# Using Random Forests for Handwritten Digit Recognition

Simon Bernard, Laurent Heutte, Sébastien Adam

# Using Random Forests for Handwritten Digit Recognition

Simon Bernard, Laurent Heutte, Sébastien Adam
Laboratoire LITIS EA 4108
UFR des Sciences, Université de Rouen, France.
{simon.bernard,laurent.heutte,sebastien.adam}@univ-rouen.fr

## Abstract

*In the Pattern Recognition field, growing interest has been shown in recent years for Multiple Classifier Systems and particularly for Bagging, Boosting and Random Subspaces. Those methods aim at inducing an ensemble of classifiers by producing diversity at different levels. Following this principle, Breiman has introduced in 2001 another family of methods called Random Forest. Our work aims at studying those methods in a strictly pragmatic approach, in order to provide rules on parameter settings for practitioners. For that purpose we have experimented the Forest-RI algorithm, considered as the Random Forest reference method, on the MNIST handwritten digits database. In this paper, we describe Random Forest principles and review some methods proposed in the literature. We present next our experimental protocol and results. We finally draw some conclusions on Random Forest global behavior according to their parameter tuning.*

## 1. Introduction

Machine Learning issues are concerned by several learning approaches aiming at building high performance classification systems, with respect to a set of data. One of them, arousing growing interest in recent years, deals with combining classifiers to build Multiple Classifier Systems (MCS) also known as Classifier Ensembles.

MCS attempt to take into account complementarity between several classifiers in order to improve reliability in comparison with individual classifier models. The hope is that aggregating several classifiers will allow to bring the resultant combining classifier closer to the optimal classifier thanks to the diversity property, which is nowadays recognized as one of the characteristics required to achieve those improvements [11].

In [11] Kuncheva presents four approaches aiming at building ensembles of diverse classifiers :

1. The combination level : Design different combiners
2. The classifier level : Use different base classifiers
3. The feature level : Use different feature subsets
4. The data level : Use different data subsets

Those two latter categories has proven to be extremely successful owing to the Bagging, the Boosting or the Random Subspaces methods [2, 8, 10, 11, 7].

The main idea of the Boosting is to iteratively build an ensemble of base classifiers, each one being a ”boosted” version of its predecessors [8]. In other words, a ”classical” classifier is progressively specialized, by increasingly paying more attention to misclassified instances. All the classifiers obtained at each iteration are finally combined to participate in the same MCS.

The Bagging technique which was introduced by Breiman as an acronym for Bootstrap Aggregating [2], consists in building an ensemble of base classifiers, each one trained on a bootstrap replicate of the training set. Predictions are then obtained by combining outputs with plurality or majority vote.

The Random Subspace principle leans on producing diversity by using randomization in a feature subset selection process [10]. For each base classifier, a feature subset is randomly selected among all the original inputs. All samples are projected to this subspace and the classifier is then trained from those new representations.

Few years later, Breiman proposed a family of methods based on a combination of those principles, called Random Forest [3] (RF). It consists in a general MCS building method using Decision Trees as base classifiers. The particularity of this ensemble is that each of them has to be built from a set of random parameters. The main idea is that this randomization introduces more diversity into the base classifiers ensemble. The definition given by Breiman in this paper is deliberately generic enough to let this randomization be introduced anywhere in the process. Therefore a RF could be built by sampling the feature set (like in Random Subspace principle), the data set (like in Bagging principle), and/or just varying randomly some parameters of the trees.

Since it has been introduced in 2001, RF have been focused on and studied by a lot of researchers. They have also been compared to the other main ensemble methods, as the two previously mentioned Bagging and Boosting. In most of those works, RF are said to be competitive with Boosting – known as one of the most efficient [3, 11]. However, though lots of parameters can be tuned for using RF, it does not exist any practical study in the literature that examines more deeply the influence of parameter choices on its performance.

In this paper we propose a preliminary work to study the Random Forest mechanism in a pragmatic way, by taking a practitioner point of view. Our aim is not to search for best intrinsic performance but rather to analyze the global behavior of this family of methods with respect to their parameter settings. For that purpose we have investigated one variant of RF called Forest-RI [3] to the recognition of handwritten digits from the well known MNIST database [12].

This paper is divided into three main parts. In section 2, we first detail Random Forest principles and review different methods proposed in the literature. We then explain our experimental protocol for using Forest-RI on the MNIST Database in section 3. Finally we present some results and discussion to conclude on the Random Forest global behavior according to the studied parameters.

## 2. Random Forests

Actually, Random Forest is a general term for classifier combination that uses $L$ tree-structured classifiers $\{h(x, \Theta_k),\ k = 1, ...L\}$ where $\Theta_k$ are independent identically distributed random vectors and $x$ is an input. With respect to this definition, one can say that Random Forest is a family of methods in which we can find several algorithms, such as the Forest-RI algorithm proposed by Breiman in [3], and cited as the reference method in all RF related papers.

In Forest-RI algorithm, Bagging is used in tandem with a random feature selection principle. The training stage of this method consists in building multiple trees, each one trained on a bootstrap sample of the original training set – i.e. the Bagging principle – and with a CART-like induction algorithm [4]. This tree induction method, sometimes called RamdomTree, is a CART-based algorithm that modifies the feature selection procedure at each node of the tree, by introducing a random pre-selection — i.e. the Random Subspace principle.

Each tree is grown as follows :

- For $N$ instances in the training set, sample $N$ cases at random with replacement. The resulting set will be the training set of the tree.

- For $M$ input features, a number $K << M$ is specified such that at each node, a subset of $K$ features is drawn at random, and among which the best split is selected.

- Each tree is grown to its maximum size and unpruned.

Consequently this algorithm works according to two main parameters : the number $L$ of trees in the forest, and the number $K$ of features pre-selected for the splitting process.

During the past few years, the RF family has been enlarged by several researchers, each of them proposing a variant of the Forest-RI algorithm.

Breiman has introduced in [3] another procedure for growing a RF, called Forest-RC, in which the split at each node is based on linear combinations of features instead of a single one. This allows to deal with cases with only few inputs supplied, that the original Forest-RI method can hardly handle.

Robnik in [15] tries to improve the combination procedure of the original Forest-RI, by introducing a weighted voting method. The goal is to take into account a restricted subset of the classifier outputs, based on individual accuracies on similar instances. According to Breiman procedure for similarity evaluation [3], classifier accuracies on similar instances are examined in order to remove from the vote those that show the lowest values.

Some works have also studied the random aspect of the RF, and have tried to go one step further in that way. Geurts et al. for example have proposed in [9], the Extremely Randomized Trees method. It consists in using randomization for the feature selection at each node, as in Forest-RI, but also for a cut-point selection procedure. In that way, each tree is totally randomly grown. Earlier, Cutler and Zhao introduced in [6] the PERT algorithm (for Perfect Random Tree Ensemble) for which the process is almost the same, except that the cut-point is computed from two randomly selected instances.

Another direction of RF investigations proposed in the literature is what Boinee et al. called Meta Random Forests [1]. It consists in using RF as base classifiers of combination techniques. They experiment Bagged Random Forests and AdaBoosted Random Forests to respectively study Bagging and Boosting using RF as base classifiers.

In spite of all those investigations on designing an accurate RF method, none of them have actually managed to prove the superiority of one method over the others. In addition, there does not exist any referenced work that present experimental results on using RF algorithms in a practical point of view. And some parameter values are commonly used without any theoretical nor empirical justifications. For example, the number $L$ of trees in forests is commonly arbitrarily fixed to 100. Breiman also chooses to test the Forest-RI algorithm with $K = log_2 M + 1$, where $M$ is the training set size, but without explaining the reason for this choice. We thus propose to study this practical approach by testing RF algorithms in a parameter tuning process. We detail our experimental protocol in the following section.

# 3. Experiments

The idea of our experiments is to tune the RF main parameters in order to analyse the "correlation" between the RF performances and the parameter values.

In this section, we first detail the parameters studied in our experiments and we explain the way they have been tuned. We then present our experiment protocol, by describing the MNIST database, the test procedure, the results recorded and the features extraction technique used.

## 3.1. Parameters

As mentioned above, we tuned the two parameters of the Forest-RI method in our experiments : the number $L$ of trees in the forest, and the number $K$ of random features pre-selected in the splitting process. In [3] Breiman states that $K$ has to be greater than 1, in which case the splitting variable would be totally randomly selected, but does not have to increase so much. Our experiments aim at progressively increasing this value to highlight whether or not this statement is true. Breiman also decides for his experiments to arbitrarily fix the number of trees to 100 for the Forest-RI algorithm. Thus, another goal of this work is to study the behavior of the method according to the number of trees, so that we would be able to distinguish a global tendency. As RF training process is quite fast, a wide range of trees can be grown inside the forest.

Consequently, we have drawn two ranges of values for $K$ and $L$. Concerning the number $L$ of trees, we have picked six increasing values, from 10 to 300 trees. They have been chosen according to the global tendency that appeared during the experiments. Using less than 10 trees has proven to be useless, as well as increasing the number of trees beyond 300 trees does not influence the convergence of the recognition rate.

Concerning the number of features we have tested 20 values following the same approach. This time small values have proven to be more interesting for seeing the global tendency of the recognition rate. Thus we have tested each value of $K$ from 1 to 16, and then five more greater values from 20 to 84.

## 3.2. Experimental protocol

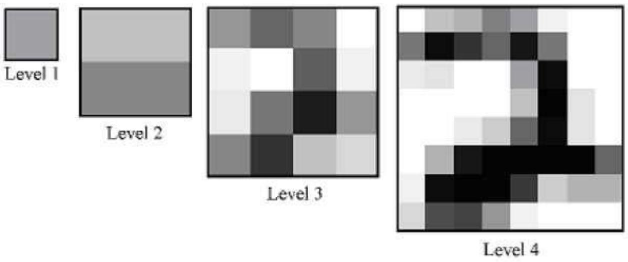
The handwritten digit MNIST database is made of 60,000 training samples and 10,000 test samples [12]. The digits have been size-normalized and centered in a fixed-size image. It is a good database for people who want to try learning techniques and pattern recognition methods on real-world data while spending minimal efforts on preprocessing and formatting.

In this experiment we would like to have an idea of the result variabilities. We have therefore divided the original training set into five training subsets of 10,000 samples. Let $L_s$ denote the original 60,000 samples training set and $T_s$ the 10,000 samples test set. We denote by $L_{s_i}$ each of the 5 learning subsets. In $L_s$ the classes are not equally represented, that is to say that some of them contain less than 6,000 samples. However we would like to use strictly balanced training sets, i.e. training sets with equally distributed classes. We have consequently decided to use only five subsets instead of six. Moreover it has allowed us to reduce the tree-structure complexities.

The Forest-RI algorithm has been run with each couple of parameters on the five $L_{s_i}$ training sets, so that a RF was grown for one couple of parameters associated to one $L_{s_i}$. Results on each run have been obtained by testing on the $T_s$ set. Consequently we have obtained five recognition rates for each couple of parameters, for which we have computed the mean value. By recognition rate we mean the percentage of correctly classified instances among all the test set samples, obtained with the forest built in the training stage.

With this work, our aim was not to discuss the influence of the feature quality on the performance of the classifier nor searching for best intrinsic performance. Our aim is rather to understand the role of the parameter values on the behavior of the RF. That is why we have decided to arbitrarily choose a commonly used feature extraction technique based on a greyscale multi-resolution pyramid [14]. We have extracted for each image of our set, 84 greyscale mean values based on four resolution levels of the image, as illustrated in figure 1.

**Figure 1. Example of multiresolution pyramid of greyscale values of an image**



The results and tendencies are discussed in the following section.

## 4. Results and Discussion

Table 1 presents a synthesis of our results, according to the two studied parameters. For each element, we have noticed the mean recognition rate on the left and the maximum and minimum value among the five run, respectively on the top-right corner and the bottom-right corner. Highest mean recognition rates, i.e. greater than 93%, are highlighted.

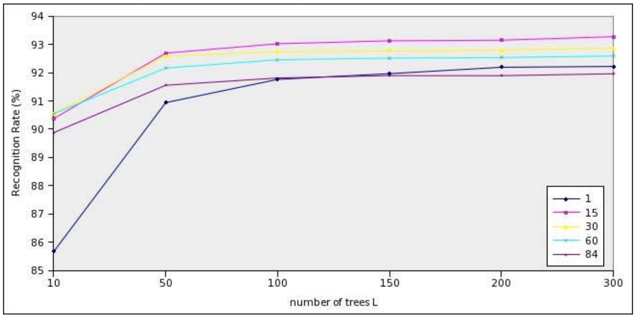
By examining the differences between maximum and minimum values of the recognition rates, we can have an idea of the performance variabilities. One can note that those values are almost constant, and does not show any global evolution tendency with respect to the two studied parameters.

**Table 1. Variation in recognition rate mean values according to $K$ and $L$**

| Nb Trees<br>Nb Feat | 10 | | 50 | | 100 | | 150 | | 200 | | 300 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | **85,68** | 86,32<br>84,52 | **90,94** | 91,37<br>90,55 | **91,76** | 91,96<br>91,58 | **91,96** | 92,22<br>91,78 | **92,19** | 92,34<br>92,01 | **92,22** | 92,48<br>92,04 |
| 2 | **86,70** | 87,26<br>86,42 | **91,61** | 91,76<br>91,27 | **92,10** | 92,29<br>91,92 | **92,25** | 92,33<br>92,18 | **92,37** | 92,58<br>92,16 | **92,57** | 92,77<br>92,46 |
| 3 | **87,96** | 88,28<br>87,53 | **91,93** | 92,12<br>91,56 | **92,46** | 92,19<br>92,19 | **92,64** | 92,83<br>92,35 | **92,67** | 92,98<br>92,33 | **92,77** | 93,03<br>92,49 |
| 4 | **88,18** | 88,48<br>87,90 | **92,13** | 92,25<br>92,02 | **92,65** | 92,85<br>92,47 | **92,78** | 93,00<br>92,40 | **92,90** | 93,10<br>92,63 | **92,93** | 93,15<br>92,42 |
| 5 | **88,82** | 88,90<br>88,71 | **92,15** | 92,38<br>91,98 | **92,70** | 92,87<br>92,42 | **92,83** | 93,05<br>92,50 | **92,94** | 93,19<br>92,61 | **93,04** | 93,20<br>92,79 |
| 6 | **89,33** | 89,54<br>89,15 | **92,46** | 92,63<br>92,09 | **92,90** | 93,10<br>92,61 | **92,98** | 93,16<br>92,55 | **93,06** | 93,24<br>92,72 | **93,14** | 93,32<br>92,82 |
| 7 | **89,70** | 89,70<br>88,88 | **92,47** | 92,80<br>92,09 | **92,88** | 92,59<br>92,59 | **93,04** | 93,22<br>92,72 | **93,05** | 93,21<br>92,80 | **93,13** | 93,36<br>92,90 |
| 8 | **89,60** | 89,86<br>88,99 | **92,71** | 92,87<br>92,31 | **93,00** | 93,22<br>92,67 | **93,08** | 93,30<br>92,74 | **93,11** | 93,27<br>92,81 | **93,21** | 93,49<br>92,96 |
| 9 | **89,80** | 90,28<br>89,38 | **92,63** | 92,87<br>92,35 | **92,98** | 92,98<br>92,66 | **93,06** | 93,30<br>92,81 | **93,16** | 93,43<br>93,00 | **93,24** | 93,51<br>92,96 |
| 10 | **89,92** | 90,26<br>89,69 | **92,69** | 92,91<br>92,50 | **92,97** | 93,23<br>92,61 | **93,13** | 93,47<br>92,66 | **93,20** | 93,54<br>92,82 | **93,20** | 93,56<br>92,84 |
| 12 | **90,18** | 90,39<br>89,75 | **92,67** | 92,82<br>92,33 | **93,05** | 93,34<br>92,54 | **93,20** | 93,48<br>92,84 | **93,24** | 93,47<br>92,90 | **93,27** | 93,45<br>93,09 |
| 13 | **90,28** | 90,82<br>90,01 | **92,72** | 92,95<br>92,54 | **92,98** | 93,25<br>92,69 | **93,11** | 93,45<br>92,68 | **93,19** | 93,51<br>92,90 | **93,24** | 93,49<br>92,97 |
| 14 | **90,36** | 90,69<br>89,98 | **92,72** | 92,98<br>92,47 | **93,02** | 93,27<br>92,68 | **93,08** | 93,35<br>92,68 | **93,13** | 93,38<br>92,75 | **93,20** | 93,47<br>92,90 |
| 15 | **90,37** | 90,77<br>89,88 | **92,69** | 92,91<br>92,45 | **93,02** | 93,37<br>92,60 | **93,12** | 93,40<br>92,74 | **93,15** | 93,46<br>92,85 | **93,27** | 93,59<br>92,96 |
| 16 | **90,50** | 90,86<br>90,25 | **92,63** | 93,01<br>92,43 | **92,91** | 93,40<br>92,68 | **92,98** | 93,32<br>92,72 | **93,11** | 93,52<br>92,83 | **93,13** | 93,39<br>92,97 |
| 20 | **90,44** | 90,75<br>89,75 | **92,71** | 93,11<br>92,36 | **92,95** | 93,22<br>92,47 | **93,10** | 93,28<br>92,78 | **93,06** | 93,31<br>92,76 | **93,19** | 93,47<br>92,88 |
| 30 | **90,56** | 90,83<br>90,10 | **92,58** | 92,89<br>92,16 | **92,74** | 92,91<br>92,63 | **92,77** | 92,90<br>92,58 | **92,80** | 92,96<br>92,51 | **92,86** | 93,11<br>92,51 |
| 42 | **90,55** | 91,23<br>90,02 | **92,16** | 92,38<br>91,64 | **92,45** | 92,06<br>92,06 | **92,51** | 92,84<br>92,09 | **92,53** | 92,85<br>92,05 | **92,59** | 92,85<br>92,14 |
| 60 | **89,88** | 90,26<br>89,39 | **91,55** | 91,82<br>91,23 | **91,81** | 92,10<br>91,51 | **91,89** | 92,15<br>91,62 | **91,89** | 92,18<br>91,54 | **91,96** | 92,25<br>91,48 |
| 84 | **88,61** | 89,05<br>88,20 | **89,99** | 90,35<br>89,49 | **90,24** | 89,73<br>89,73 | **90,28** | 90,71<br>89,70 | **90,30** | 90,62<br>89,72 | **90,51** | 90,83<br>89,92 |

We have first studied the influence of the parameter $L$, i.e. the number of trees in the forest. Figure 2 presents recognition rates with respect to the number of trees for fixed values of $K$. We can see a global tendency of the recognition rate to raise for an increasing number of trees. It appears that this increasement is not linear but logarithmic. One can conclude from this, that with respect to an increasing number of trees, the Random Forests accuracy converges. It seems on this figure that the rise of the recognition rate begins to considerably slow down from 100 trees in the forest. However, we think that more investigations on this direction are needed to confirm this value. We also have noticed that the behavior of the random forests is different with $K = 1$, in which case the splitting feature is totally randomly selected. This confirms Buntine et al. and Liu et al. results given in [5, 13], that conclude that the tree inducing process should implement a splitting selection measure rather than randomly choose splitting tests.

**Figure 2. Recognition rates wrt $L$**



Then we have focused on the second parameter $K$, i.e. the number of features randomly pre-selected for the splitting procedure. Figure 3 presents some curves of the recognition rate with respect to the number of features, and each one for a fixed number of trees. This time, there is not a global increase. All the curves begin to raise for an increasing number of features from $K = 1$ up to $K = 6$, then are almost constant till $K = 20$, and finally begin to decrease to reach the minimum in $K = 84$ – except for $L = 10$ for which the minimum is reached for $K = 1$. According to Breiman's work the reason of this decrease can be that a too much important portion of features pre-selected, makes the diversity decrease between trees in the forests. Indeed the more features are randomly pre-selected, the more the splits will be identical from one tree to another – since they are then selected according to the splitting criterion. As for the previous comment, we think here that those values need to be studied one step further to be confirmed.

Figure 4 proposes another synthetic 3-D representation of our results. In this diagram, tendencies with respect to the two parameters simultaneously, clearly appear and a maxima surface can be identified with dark shaded greyscales. The recognition rate maxima are reached in an area defined by the intervals $[100, 300]$ for $L$ and $[5, 20]$ for $K$.

## 5. Conclusion and Future Works

With this work we have experimented the Forest-RI algorithm with different parametrization values in order to present a study on Random Forest in a strictly pragmatic approach. For that purpose we have chosen to test the method

**Figure 3. Recognition rates wrt $K$**

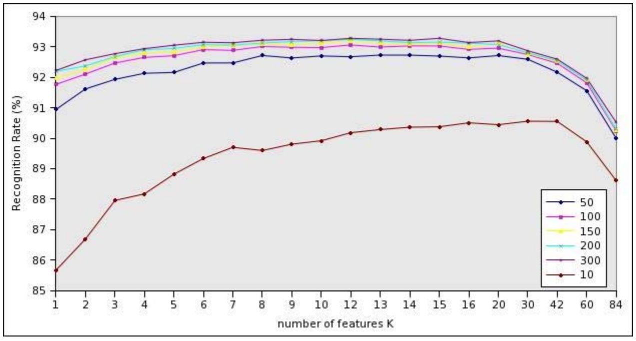
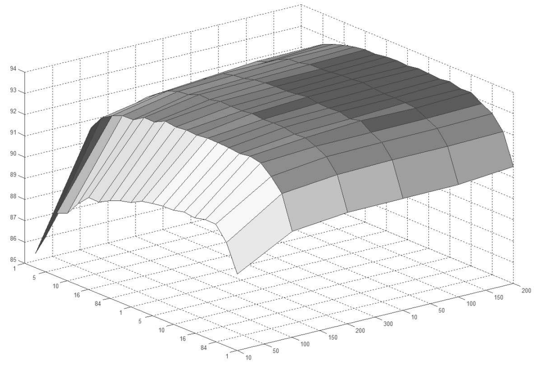


**Figure 4. Recognition rates wrt $K$ and $L$**



on the MNIST handwritten digit database, that provides a large number of samples of real-world context data. We have managed to highlight global tendencies of the Random Forest behavior according to some classical parameters. This has allowed us to present empirical results that draw primary conclusions on parametrization influence for using Random Forests. We have shown that the number $K$ of features randomly selected in the Forest-RI process has to be greater than 1, but should not be actually so high (i.e. greater than 20). We have also highlighted that the recognition rate tends to converge for an increasing number of trees. We can see that for a well defined value of $K$ (i.e. $K \simeq 12$ in this case) the recognition rate does not rise a lot beyond 100 trees.

Obviously we think that further investigations in that way are needed to confirm those conclusions. For example it would be necessary to make similar experiments with different databases, in terms of features space dimension and training set size, to generalize our results or even to determine whether or not the interesting values found in those experiments depend on those characteristics. It would also be interesting to focus on other characteristics of the Random Forest process such as diversity or individual tree strength, to empirically confirm Breiman theorical statements about parameter influence on correlation and individual strength of Random Forests [3] [11].

## References

[1] P. Boinee, A. D. Angelis, and G. Foresti. Meta random forests. *Internationnal Journal of Computationnal Intelligence*, 2(3):138–147, 2005.

[2] L. Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

[3] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[4] L. Breiman, J. Friedman, R. Olshen, and C. Stone. *Classification and Regression Trees*. Chapman and Hall (Wadsworth, Inc.): New York, 1984.

[5] W. Buntine and T. Niblett. A further comparison of splitting rules for decisin-tree induction. *Machine Learning*, 8:75–85, 1992.

[6] A. Cutler and G. Zhao. Pert - perfect random tree ensembles. *Computing Science and Statistics*, 33, 2001.

[7] T. Dietterich. An experimental comparison of three methods for constructing ensembles of decision trees : Bagging, boosting, and randomization. *Machine Learning*, 40:139–157, 1999.

[8] Y. Freund and R. Schapire. Experiments with a new boosting algorithm. *ICML*, 1996.

[9] P. Geurts, D. Ernst, and L. Wehenkel. Extremely randomized trees. *Machine Learning*, 36(1):3–42, 2006.

[10] T. Ho. The random subspace method for constructing decision forests. *IEEE Trans. on PAMI*, 20(8):832–844, 1998.

[11] L. Kuncheva. *Combining Pattern Recognition. Methods and Algorithms*. John Wiley and Sons, 2004.

[12] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[13] W. Liu and A. White. The importance of attribute selection measures in decision tree induction. *Machine Learning*, 15(1):25–41, 1994.

[14] Y. Prudent and A. Ennaji. A k nearest classifier design. *Electronic Letters on Computer Visions and Image Analysis*, 5(2):58–71, 2005.

[15] M. Robnik-Sikonja. Improving random forests. *ECML, LNAI 3210, Springer, Berlin*, pages 359–370, 2004.

[16] J. Rodriguez, L. Kuncheva, and C. Alonso. Rotation forest : A new classifier ensemble method. *IEEE Trans. on PAMI*, 28(10), 2006.

[17] I. Witten and E. Frank. *Data Mining: Practical machine learning tools and techniques, 2nd Edition*. Morgan Kaufmann, San Francisco, 2005.