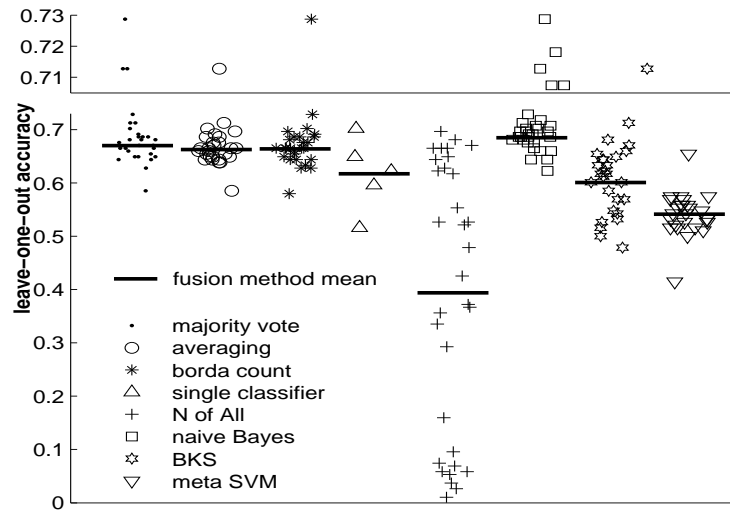


Figure 1. Fraction of correct calls for each FMCM. Note the upper set of axes is on a different scale to highlight detail. Each point represents a unique combination of classification methods.

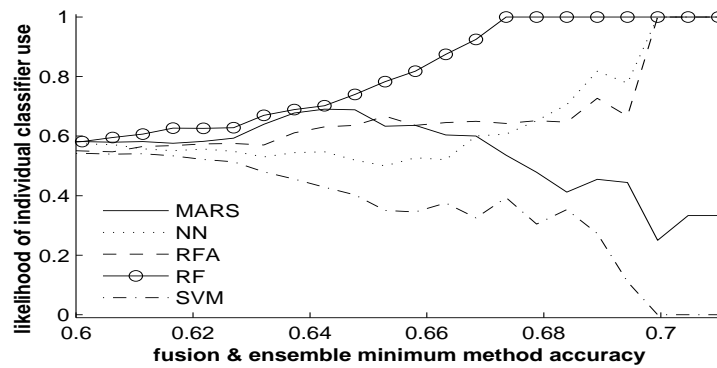


Figure 2. The likelihood that a classifier is employed in a fusion method and classification method combination of a given or greater accuracy.

The likelihood that a classifier of a given or greater accuracy is included in a FMCM (e.g., out of all of the fusion method and classification method combinations with an accuracy ≥ 0.68 , about 41% used MARS, 65% used NN, etc.) is plotted in Figure 2. SVM is particularly unlikely to appear in highly accurate classifiers, while RF is particularly likely (occurring in all of the 28 most accurate FMCMs).

We speculate the RF is included in the most accurate FMCMs because it is the single most accurate individual classifier. The RFA has the worst single classifier accuracy, but it is the least correlated with other classification methods (Table 3) and somewhat uncorrelated with the FCM (which is important in catching errors in the FCM), so its diversity outweighs its poor performance. Of the remaining three classifiers, SVM and MARS are highly correlated with RF (Table 3); NN is both the least correlated and most accurate of the three, so is employed at high accuracy.

4 Quality Assurance Architecture

Based on the results above, the NN, RFA, RF ensemble using the majority vote fusion scheme was adopted to monitor the performance of the FLE and assure the quality of the production engine decisions. In addition, this fusion will identify the best cases that could be used to tune the production engine in future releases, as well as controversial or unusual cases that should be highlighted for manual audit by senior underwriters, as part of the Quality Assurance system. Figure 3 is a diagram of the system.

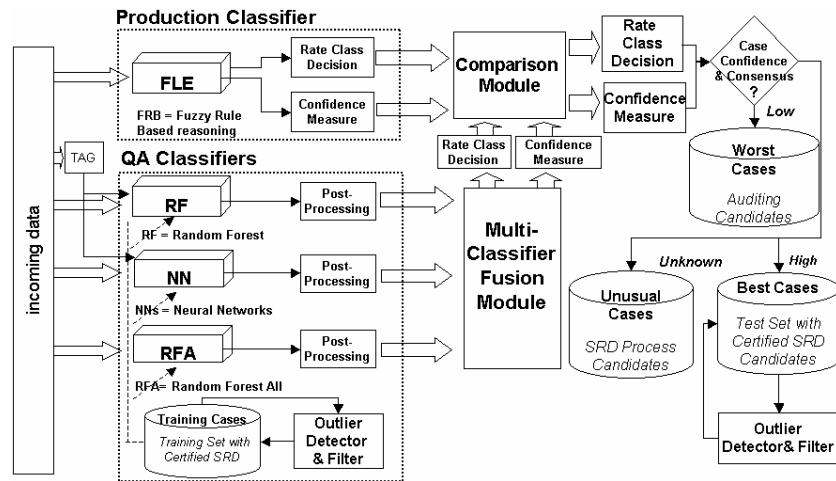


Figure 3. The Production classifier and QA system.

Initially, each classifier in the system was forced to commit to a particular rate class. The results reported in this paper reflect this constraint. If preferred, we can modify this specific tradeoff of coverage versus accuracy by implementing post-processing filters prior to the fusion process. We can treat each classifier's output as a discrete membership distribution over the rate classes considered, and compute four features to summarize such memberships: cardinality, entropy, difference and rank order separation between the highest and the second highest values of the outputs. Then we can impose a set of thresholds (lower or upper bounds) on these features to identify the cases with weak decisions. For such cases, we change their final conclusions to "un-

known”. In our experiments, the values of the threshold were obtained using local search, looking for different tradeoffs with better accuracy at the expense of coverage.

4.1 Validation Results using Production Data

The QA system was validated using 3292 cases, of which 393 showed disagreement between the FLE and the classifier fusion. Of those cases, 131 were randomly selected for evaluation by a human underwriter. Of the 131 cases, the FLE was correct in 77 of them, i.e., false positives (in the sense that the QA classifier incorrectly identified them as being incorrectly classified by the FLE). The fusion was right in 43 cases, implying a correction to the automated FLE. Neither was correct in 11 cases, which is still a good call by the QA system insofar as it will entail a correction of the FLE decision by a human underwriter. Note that this analysis gives no insight into false negatives (false negatives is defined here as both the fusion and the FLE are wrong).

The precision of the QA system was 44.2%, down over 11 percentage points from the training data. However, the distribution of the validation data was substantially different from the training data (Figure 4). In light of the dramatic distribution changes from training to production, the QA architecture seems quite robust, and provides excellent guidance for the auditor (by flagging disagreement with the FLE).

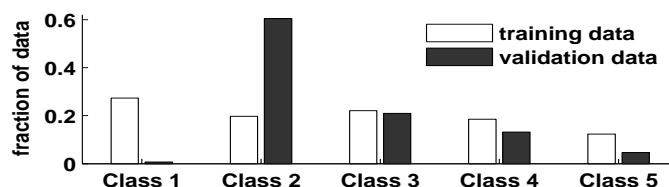


Figure 4. Class distribution of training and validation data.

5 Conclusions

We applied the fusion to the quality assurance (QA) problem for automated underwriting. All combinations of five classification methods and seven fusion methods (plus single classifier) were assessed from the perspective of accuracy and pairwise diversity of the classifiers, and accuracy, precision, and coverage of the fused classifiers. The final classifier ensemble and fusion method performed well, despite considerable differences in the distribution of the training and validation data. This QA system can be used to monitor the performance of the automated decision making system, identifying cases that might be suspect, and should be examined by a human. Moreover, these cases can be incorporated into the standard reference decision set, to further tune the performance of both the automated decision maker (the FLE) and the QA system.

One of the most interesting results is the insight into the tradeoff between accuracy and diversity (Figure 2). We intend to exploit this behavior and plan some future experiments to further explore this tradeoff.

References

- [1] Aggour, K., Pavese M., Bonissone, P. and Cheetham W. SOFT-CBR: A Self-Optimizing Fuzzy Tool for Case-Based Reasoning, *5th Int. Conf. on Case-Based Reasoning (ICCBR)*, pp. 5-19, Springer-Verlag, Trondheim, Norway, 2003.
- [2] Bonissone, P. The life cycle of a Fuzzy Knowledge-based Classifier, *North American Fuzzy Information Processing Society (NAFIPS 2003)*, pp. 488-494, Chicago, IL, Aug. 2003.
- [3] Bonissone, P. Automating the Quality Assurance of an On-line Knowledge-Based Classifier By Fusing Multiple Off-line Classifiers, *Proc. IPMU 2004*, 309-316, Perugia, Italy, 2004.
- [4] Bonissone, P. and Cheetham W. Fuzzy Case-based Reasoning for Decision Making, *IEEE Int. Conf. on Fuzzy Systems*, pp. 995-998, Melbourne, Australia, 2001.
- [5] Bonissone, P., Goebel, K., and Yan, W. Classifier Fusion using Triangular Norms, *Proc. 2004 Multi Classifier Systems (MCS'04)*, pp. 154-163, Cagliari, Italy, 2004.
- [6] Bonissone, P., Subbu, R., and Aggour, K. Evolutionary Optimization of Fuzzy Decision Systems for Automated Insurance Underwriting, *IEEE Int. Conf. on Fuzzy Systems (FUZZ-IEEE '02)*, pp 1003-1008, Honolulu, Hawaii, USA, 2002.
- [7] Breiman, L. Bagging predictors. *Machine Learning*, 24(2), 123-140, 1996.
- [8] Breiman, L. Random forests. *Machine Learning*, 45(1), 5-32, 2001.
- [9] Breiman, L., Friedman, J., Olshen, R. and Stone, C. *Classification and Regression Trees*. Wadsworth, Belmont, CA, 1984.
- [10] Friedman, J. Multivariate Adaptive Regression Splines. *Annals of Statistics*, 19: 1-141, 1991.
- [11] Ho, T., Hull, J., and Srihari, S. Decision Combination in Multiple Classifier Systems, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 16, No.1, pp.66-75, 1994.
- [12] Huang, Y. and Suen, C. A method of combining multiple experts for the recognition of unconstrained handwritten numerals. In *Trans. IEEE Pattern Analysis and Machine Intelligence* 17(1), pages 90-94, 1995.
- [13] Kuncheva L. Switching between selection and fusion in combining classifiers: An experiment, *IEEE Transactions on SMC, Part B*, 32 (2), 2002, 146-156.
- [14] Kuncheva, L., and Whitaker C. Ten measures of diversity in classifier ensembles: Limits for two classifiers, *Proceedings of IEE Workshop on Intelligent Sensor Processing*, Birmingham, February, 2001, 10/1-10/6, 2001.
- [15] Langley, P., Iba, W., and Thomson, K. An analysis of Bayesian classifiers. *Proceeding of National Conference on Artificial Intelligence (AAAI-92)*, pp. 223-228, 1992.
- [16] Niyogi, P Pierrot, J-B, and Siohan.O. On decorrelating classifiers and combining them, MIT AI Lab., September 2001
- [17] Partridge, D., Yates, W. Engineering multiversion neural-net systems. *Neural Computation*, 8:869-893, 1996.
- [18] Patterson, A., Bonissone, P. and Pavese, M. Six Sigma Quality Applied Throughout the Lifecycle of an Automated Decision System, *Journal of Quality and Reliability International* (2005, to appear).
- [19] Petrakos, M., Kannelopoulos, I., Benediktsson, J., and Pesaresi, M. The Effect of Correlation on the Accuracy of the Combined Classifier in Decision Level Fusion, *Proceedings of IEEE 2000 International Geo-science and Remote Sensing Symposium*, Vol. 6, 2000.
- [20] Roli, F., Giacinto, G., and Vernazza, G. Methods for Designing Multiple Classifier Systems, *MCS 200, LNCS 2096*, 78-87, 2001
- [21] Tumer, K.& Ghosh, J. Error correlation and error reduction in ensemble classifiers, *Connection Science*, 8:385-404, 1996.
- [22] Vapnik, V.N., *The Nature of Statistical Learning Theory*, Springer-Verlag, New York, 1995.