Stability evaluation of Neural and statistical Classifiers based on Modified Semi-bounded Plug-in Algorithm

Ibtissem Ben Othman and Faouzi Ghorbel

***Abstract*—**This paper illustrates a new criterion for evaluating neural networks stability compared to the Bayesian classifier. The stability comparison is performed by the error rate probability densities estimation using the modified semi-bounded Plug-in algorithm. We attempt, in this work, to demonstrate that the Bayesian approach for neural networks improves the performance and stability degree of the classical neural classifiers.

***Keywords*—**Bayesian neural networks, error rate density, modified semi-bounded Plug-in algorithm, stability, theoretical error rates.

I. Introduction

IN high dimension spaces, the samples have a limited size, the classification in these spaces requires dimension reduction in the first step. Artificial neural networks (ANNs or NNs) have been in use for some time now, and we can find them working in data classification and non-linear dimension reduction. Actually, the ANNs partially surpassed the statistical methods in the industrial field.

Various experimental comparisons of neural and statistical classifiers have been reported in the literature. Paliwal and Kumar presented in [14] a recent review of these studies which aims to give a useful perception into the neural and statistical methods capabilities. Using these methods to solve complex problems has proven quite successful in many application areas, as the dimension reduction and classification problems. Tam and Kiang showed in [12], by comparing the neural networks and the linear classifiers (Discriminant Analysis, logistic regression and k Nearest Neighbor) for bank bankruptcy prediction in Texas, the benefit of using ANNs in predictive accuracy compared to other classifiers.

A performance evaluation of the neural networks against discriminant analysis was presented by Patuwo and al, in [27], for some classification problems. Their study proved that neural approaches are not better than the discriminant analysis but they are comparable in two-group two-variable problems. Most of researchers compare the neural and statistical techniques by comparing their accuracy prediction while forgetting the NNs instability criterion. This paper studies the stability of different network classifier results compared to the statistical methods.

The stabilities evaluation is based on estimating the error rate probability density function (pdf) of each classifier. The pdf is estimated by applying the Plug-in kernel algorithm, which optimizes its smoothing parameter. The misclassification error is positive value, so we choose to improve the pdf estimation precision by using the modified semi-bounded Plug-in algorithm since pdf support information is known.

The next section summarizes the statistical approaches used for classification and dimension reduction. Section 3 formulates the problem and present different neural networks structures used in this paper. Here we deal with the Bayesian approach for the artificial neural networks. Next, we lead a comparative study between the neural and the statistical approaches. The stability degree is performed by visualizing the results through multivariate Gaussian distributions. Then, we intend to test the classifiers stability and performance for the handwritten digits recognition problem. Finally, we present our works conclusions.

II. Statistical Approaches

The Traditional statistical classification methods are based on the Bayesian decision rule, which presents the ideal classification technique in terms of the minimum of the probability error. However, in the non parametric context, applying Bayes classifier requires the estimation of the conditional probability density functions. It is well known that such task needs a large samples size in high dimension. However, a dimension reduction is required in the first step. The Principal Components Analysis (PCA) and the Linear Discriminant Analysis (LDA) are generally used to reduce the dimension of the feature space. They are applied to the original feature space in order to select a limited number of discrimination directions before applying non parametric Bayes classifier [26]. While PCA seeks for efficient representation directions, the Fisher LDA tries to find efficient

Ibtissem Ben Othman is with the School of Computer Sciences, CRISTAL Laboratory, GRIFT Research Group, Manouba 2010, Tunisia. (e-mail: ben.othman.ibtissem@ gmail.com).

Faouzi Ghorbel is a professor at the School of Computer Sciences and the director of CRISTAL Laboratory, GRIFT Research Group, Manouba 2010, Tunisia. (e-mail: faouzi.ghorbel@ensi.rnu.tn).

discrimination directions. The LDA is commonly preferred over the PCA. In fact, the LDA is able to recognize the different classes, whereas the PCA deals with the data without paying any particular attention to the underlying class [1].

III. Neural Networks Approaches

Among various kinds of artificial neural networks, the Multi-Layered Perceptron (MLP) has become the most widely used network architecture in NNs applications, especially in the classification process. For the MLP, all the nodes (or neurons) and the layers are arranged in a feedforward manner. Thus, all the units in the adjacent layers are connected starting from the input layer to the output one. In between, there is, at least, one hidden layer. As the hidden layers are capable to reduce the data dimension in a non-linear way, these extracted primitives can be classified in the output layer by applying a non-linear separation.

Based on the results from [11, 19], a MLP with one hidden layer is generally sufficient for most problems including the classification. Thus, all used networks in this study will have a unique hidden layer. The number of neurons in the hidden layer could only be determined by experience and no rule is specified. However, the number of nodes in the input and output layers is set to match the number of input and target parameters of the given process, respectively. Thus, the NNs have a complex architecture that the task of designing the optimal model for such application is far from easy.

In order to reduce the difference between the ANN outputs and the known target values, the training algorithm estimates the weights matrices, such that an overall error measure is minimized. The mean squared error (MSE) is the most commonly used performance measure. It is defined as:

$$MSE = \frac{1}{N}\sum_{j=1}^{N}\left(t_j - y_j\right)^2 \qquad (1)$$

where t_j and y_j represent the target and network output values for the j^{th} training sample respectively, and N is the training samples size. For the classification, the NNs training algorithms are mostly employed in a supervised learning process. The proposed technique requires improvements for MLP with the back-propagation algorithm.

A. Neural Networks limitations

As the ANN produces a black box model in terms of only crisp outputs, it cannot be mathematically interpreted as in statistical approaches. The "local representation" is the most regular representation mode of the output layer in pattern recognition. Each output in this representation neuron constitutes one of the classes to which samples can belong [14]. The MLP desired outputs are considered as homogeneous to a posterior probability. Till today, no proof of the quality of this approximation has been presented. However, for the users of these networks, this approximation is presented as a thresholding function to binarize the obtained outputs. Thus, the instability of its classification results as against the statistical ones can be explained by the black box nature, the non proved approximation, the lack of control over its mathematical formulation and the non fixed architecture of the optimal NN model. Therefore, a large variance in its prediction results samples can be introduced, after the training phase, by small changes in the test. In order to reach optimal results, the NN classifier might learn the data very well during the training stage. As a consequence, this can lead to the NN instability: the overfitting may create a high variance while testing the new data. Indeed, researches kept looking for suitable methods to solve the overfitting related problems. The cross validation method, mentioned in [18,21], presents the classical solution. The performance and stability classification may also be improved by combining several neural classifiers [5,8,13,18]. German and al introduced, in [20], the *bias plus variance* decomposition of the prediction error, which presents an interesting solution for the overfitting problem. Intending to reduce the overfitting effect of NNs, a probabilistic interpretation of NNs learning methods has been proposed by Mackay, in [3], thereby using Bayesian techniques.

B. Bayesian Neural Networks

MacKay in [3] originally developed Bayesian methods for NNs, and these methods were reviewed by Bishop in [2] and Mackay in [4]. This approach has been dedicated to improve the conventional NN learning methods while adding a penalty term to the classical error function. The resulting function can be described as follows:

$$S(w) = E_D(w) + \mu E_w(w) \qquad (2)$$

where $E_D(w)=MSE$ and w presents the parameters of size m. The penalty term $E_w(w) = \frac{1}{2}\sum_{i=1}^{m} w_i^2$ controls the model complexity. This approach consists to find the optimal value of the regularization coefficient μ which gives the best tradeoff between the overfitting and the underfitting problems. To find this optimal value, probabilistic interpretation of NN learning, which controls automatically its complexity, can be used.

In the Bayesian approach, to each NN parameter w_i (weights, biases, number of neurons, NN outputs, etc), get assigned a probability density function (pdf). This pdf is initially affected to a prior distribution, and once the data have been observed, it will be converted to a posterior distribution using Bayes theorem.

IV. Performance and Stability Comparison

Some classifiers are instable, small changes in their training sets or in constructions may cause large changes in their classification results. Therefore, an instable model may be too dependent on the specific data and has a large variance. 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. The classifier, whose curve is on the left, is the most efficient one. Also, a classifier with the largest density curve is the least stable one. Therefore, a good model should find a balanced equilibrium between the error rate bias and variance.

A. Non-parametric density estimation

The first step before comparing is to train the two classifiers, then we proceed by measuring the error rate produced by each classifier with each one of N independent test sets. Let's consider $(X_i)_{1\le i\le N}$ the N generated error rates of a given classifier (Bayes or ANN). These error rates are random variables which have the same probability density function (pdf), $f_X(x)$. These $(X_i)_{1\le i\le N}$ are supposed to be independent and identically distributed.

We suggest to use the kernel method proposed in [10,25], to estimate the pdf of the error rates for each classifier. In this method, an approximation of the integrated mean square error (IMSE) is optimized in order to estimate the involved smoothing parameters h_N. The kernel estimator of the probability density is defined by:

$$\hat{f}_N(x) = \frac{1}{Nh_N}\sum_{i=1}^{N} K\left(\frac{x - X_i}{h_N}\right) \tag{3}$$

In our study, $K(.)$ is chosen as the Gaussian kernel:

$$K(x) = \frac{1}{\sqrt{2\pi}}\exp\left(-\frac{1}{2}x^2\right) \tag{4}$$

It's very important to choose the optimal smoothing parameter h_N^*. Moreover, different methods were proposed to minimize the integrated mean square error ($IMSE \approx \frac{M(K)}{Nh_N} + \frac{J(f(X))h_N^4}{4}$) to determine the optimal bandwidth. The smoothing parameter h_N^* becomes as follows:

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

where;

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

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

B. Conventional Plug-in algorithm

The choice of the optimal value for the smoothing parameter will determine the goodness level of the estimation. It seems very difficult to find a direct resolution for the equation (4) to calculate this optimal value. So we choose to determine it by the recursive resolution: The Plug-in algorithm. Actually, a fast variant of known conventional Plug-in algorithm has been developed [24]. In order to approximate the function J(f), a double derivation of the kernel estimator analytical expression is applied directly. The different steps of conventional Plug-in algorithm are as 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%).

C. Modified semi-bounded Plug-in algorithm

The set of the observed error rates $(X_i)_{1\le i\le N}$ of each classifier is a set of positive values. There won't be any interest to use the kernel density estimation method in this situation. During the estimation of the probability densities, which are defined in a bounded or semi-bounded space $U \subset \Re^d$, some convergence problems may occur at the edges: the Gibbs phenomenon. Several researchers have tried to solve this issue and some methods got described in order to estimate the probability densities under topological constraints on the support. Two interesting solutions mentioned in [22, 23] present interesting results: the orthogonal functions method and the kernel diffeomorphism method. The kernel diffeomorphism method is based on a suitable variable change by a C1-diffeomorphism. Although, the value of the smoothing parameter must be maximized, otherwise there won't be any warranty to get a good estimation quality. The Plug-in diffeomorphism algorithm which is a generalization of the conventional Plug-in algorithm [16] is used to perform the optimization of the smoothing parameter.

For complexity and convergence reasons, we suggest a modified semi-bounded Plug-in algorithm. This algorithm version is based on the variable change of the positive error rates: $Y = Log(X)$. In order to specify a new classification quality measure, we perform a sequence of three steps:

Step 1: using the variable change $Y = Log(X)$, the kernel estimator expression becomes:

$$\hat{f}_Y(y) = \frac{1}{Nh_N^*}\sum_{i=1}^{N} K\left(\frac{y - Y_i}{h_N^*}\right) \tag{6}$$

Step 2: iterate the conventional Plug-in algorithm for the transformed data.

Step 3: compute $\hat{f}_X(x) = \frac{\hat{f}_Y(Logx)}{x}$

V. SIMULATIONS

The comparison between the statistical and neural classifiers used in the present work (PCA-Bayes, Fisher-Bayes, MLP and Bayesian NN) is summarized by the multivariate Gaussian mixture classification problem. With the same train set (including 1000 samples for each class), we look to find the optimum transformation that represents the dimension reduction for both PCA and LDA methods before applying the Bayesian rule, and then to fix the optimal NN model parameters for both classical and Bayesian NN.

After the training phase, we generate 100 supervised and independent test sets (including 1000 samples for each class). For each test set, the classifier performance is evaluated by its error rate calculated from the confusion matrix. The error rate probability densities, retained for both approaches, are

estimated using the modified semi-bounded Plug-in algorithm in order to compare the stability degree.

Figure 1 shows the estimated error rate probability densities generated for the different classifiers on a mixture of two homoscedastic Gaussians (Fig.1.a and Fig.1.b) and two heteroscedastic Gaussians (Fig.1.c). The classifiers performance and stability were often compared with the presentation of the theoretical error probability. This latter is calculated using the discriminant functions. Figure 2 illustrates the results of two heteroscedastic superposed Gaussians (Fig.2.a and Fig.2.b) and two truncated ones (Fig.2.c). The stability and performance of the classifiers are also analyzed by presenting their error rate means and variances in table 1 and table 2.

By analyzing the results shown in Figure 1 and table 1, the statistical classifiers (ACP-Bayes and Fisher-Bayes) admit the smallest error rate means: they are proved to be more efficient than the neural ones. However, the error rate probability density functions of the neural models are on the left for the complex cases of the two heteroscedastic superposed Gaussians and the truncated ones. We deduce then the efficiency of these models. Although, the neural approach remains the least stable classifier that presents the greatest variance and thus the widest curve for all the six cases. However, the Bayesian approach of NN improves its stability and performance

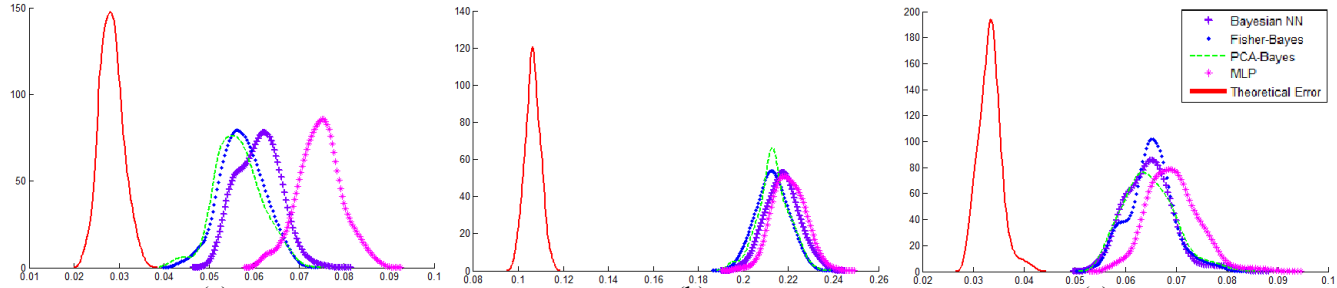


Fig. 1 Error rate densities of PCA-Bayes (in green(--)), Fisher-Bayes (in blue(..)), MLP (in pink(*)), Bayesian NN (in purple(+)) and theoretical error rates (in red).

Table 1 Comparison results of ANN, statistical classifiers and theoretical error rates.

Cases	Distributions		Theoretical error		PCA-Bayes		Fisher-Bayes		MLP		Bayesian NN	
	Gaussian 1	*Gaussian 2*	*Mean*	*Variance*	*Mean*	*Variance*	*Mean*	*Variance*	*Mean*	*Variance*	*Mean*	*Variance*
a	$\mu_1=(0,..,0),\sum_1=I_{10}$	$\mu_2=(1,..,1),\sum_2=I_{10}$	0.0283	0.0598	0.0564	0.2253	0.0573	0.2205	0.0746	0.2267	0.0610	0.1984
b	$\mu_1=(1.5,..,1.5),\sum_1=I_{10}$	$\mu_2=(2,..,2),\sum_2=I_{10}$	0.1065	0.0111	0.2134	0.0491	0.2128	0.0489	0.2201	0.0575	0.2174	0.0517
c	$\mu_1=(1.5,..,1.5),\sum_1=2*I_{10}$	$\mu_2=(3,..,3),\sum_2=3*I_{10}$	0.0332	0.0051	0.0647	0.0239	0.0650	0.0195	0.0693	0.0246	0.0646	0.0214

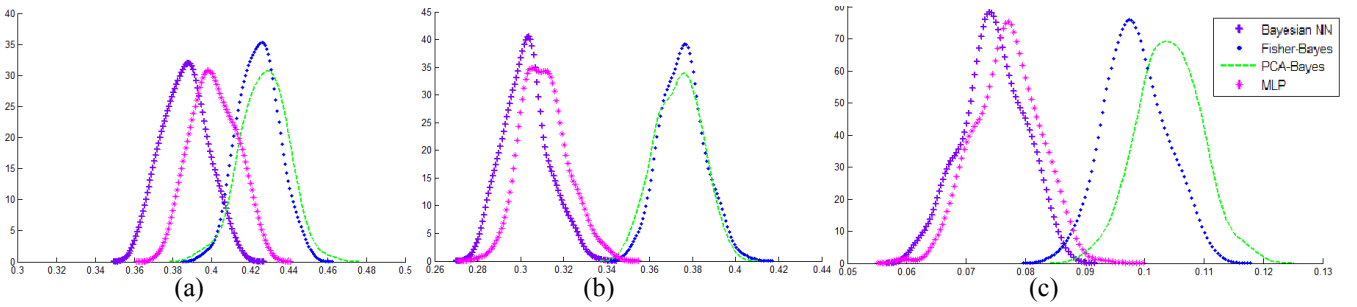


Fig. 2 Error rate densities of PCA-Bayes (in green(--)), Fisher-Bayes (in blue(..)), MLP (in pink(*)) and Bayesian NN (in purple(+)).

Table 2 Comparison results of ANN and statistical classifiers.

Cases	Distributions		PCA-Bayes		Fisher-Bayes		MLP		Bayesian NN	
	Gaussian 1	*Gaussian 2*	*Mean*	*Variance*	*Mean*	*Variance*	*Mean*	*Variance*	*Mean*	*Variance*
a	$\mu_1=(1,..,1),\sum_1=2*$Identity	$\mu_2=(1,..,1),\sum_2=3*$Identity	0.4271	0.1414	0.4241	0.1037	0.4012	0.1289	0.3863	0.1261
b	$\mu_1=(0,..,0),\sum_1=$Identity	$\mu_2=(0,..,0),\sum_2=2*$Identity	0.3734	0.1054	0.3753	0.0983	0.3110	0.1140	0.3026	0.1094
c	$\mu_1=(0,0,0)$ $\sum_1=[0.06\ 0\ 0$ $0\ 0.01\ 0$ $0\ 0\ 0.01\ ]$	$\mu_2=(0.1,0.1,0.1)$ $\sum_2=[0.01\ 0\ 0$ $0\ 0.06\ 0$ $0\ 0\ 0.05\ ]$	0.1041	0.2804	0.0985	0.2612	0.0768	0.2877	0.0745	0.2709

VI. Application to Handwritten Digit Recognition

In this section, we study the handwritten digit recognition problem, which 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 MNIST database. For the training and test sets, we select randomly, from the MNIST training and test sets respectively, single digit images (the both sets contains 1000 images for the 10 digit classes). Random sampling images are shown in Fig.3.

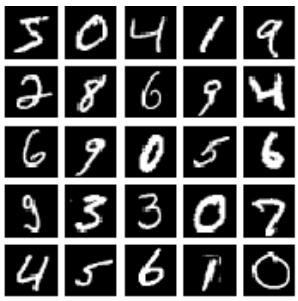
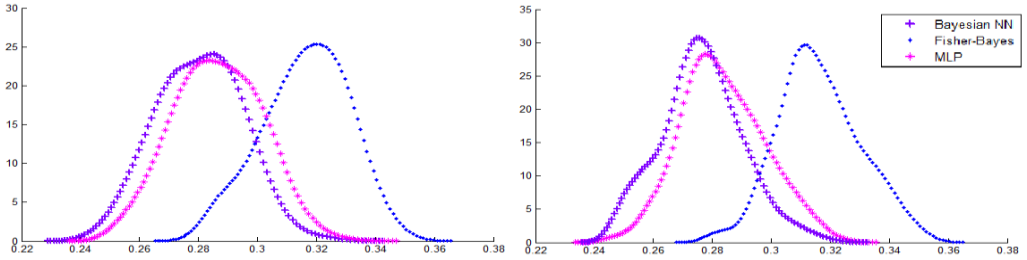


Fig. 3 Random sample images of MNIST database. [28]

Choosing the suitable features represents on itself a quite difficult step in handwritten digit recognition problems. We must verify that the chosen features validate, under the geometrical transformations, a non-exhaustive set of criteria such as fast computation, stability, completeness, powerful discrimination and invariance. Ghorbel in [6] presented an invariant descriptors family which satisfies the various criteria listed above. Therefore, we will describe each image by this type of invariants and by Fourier descriptors (FD). We select a high descriptors size (D = 14).

The train set has two goals. The first goal is to find the optimum transformation that represents the two dimension reduction for LDA Fisher before applying the Bayesian rule. The second goal is to fix the parameters of the optimal NN model for both classical and Bayesian NN. Thus, we have used a MLP and a Bayesian NN with three layers having, respectively, 14, 12 and 10 neurons. We intend to compare the classifiers stability by evaluating their respective performances for 100 times using the k-folds cross validation algorithm (k=10 in our study). We use the CV algorithm from the MNIST test set to select the test sets (N=1000 images for each class). With these sets, we calculate the misclassification rate (MCR) of each classifier.

Figure 4 shows the classifiers error rate probability densities estimated using the modified semi-bounded Plug-in algorithm for Fourier descriptors and Ghorbel descriptors, respectively. In table 3, we summarize the MCR means and variances obtained for the two types of descriptors using the three classifiers (Fisher-Bayes MLP and Bayesian NN). We note the best results for these classifiers with Ghorbel descriptors. The results show also the performance of the MLP against the Bayesian classifier, but the superiority of its error rate variances proves their low stability against the statistical approaches. For these two complex cases, the linear reduction dimension method (Fisher LDA) fails to find the optimal projection subspace. Whereas, the Bayesian and classical NN perform well due to their non linear reduction dimension capability. Although and in addition to providing a better performance, the Bayesian NN is relatively more stable than the classical NN. Thus, we can approve that the stability and performance of the conventional NN increases with the Bayesian approach for ANNs.



(a) (b)

Fig. 4 Error rate densities of Fisher-Bayes (in blue(..)), MLP (in pink(*)) and Bayesian NN (in purple(+)) for Fourier descriptors (in the left) and Ghorbel descriptors (in the right).

Table 3: Comparison results of neural and Bayesian classifier on the MNIST database.

	Fourier Descriptors		Ghorbel Descriptors	
	Mean	**Variance**	**Mean**	**Variance**
Fisher-Bayes	0.3163	0.1957	0.3157	0.1877
MLP	0.2864	0.2027	0.2836	0.1892
Bayesian NN	0.2803	0.1951	0.2765	0.1852

VII. Conclusions

This paper provided a novel approach to comparing neural and Bayesian classifiers. The use of the semi-bounded Plug-in algorithm tends to be a good criterion for the stability comparison of the different classifiers. This algorithm produces a sufficient precision for the densities estimation and the stability aspect.

By stochastic simulations, we proved that the statistical approaches are more stable compared to the neural networks. Also we found that using the Bayesian approach to model NNs improves their performance and stability. In this study, we provided a new conception to compare the stability results of the neural networks against other classifiers classes. Combining the classifiers to improve their stabilities can be a quiet interesting point to focus on in our future works.

References

[1] A.M. Martinez, A.C. Kak, "PCA versus LDA", IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 23, no. 2, pp. 228-233, 2001.

[2] C. M. Bishop, "Neural networks for pattern recognition". *Oxford university press*, 1995.

[3] D.J.C. MacKay, "A practical Bayesian framework for back-propagation networks". *Neural Comput*, 4(3), 448–72, 1992.

[4] D.J.C. Mackay, "Bayesian methods for neural networks: theory and applications", 1995.

[5] D.W. Miller, "Fitting frequency distributions", *Book Resource*, second edition, 1998.

[6] F. Ghorbel, "Towards a unitary formulation for invariant image description: application to image coding". *Annals of telecommunication*, vol. 53, France, 1998.

[7] F. Menasri, N. Vincent, E.Augustin, "Reconnaissance de chiffres farsi isolés par réseau de neurones à convolutions", CIFED'08, pp. 127-132, 2008.

[8] G.P. Zhang, "Neural networks for classification: a survey". *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions*, vol. 30, no 4, p. 451-462, 2000.

[9] I.T. Jolliffe, "Principal Component Analysis". *Springer-Verlag*, 1986.

[10] K. Fukunaga, "Introduction to Statistical Pattern Recognition", second ed. *Academic Press*, 1990.

[11] K. Steven, Rogers, M. Kabrisky, "An Introduction to Biological and Artificial Neural Networks for Pattern Recognition", *SPIE Optical Engineering Press*, vol. 4, 1991.

[12] K.Y. Tam, M.Y. Kiang, "Managerial applications of neural networks: The case of bank failure predictions". *Management Science*, vol. 38, no. 7, pp. 926–947, 1992.

[13] L. K. Hansen and P. Salamon, "Neural network ensembles," *IEEE Trans.Pattern Anal. Machine Intell.*, vol. 12, no. 10, pp. 993–1001, 1990.

[14] M. Paliwal, U.A. Kumar, "Neural networks and statistical techniques: A review of applications", *Expert Syst. Appl*., vol. 36, no. 1, pp. 2–17, 2009.

[15] M. P. Perrone and L. N. Cooper, "When networks disagree: Ensemble methods for hybrid neural networks", *Neural Networks for Speech and Image Processing*, R. J. Mammone, Ed. London, U.K.: Chapman & Hall, 1993, pp. 126–142.

[16] M. Troudi, F. Ghorbel, "The generalised Plug-in algorithm for the diffeomorphism kernel estimate". *International Conference on Systems, Control, Signal Processing and Informatics*, 2013.

[17] N.L. Johnson, S. Kotz, "Distribution in statistics: Continuous univariate distribution", *Wiley-interscience*, vol. 1, New York, 1969.

[18] N. Morgan and H. Bourlard, "Generalization and parameter estimation in feedforward nets: Some experiments," Adv. *Neural Inform. Process. Syst.*, vol. 2, pp. 630–637, 1990.

[19] R. Lepage, B. Solaiman, "Les réseaux de neurones artificiels et leurs applications en imagerie et en vision par ordinateur". Montréal, 2003.

[20] S. Geman, E. Bienenstock, and T. Doursat, "Neural networks and the bias/variance dilemma", *Neural Comput.*, vol. 5, pp. 1–58, 1992.

[21] S.M. Weiss and C.A. Kulilowski, "Computer Systems that Learn". San Mateo, CA: Morgan Kaufmann, 1991.

[22] S. Saoudi, F. Ghorbel, A. Hillion, "Some statistical properties of the kernel-diffeomorphism estimator", *Applied Stochastic Models and Data Analysis Journal*, Vol. 13, no. 1, pp. 39-58, 1997.

[23] S. Saoudi, F.Ghorbel, A.Hillion, "Nonparametric probability density function estimation on a bounded support: applications to shape classification and speech coding", *Applied Stochastic Models and Data Analysis Journal*, vol. 10, no. 3, pp. 215–231, 1994.

[24] S. Saoudi, M. Troudi, F. Ghorbel, "An iterative soft bit error rate estimation of any digital communication systems using a non parametric probability density function", *Eurasip Journal on wirless Communications and Networking*, 2009.

[25] U.A. Kumar, "Comparison of neural networks and regression analysis: A new insight". *Expert Systems with Applications*, vol. 29, no. 2, pp. 424–430, 2005.

[26] W. Drira, F. Ghorbel, "Vers un classifieur de Bayes pour les grandes dimensions par analyse discriminante basée sur un estimateur non paramétrique de la L2-mesure de dépendance probabiliste", *Récentes avancées en Reconnaissance de Formes Statistique*, ed. Art-pi, Tunis, www.arts-pi.org.tn, 2012.

[27] W. Patuwo, M.Y. Hu, M.S. Hung, "Two-group classification using neural networks". *Decision Sciences*, vol. 24, pp. 825–845, 1993.

[28] www.wordpress.com.