International Journal of Innovative Studies in Sciences and Engineering Technology (IJISSET)

ISSN 2455-4863 (Online) www.ijisset.org *Volume: 3 Issue: 10 | October 2017*

The CART Stability Analysis by the Diffeomorphism-Kernel Plug-in

Ibtissem Ben Othman [1], Molka Troudi[2], Faouzi Ghorbel[1]

[1]GRIFT Research Group,CRISTAL Laboratory, School of Computer Sciences, Manouba 2010, Tunisia

[2] Department of Quantitative Methods, IHEC Carthage, Carthage 2016, Tunisia

Abstract: *The traditional statistical classifiers have been later used in many applications across different disciplines. However, they can attend problems in high dimension space. The Classification and Regression Trees (CART) presented an alternative solution which is not based on normality assumption like some classical statistical classifiers such as the Bayesian decision rule. We perform, in the present research, a new insight to compare different models of CART decision trees by an adjusted non-parametric probability density function estimate of misclassification error rates. Such estimator is computed by the diffeomorphism-kernel Plug-in algorithm which considers the error rates positivity support. The bagging and Adaboost training algorithms may improve the classification efficiency for CART decision trees. Some stochastic simulations try to prove statistical stability criterion. After that, the experimental results will be presented in handwritten digits recognition problem in the features space. Different invariant descriptors will be compared.*

Keywords: *Adaboost, Bagging, Classification And Regression Tree, Probability density estimate, diffeomorphism-kernel Plug-in algorithm, Stability.*

1. INTRODUCTION

In practice, it is well known that the precision of the estimation, in high dimension spaces, requires non-realistic training samples size. Thus, the sample data size required to obtain satisfying classification accuracy, increases exponentially with data space dimension. Hence, the dimension reduction step is needed to overcome this problem in pattern recognition. The traditional statistical methods such as the Bayesian decision theory and the Linear Discriminant Analysis have been successfully used in many application areas, as the linear dimension reduction and classification purposes. However, the Bayesian classifier [7] is based on the normality assumption. The Classification and Regression Trees (CART) [4] present an alternative solution to overcome the normality hypothesis. In the same direction, the Artificial Neural Networks (ANNs or NNs) have proved quite successful to solve complex problems in many applications [18]. Thus, we can find them working in data classification and non-linear dimension reduction. However, the lack of control over its mathematical formulation explains the instability of its classification results in some situations, compared to the statistical approach.

In [5, 6, 14, 32], the decision trees have been compared to the ANNs. Some articles involved CART decision trees in their comparative study [1, 28]. Brown and al. have proved in [5] that NNs prediction accuracy was better than that of CART models on multimodal classification problems where data sets are large with few attributes. The authors have also concluded that the CART model did better than the NNs one with smaller data sets and large numbers of irrelevant attributes. For non-linear data sets, NNs and CART models outperform Linear Discriminant Analysis [6].

In order to compare the different approaches, most of the researchers have tried to compare their accuracy prediction while forgetting the instability criterion of some classifiers. Thus, the objective comparison studies stayed marginal. In the present study, we are performing a new criterion to conduct a comparative study between several classifiers by estimating the probability density function (pdf) of their error rates. Since the support of an error rate probability density is bounded, we used the diffeomorphism-kernel-pdf-estimate bandwidth optimization by the plug-in algorithm.

For this comparative study, we have focused on the numerous classification procedures with proven effectiveness to improve the classifiers efficiency like the bootstrap aggregation and Adaboost algorithm for the CART decision trees.

The present manuscript is structured as follows: We begin in section 2 by recalling the classification and regression trees and the techniques which may improve their stability degree. The main idea of the classifiers stability comparison is to estimate their error rates density. Thus, the non-parametric kernel method is described in the subsection 2.2. While the classifiers misclassification rates (MCR) density

functions are known by their bounded support, we refer to the diffeomorphism-kernel plug-in algorithm. In section 3, we introduce the new criterion based on the diffeomorphism-kernel plug-in algorithm. The comparison studies are performed by visualizing the results through stochastic simulations of multivariate Gaussian distributions. In section 4, we will apply these comparative analyses to the evaluation of real pattern recognition problem. So, we intend to test the different classifiers efficiency for the handwritten digits recognition problem by classifying their corresponding descriptors. Such features form a set of invariant parameters under similarity transformations and closed curve parameterizations. This set has good properties as completeness and stability.

2. STATE OF THE ART

In the present section, we are going to review the most important topics for CART decision trees. The estimation of the classifiers error rates density function presents the basic idea for our comparative analysis. There are two types of methods for probability density estimating; parametric and non-parametric. The parametric techniques (such as the maximum-likelihood estimation and the Pearson system) are based on some assumptions and specified statements. Since the misclassification rates densities are a priori unknown, the non-parametric methods are the most accurate. The histogram estimator, the orthogonal functions and the kernel method present the most frequently nonparametric techniques used in the literature. The histogram method has the disadvantage of discontinuity. Although the orthogonal function technique is suitable for any type of support, it may encounter the Gibbs effect. We suggest using the kernel method in our research.

2.1 Classification and regression tree

The Classification and Regression Tree (CART) is a non-traditional statistical non-parametric model developed by Breiman et al. in [4]. The CART is obtained by a binary recursive partitioning procedure which can be graphically illustrated as a decision tree. The CART models have been used in the areas of prediction and classification. In our research, we have used the classification-type CART model which is applied for classifying discrete dependent variables.

First, the CART uses the Gini index for its impurity function to construct a large tree and then prune it to a smaller size to minimize an estimate of the misclassification error. It employs the 10-folds (default) cross validation for this purpose. Thus, pruning CART decision trees assists in its architecture stability. Furthermore, the Bagging and Boosting algorithms present general combining methods for stabilizing the CART decision trees.

Unlike bagging, which is based on a simple averaging of predictions, boosting presents an iterative procedure which uses a weighted average of results obtained from applying a prediction method to various samples. Boosting is a method of combining classifiers, which are iteratively created from weighted versions of the learning sample, with the weights adaptively adjusted at each step to give increased weight to the cases which were misclassified on the previous step. The final predictions are obtained by weighting the results of the iteratively produced predictors. In addition, boosting is originally applied to weak learners (having an error rate of 50%) whereas this is not the case with bagging.

Adaboost is a boosting algorithm developed by Freund and Schpire [8] to be used with classifiers. There are two versions of Adaboost: Adaboost.M1 and Adaboost.M2. When only two classes are involved, there is no difference between the two versions. However, when the number of classes increases, Adaboost.M2 gives better results than Adaboost.M1. Indeed, for the method to be effective, the weighted re-substitution error rates must be less than 0.5 for the weak learners, which can be difficult to achieve when several classes are involved.

2.2 The kernel estimate method

The classifiers stability evaluation criterion is based on their error rates density estimation. For this purpose, we refer to the non-parametric kernel method.

The KDE (Kernel Density Estimate) is a non-parametric estimate which have been introduced by Rosenblatt in 1956 [29] and developed by Parzen in 1962 [25]. Considering a sample of size N noted by (X_1, ..., X_N), the density probability function is expressed by:

$$\hat{f}_N(x) = \frac{1}{Nh_N}\sum_{i=1}^{N} K\left(\frac{x - X_i}{h_N}\right)$$

where h_N is called bandwidth or smoothing parameter and K is a kernel function having the following properties:

$$\int_{-\infty}^{+\infty} K(u)du = 1$$

$$K(-u) = K(u)$$

In this study, *K(.)* is chosen as the Gaussian kernel which is expressed by:

$$K(x) = \frac{1}{\sqrt{2\pi}} \exp(-\frac{1}{2} x^2)$$

Following an asymptotic study, the MISE (Mean Integrated Square Error) is approximated by (ref):

$$MISE \approx \frac{M(K)}{Nh_N} + \frac{J(f(X))h_N^4}{4}$$

where;

$$M(K) = \int_{-\infty}^{+\infty} K^2(x)dx$$

$$J(f_X) = \int_{-\infty}^{+\infty} (f_X''(x))^2 dx$$

Minimizing the MISE, the optimal value of the smoothing parameter, noted by h_N^*, has the following expression:

$$h_N^* = N^{-1/5}[J(f_X)]^{-1/5}[M(K)]^{+1/5}$$

M (K) can easily be determined analytically or numerically. However, we note that *J(f)* is a function of the unknown density. Several methods have been developed in the literature in order to approach the optimum value of the smoothing parameter [33].

In this paper, we deal with the semi-bounded densities because the error rate is strictly positive. Therefore, a more accurate estimate will be obtained using the diffeomorphism-kernel method [30, 31] which significantly reduces the Gibbs' effect. This estimator is a generalization of the KDE. It is suitable for the functions defined on the interval [a, b]. The density function is expressed by:

$$\hat{f}_N(x) = \frac{|\phi'(x)|}{Nh_N} \sum_{i=1}^{N} K\left(\frac{\phi(x) - \phi(X_i)}{h_N}\right)$$

where φ is a C1-diffeomorphism which have the infinity for limit as *'x'* approaches *'a'* or *'b'*.

The problematic of optimizing the smoothing parameter can be resolved by using the same methods as those used for conventional kernel analysis.

However, as shown in [34], an asymptotic study of the diffeomorphism-kernel estimate, allows better approach to optimal smoothing parameter in the MISE sense. Then, its expression becomes the following:

$$h_N^* = \left[M_\phi(K)\right]^{\frac{1}{5}} \left[J_\phi(f)\right]^{-\frac{1}{5}} N^{-\frac{1}{5}}$$

where

$$M_\phi(K) = M(K) \int_R |\phi'(x)| f(x)\, dx$$

$$J_\phi(f) = \int_R \frac{F^2(x)}{\left[\phi'(x)\right]^8} dx$$

and

$$F(x) = \left[f(x)\left[3\phi''(x)^2 - \phi'(x)\phi'''(x)\right] \right] - 3f'(x)\phi'(x)\phi''(x) + f''(x)\left[\phi'(x)\right]^2$$

In this section, we will recall, in a first time, the conventional plug-in algorithm and the diffeomorphism-kernel plug-in algorithm. Then, we are going to show, through a comparative study, the benefit of using the diffeomorphism-kernel plug-in algorithm for densities having a semi bounded support.

2.2.1. Conventional Plug-in algorithm

The plug-in algorithm allows a close estimation of the optimal value of the smoothing parameter for kernel estimator using a recursive resolution. In a first time, the unknown density *f* is estimated using a random value of h_N noted h_N^0. This estimate is, of course, a bad estimate. However, this first approach to the unknown density will give new value of h_N^1 which is closer to the optimal value. So, h_N^1 is used to re-estimate the unknown density and so on.

The different steps of conventional Plug-in algorithm are the following:

Step 1: initialisation of *M(K)* and *J(f)*.

Step 2: computing $h_N^{(0)}$.

Step 3: estimation of the pdf $f^{(0)}$.

Step 4: re-estimation of $J^{(k)}(f)$.

Step 5: return to the second step.

Step 6: stopping the algorithm when the difference between $h_N^{(k)}$ and $h_N^{(k-1)}$ is very low (less than 1%).

2.2.2. Diffeomorphism-Kernel plug-in algorithm

The implementation of this extended version presents further difficulties compared to classical plug-in algorithm. Indeed, for the plug-in algorithm for KDE,

M(K) is a constant which can be determined analytically or numerically. With regard to the plug-in algorithm adapted to DKDE, *M(K)* depends on unknown pdf. Similarly, *J(f)* depends not only on *f''*, but also on *f* and *f'*. Therefore, the complexity of the plug-in algorithm adapted to DKDE is increasing.

We describe below the algorithm and the computation complexity, as follows:

Step 1: Initialize arbitrary $M_\varphi(K)$. In practice $M_\varphi^0(K)$ can be equal to *M(K).*

Step 2: Fix arbitrary $J_\varphi^0(f)$, then deduce the first value of the optimal bandwidth; h_N^0.

Step 3: Estimate $f^{(0)}$.

Step 4: Approximate the different quantities: $M_\varphi^{(k)}(K)$, $f^{(k)'}$ et $f^{(k)''}$ for each iteration *k.*

Step 5: Estimate $J_\varphi(f^{(k)})$. The value of $h_N^{(k)}$ is so deducted from the k^{th} iteration.

Step 6: Approximate $f^{(k)}$. Stop the algorithm when the difference between $h_N^{(k)}$ and $h_N^{(k-1)}$ is relatively low (below 1%).

3. Comparison Study between CART Models

In the literature, numerous authors have compared the classifiers performance while they have ignored the stability criterion. Some classifiers are instable, small changes in their training sets or in constructions may cause large changes in their classification results. Therefore, an instable classifier may be too dependent on the specific data and has a large variance. Thereby, a good model should find a balanced equilibrium between the error rate bias and variance. These two later terms present the first and second order statistical moments of the classifiers error rates values. Therefore, these two low order moments do not enable to describe completely the statistical dispersions, especially for complex situations as the multimodal conditional error rates distributions. In order to analyze and compare the stability and performance of each classifier, we have to illustrate their error rate probability densities in the same figure. While the probability density curve on the left has the small mean, the one on the right has the high mean. Clearly, the classifier, whose curve is on the left, is the most efficient one. An instable classifier is characterized by a high variance. When the variance is large, the curve is short and wide, and when the variance is small, the curve is tall and narrow. As a result, the classifier with the largest density curve is the least stable one. This criterion is basic for any stability and performance analysis of each classifier.

3.1 *The stability criterion*

The first step before comparing is to train the different classifiers, and then we proceed by measuring their prediction results for *N* independent test sets. Suppose that $(X_i)_{1\le i\le N}$ are the *N* generated error rates resulting from testing a particular classifier (such as Bayes, CART or ANN). These misclassification rates (MCR) are viewed as random variables and are supposed to be independent and identically distributed. They are also considered to have the same probability density function (pdf). Letting f_X denotes this pdf. Such pdf is estimated by applying the Plug-in kernel algorithm, which optimizes the mean integrated square error criterion to search the best estimator smoothing parameter.

In practice, the observed misclassification rates are real positive values. Thus, using the classical kernel density estimator may cause some convergence problems at the edges: the Gibbs phenomenon. The use of the kernel diffeomrphism plug-in algorithm presents the alternative solution.

3.2 *Stability evaluation by simulations*

The stability comparison between the CART models is performed by stochastic simulations. For this purpose, we have considered a binary classification problem adapted to a mixture of two different Gaussian distributions.

For the training phase, we have generated one set including 1000 samples for each class. By using this training set, we look for fixing the optimal CART models parameters.

For the test phase and in order to analyse the classifiers stability, we have generated 100 supervised and independent test sets having the same size of the training one (including 1000 samples for each class). Then a set of 100 error rates are retained for each classifier. Their probability densities are estimated using the diffeomorphism-kernel Plug-in algorithm reviewed in the previous section.

Table 1 enumerates the distributions parameters corresponding to the different illustrations of figure 1. These simulations present six mixtures of two Gaussians in spaces of dimensions 3 and 10, respectively. The two cases '*a*' and '*d*' illustrate Homoscedastic Gaussians. The other cases present heteroscedastic ones; cases '*b*' and '*f*' treat the problem of the two superposed Gaussians having the same

means, vector and different covariance matrices. Case '*c*' shows the problem of the two truncated Gaussians. In this situation, the second samples cloud surrounds the first one in a ball centred at the origin.

In figure 1, we try to compare the different models of the CART decision trees; conventional CART, CART-Bagging and CART-Adaboost.

The density of the CART error rates is illustrated by the curve furthest to the right (having the largest error rates mean) and the wider (with the largest error rates variance). Thus, CART is the least efficient and least stable decision tree for the six simulations.

In table 2, the three CART decision tree models, are moreover analysed by presenting their error rates biases and variances.

The results show that the optimization procedures for the CART decision trees (Bagging and Adaboost algorithms) improve their performance and stability. Although, the Adaboost training algorithm gives better results than the Bagging technique and conventional CART.

Table 1: *Distributions Parameters*

		Gaussian 1		*Gaussian 2*	
		μ_1	$\sum_1$	μ_2	$\sum_2$
3 di m	*a*	(1,1,1)	$\mathbf{I_3}$	(1.5,1.5,1.5)	$\mathbf{I_3}$
	b	(1.5,2,-3)	$0.5x\mathbf{I_3}$	(1.5,2,-3)	$2x\mathbf{I_3}$
	c	(0,0,0)	[0.06 0.01 0.01] x $\mathbf{I_3}$	(0.1,0.1,0.1)	[0.01 0.06 0.05] x $\mathbf{I_3}$
10 di m	*d*	$(1.5,..,1.5)_{10}$	$\mathbf{I_{10}}$	$(2,..,2)_{10}$	$\mathbf{I_{10}}$
	e	$(0,..,0)_{10}$	$\mathbf{2xI_{10}}$	$(2,..,2)_{10}$	$3x\ \mathbf{I_{10}}$
	f	$(0,..,0)_{10}$	$\mathbf{I_{10}}$	$(0,..,0)_{10}$	$2x\ \mathbf{I_{10}}$

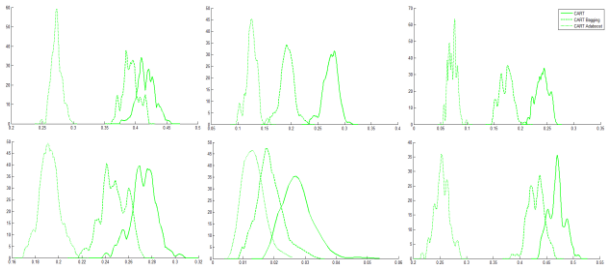

Figure 1: *Error rates densities of CART (-), CART-Bagging (--) and CART-Adaboost (..), of the various simulations of table 1.*

Table 2: *Comparison results of CART models.*

Cases	*CART*		*CART-Bagging*		*CART-Adaboost*	
	Mean	*Variance x10^{-4}*	*Mean*	*Variance x10^{-4}*	*Mean*	*Variance x10^{-4}*
a	0.4166	1.9931	0.3921	1.6294	0.2727	0.6715
b	0.2744	1.7386	0.1927	1.5289	0.1260	1.0595
c	0.2410	1.4345	0.1730	1.4711	0.0729	0.6412
d	0.2727	1.2285	0.2478	1.2278	0.1922	0.7202
e	0.5406	1.3074	0.5373	1.2651	0.5396	0.8698
f	0.4688	2.2906	0.4303	2.2108	0.2541	1.8215

4. APPLICATION TO HANDWRITTEN DIGIT RECOGNITION PROBLEM

In order to compare the classifiers stability and performance, we refer in the present section to the handwritten digit recognition problem. This task is still one of the most important topics in the automatic sorting of postal mails and checks registration. The database used to train and test the different classifiers described in this paper was selected from the publicly available MNIST database of handwritten digits [17]. This database contains 60,000 training images and 10,000 test ones. For the training and test sets, we randomly select, from the MNIST training and test sets respectively, single digit images. Both sets contain 1000 images for the 10 digit classes (10,000 for both sets).

4.1 *Features extraction*

The most delicate step in handwritten digits recognition problem is the invariant feature extraction. The selection of the appropriate primitives must be based on a set of non-exhaustive criteria. This choice will affect the calculation speed, the discrimination efficiency, the invariability to the geometric transformations, the completeness and the stability. In the literature, researchers often try to fulfil only the first two criteria which are not really sufficient for handwritten digits recognition which have planar shapes summarizing their contours.

The Fourier descriptors are invariant description forms and were presented in many analytical works on pattern recognition, such as handwriting recognition [27, 37]. However, researchers have shown that the Fourier descriptors of the parameterized function do not contain sufficient information to characterize the shape of an object. Therefore, the completeness criterion was introduced by Crimmins, ensuring the reconstruction of the image from its invariant features. All Crimmins descriptors have a complete set of Fourier descriptors for planar shapes, in the sense that two objects having the same shape, if and only if they have the same set of Fourier descriptors. However, the completeness property is purely algebric and does not take into account the similarity between the objects observed in their natural scene.

Based on the concept of the Fourier transform in a locally compact separable group Ghorbel introduced [11], a family of complete and stable invariant features with respect to the starting point on the curve and compared with direct similarities groups within the

forms in summarizing their contours. This stability property provides some robustness to numerical calculation errors and distortion introduced by certain measures (bad acquisition, scanning, quantification ...).

In this manuscript, we will compare the Fourier and Ghorbel descriptors for handwritten digits recognition. Thus, we calculate the Fourier coefficients from the handwritten digits contours. The experiments show that the first 14 Fourier coefficients are sufficient.

4.2 *Stability evaluation*

The experiments of the previous section have already showed that the Adaboost training algorithm gives better results than the Bagging technique and conventional CART. For these reasons, we apt for the use of CART-Adaboost for the next studies in our paper. Thus, we try to compare CART-Adaboost with the Bayesian classifier and ANN of type Bayesian neural network.

The selected descriptors size is high ($D = 14$). In order to apply Bayesian rule, dimension reduction becomes necessary. The transformation matrix is estimated for the Fisher LDA from the training set, which projects the data on the appropriate dimensions subspace (two dimensions in our study). In order to compare the classifiers stability, we evaluate their performance for 100 times using the 10-folds cross validation (CV) algorithm. Their misclassification rates (MCR) are calculated on the test sets selected by the CV algorithm from the MNIST test set (N=1000 images for each class).

In order to obtain meaningful comparison between the different types of classifiers, we evaluate their performance and stability degrees. Figure 2, 3 and 4 show the error rate probability densities estimated using the diffeomorphism-kernel semi-bounded Plug-in algorithm. This procedure is qualified by its sufficient precision on the stability aspects.

In figure 2, two digits yield the binary classification problem of the digits 2-5, 4-7 and 6-9. Figure 3 results the problem of classifying 4 digits; 1-2-3-4 and 5-6-7-8. In figure 4, the 10 digits are classified by both Fourier descriptors and Ghorbel descriptors. In table 3 and 4, we summarize the MCR means and variances obtained for the two types of descriptors using the three classifiers for Fourier descriptors and Ghorbel descriptors, respectively.

The experimental results of classifying neural and statistical classifiers on the MNIST database, can concretely resume for two points. First, CART is the best classifier for binary cases; however ANN is the best one in multiclass cases. Second, the Ghorbel descriptors are significantly more stable than the Fourier descriptors.

5. CONCLUSIONS

In the current research, a stability comparison analysis involving artificial neural networks, Bayesian decision theory and CART decision trees are performed. In addition to the prediction accuracy, a new stability criterion is applied based on the classifiers error rates probability based on densities estimation. The proposed variant of the semi-bounded diffeomorphism-kernel plug-in algorithm provides a significant precision for the densities estimations which are characterized by their semi-infinity support.

We have compared the stability and the performance of the statistical and the neural approaches using simulated data from Gaussian distributions and real-world data (handwritten digits images). The stochastic simulations demonstrated the superiority of the statistical classifiers in their performance and stability. By applying the handwritten digits recognition problem, we have proven the performance of ANNs. In addition, the classifiers combination and the Bayesian approach for modeling NNs enhance their performance and stability. Similarly, the Bagging and Adaboost procedures improve the CART model efficiency.

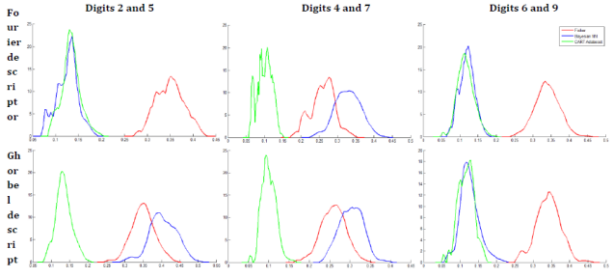

Figure 2: *Error rate densities for Fourier descriptors and Ghorbel descriptors for the two digits classes: 2 & 5, 4 & 7, and 6 & 9.*

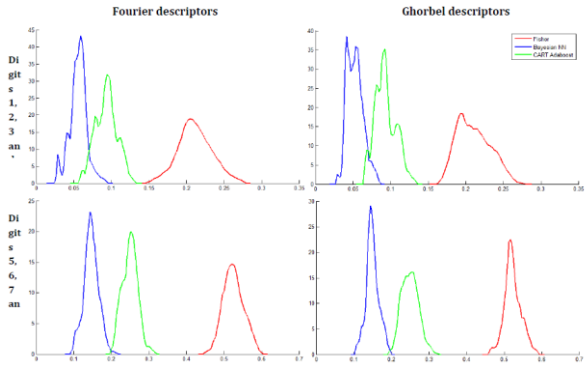

Figure 3: *Error rate densities for Fourier descriptors (in the left) and Ghorbel descriptors (in the right) for the four digits classes: 1, 2, 3 & 4, and 5, 6, 7 & 8.*

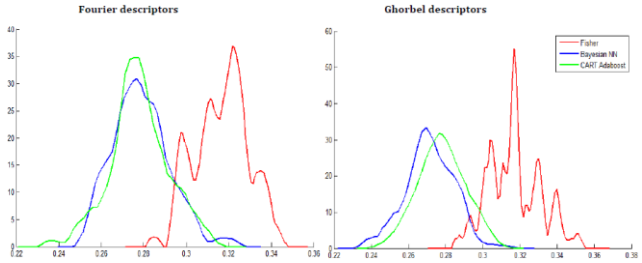

Figure 4: *Error rate densities for Fourier descriptors (in the left) and Ghorbel descriptors (in the right) for the ten digits classes.*

Table 3: *Comparison results of neural and statistical classifiers on the MNIST database for Fourier descriptors.*

Digit classes	Bayes		CART		ANN	
	Mean	Var x10^{-4}	Mean	Varx10^{-4}	Mean	Var x10^{-4}
2-5	0.3498	9.0777	0.1340	4.0354	0.1260	5.3232
4-7	0.2596	10	0.1000	4.1111	0.3285	11
6-9	0.3393	10	0.1165	4.8056	0.1195	4.1641
0-1-2-3	0.2116	4.6760	0.0932	2.0082	0.0560	1.2184
4-5-6-7	0.5252	6.9183	0.2497	3.9994	0.1468	3.6016
0..9	0.3167	1.5782	0.2779	1.8583	0.2787	1.7443

Table 4: *Comparison results of neural and statistical classifiers on the MNIST database for Ghorbel descriptors.*

Digit classes	Bayes		CART		ANN	
	Mean	Var x10^{-4}	Mean	Var x10^{-4}	Mean	Var x10^{-4}
2-5	0.3490	8.8772	0.1340	4.5202	0.4040	12
4-7	0.2591	8.5035	0.1000	2.8990	0.3030	7.8485
6-9	0.3418	11	0.1165	4.4975	0.1255	6.1187
0-1-2-3	0.2109	4.6713	0.0933	1.8920	0.0528	1.0979
4-5-6-7	0.5230	4.7403	0.2497	4.6383	0.1497	2.7279
0..9	0.3170	1.9796	0.2779	1.5415	0.2716	1.6939

This study has provided a new conception to compare the ANNs stability results and other classifiers types such as Support Vectors Machine (SVM).

REFERENCES

[1] Atlas, L.E., Cole, R.A., Connor, J.T., El-Sharkawi, M.A., Marks II, R.J., Muthusamy, Y.K., Barnard, E., 1990. Performance comparisons between backpropagation networks and classification trees on three real-world applications. Advances in Neural Information Processing Systems , 622–629.

[2] Berger, J.O., 2013. Statistical decision theory and Bayesian analysis. Springer Science Business Media.

[3] Bowman, A.W., Azzalini, A., 1997. Applied smoothing techniques for data analysis: the kernel approach with s-plus illustrations: the kernel approach with s-plus illustrations. Oxford University Press.

[4] Breiman, L., Friedman, J., Stone, C.J., Olshen, R.A., 1984. Classification and regression trees. CRC press .

[5] Brown, D.E., Corruble, V., Pittard, C.L., 1993. A comparison of decision tree classifiers with backpropagation neural networks for multimodal classification problems. Pattern Recognition 26(6), 953–961.

[6] Curram, S.P., Mingers, J., 1994. Neural networks, decision tree induction and discriminant analysis: An empirical comparison. Journal of the Operational Research Society, 440–450.

[7] Duda, P.O., Hart, P.E., 1973. Pattern Classification and Scene Analysis. Volume 3. Macmillan, Wiley, New York.

[8] Freund, Y., Schapire, R.E., 1996. Experiments with a new boosting algorithm. ICML 96, 148–156.

[9] Fukunaga, K., 2013. Introduction to statistical pattern recognition. Academic press.

[10] Geman, S., Bienenstock, E., Doursat, R., 1992. Neural networks and the bias/variance dilemma. Neural computation 4(1), 1–58.

[11] Ghorbel, F., 1990. Vers une approche unifie des aspects gomtriques et statistiques de la reconnaissance de formes planes. Doctoral dissertation, Rennes.

[12] Ghorbel, F., 1998. Towards a unitary formulation for invariant image description: application to image coding. Annales des telecommunications 53, 242–260.

[13] Hansen, L.K., Salamon, P., 1990. Neural network ensembles. IEEE Transactions on Pattern Analysis Machine Intelligence 10, 993–1001.

[14] Hart, A., 1992. Using neural networks for classification tasks–some experiments on datasets and practical advice. Journal of the Operational Research Society , 215–226.

[15] Huang,W.Y., Lippmann, R.P., 1987. Comparisons between neural net and conventional classifiers. IEEE First International Conference on Neural Networks.

[16] Jolli_e, I., 2002. Principal component analysis. John Wiley Sons, Ltd.

[17] LeCun, Y., Cortes, C., 2010. Mnist handwritten digit database. URL: http://yann.lecun.com/exdb/mnist.

[18] Lippmann, R.P., 1987. An introduction to computing with neural nets. ASSP Magazine, IEEE 4(2), 4–22.

[19] MacKay, D.J., 1992. A practical bayesian framework for backpropagation networks. Neural computation 4(3), 448–472.

[20] Martnez, A.M., Kak, A.C., 2001. Pca versus lda. IEEE Transactions on Pattern Analysis and Machine Intelligence 23(2), 228–233.

[21] Michie, D., Spiegelhalter, D.J., Taylor, C.C., 1994. Machine learning, neural and statistical classification.

[22] Miller, D.W., 1998. Fitting Frequency Distributions: Philosophy Practice. Continuous Distributions. Book Resource.

[23] Morgan, N., Bourlard, H., 1989. Generalization and parameter estimation in feedforward nets: Some experiments. NIPS , 630–637.

[24] Paliwal, M., Kumar, U.A., 2009. Neural networks and statistical techniques: A review of applications. Expert systems with applications 36(1), 2–17.

[25] Parzen, E., 1962. On estimation of a probability density function and mode.

[26] The Annals of Mathematical Statistics , 1065–1076.

[27] Persoon, E., Fu, K.S., 1977. Shape discrimination using fourier descriptors. IEEE Transactions on Systems, Man and Cybernetics 7(3), 170–179.

[28] Razi, M.A., Athappilly, K., 2005. A comparative predictive analysis of neural networks (nns), nonlinear regression and classification and regression tree (cart) models. Expert Systems with Applications 29(1), 65–74.

[29] Rosenblatt, M., 1956. Remarks on some nonparametric estimates of a density function. The Annals of Mathematical Statistics 27(3), 832–837.

[30] Saoudi, S., Ghorbel, F., Hillion, A., 1994a. Non parametric probability probability density function estimation on a bounded support : applications to shape classification and speech coding. Applied statistic Models and Data Analysis 10, 215–231.

[31] Saoudi, S., Ghorbel, F., Hillion, A., 1994b. Some statisticals properties of the kernel di_eomorphism estimator. Applied statistic Models and Data Analysis 10, 39–58.

[32] Shavlik, J.W., Mooney, R.J., Towell, G.G., 1991. Symbolic and neural learning algorithms: An experimental comparison. Machine learning 6(2), 111–143.

[33] Silverman, B.W., 1986. Density estimation for statistics and data analysis. CRC press.

[34] Troudi, M., Ghorbel, F., 2015. Extension de lalgorithme plug-in pour loptimisation du paramtre de lissage de lestimateur du noyau-di_omorphisme. Traitement du signal 31, 321–338.

[35] Tsoi, A.C., Pearson, R.A., 1991. Comparison of three classification techniques: Cart, c4. 5 and multi-layer perceptrons. Advances in Neural Information Processing Systems , 963–969.

[36] Weiss, S., Kulikowski, C., 1991. Computer systems that learn.

[37] Zahn, C.T., Roskies, R.Z., 1972. Fourier descriptors for plane closed curves. IEEE Transactions on Computers 100(3), 269–281.

[38] Zhang, G.P., 2000. Neural networks for classification: a survey. IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews 30(4), 451–462.