International Journal of Neural Networks and Advanced Applications
DOI: 10.46300/91016.2022.9.2 Volume 9, 2022

The Use of the Modified Semi-bounded Plug-in Algorithm to Compare Neural and Bayesian Classifiers Stability

[1]Ibtissem Ben Othman, [2]Faouzi Ghorbel
[1]School of Computer Sciences, CRISTAL Laboratory, GRIFT Research Group, Manouba 2010, Tunisia.
[2]School of Computer Sciences and the director of CRISTAL Laboratory, GRIFT Research Group, Manouba 2010, Tunisia.

Abstract—Despite of the widespread use of the neural networks in the industrial applications, their mathematical formulation remains difficult to analyze. This explains a limited amount of work that formally models their classification volatility. Referring to the statistical point of view, we attempt in this work to evaluate the classical and Bayesian neural networks stability degree compared to the statistical methods stressing their error rate probability densities. The comparison based on this new criterion is performed using the modified semi-bounded Plug-in algorithm.

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

I. Introduction

As the samples have a limited size, the classification in high dimension spaces remains one of the essential problems for pattern recognition. Therefore, dimension reduction is often required in the first step. The Artificial neural networks (ANNs or NNs) have been commonly used to classify the data and to solve any non-linear dimension reduction. Actually, the ANNs partially substituted the statistical methods in the industrial field. 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. 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 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], [26].

Many authors carried out many comparison studies of neural and statistical classifiers. A recent review of these studies is presented in [14]. It aims to give a useful insight into the neural and statistical methods capabilities. These methods have widely been used to solve complex problems and have proven quite successful in many applications, as the dimension reduction and classification problems. Tam and Kiang compare in [12] the neural networks and the linear classifiers (Discriminant Analysis, logistic regression and k Nearest Neighbour) for bank bankruptcy prediction in Texas. They showed that ANNs offered better predictive accuracy than other classifiers. Patuwo and al evaluate, in [27], the neural networks performance against discriminant analysis for some classification problems. They proved that neural approaches are comparable but not better than the LDA in two-group two-variable problems.

In order to compare the neural and statistical techniques, most of researchers try to compare their accuracy prediction while forgetting the instability criterion of NNs . This paper studies the stability of different network classifier results compared to the statistical methods. By estimating the error rate probability density function (pdf) of each classifier, we evaluate their stabilities. In order to estimate its pdf, we apply the Plug-in kernel algorithm, which optimizes its smoothing parameter. The miss classification error is positive value, so we opt for the modified semi-bounded Plug-in algorithm to improve the pdf estimation precision since pdf support information is known.

So, the present work will be organized as following: First, we start by presenting the main topic and briefly introduce the neural approaches. Here we deal with the Bayesian approach for the artificial neural networks. Then we lead a comparative study between the neural and the statistical approaches. Here we focus on the stability degree and visualize the results through stochastic simulation of particular distributions (for example: Gaussian, Gamma, Beta, etc…). Finally, we intend to test the classifiers stability and performance for the handwritten digits recognition problem.

II. NEURAL APPROACHES

In pattern recognition, the neuronal networks can be categorized into three different types, depending on their application objectives (size reduction, classification or both). The first networks category is the features extractors NNs. It aims to reduce the learning set dimensionality thereby extracting the relevant primitives. The classifier NNs is the second networks category. Its main duty is to classify the extracted features regardless of the dimensionality. The third networks category is the mixed NNs, which present a combination of both defined types. Once the first networks layers carry out the primitive extraction, the last layers classify the extracted features. An interesting example is the multilayer NNs that uses the back-propagation algorithm. Thanks to its several hidden layers, the multilayer perceptron (MLP) can reduce non-linearly the data dimension and extract its relevant characteristics. Finally, a linear separation is applied to classify these extracted primitives in the output layer. Based on the results from [19] and [11], a MLP with one hidden layer is generally sufficient for most problems including classification. Therefore, all used networks, in our study, will have a unique hidden layer. The number of neurons in this 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 and designing the optimal model for such application is not so easy.

By estimating the weights matrices, the training algorithm aims to reduce the difference between the ANN outputs and the known target values, such that an overall error measure is minimized. The most commonly used performance measure is the mean squared error (MSE) 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. The NNs training algorithms, for the classification, are mostly employed in a supervised learning process. In the proposed technique, improvements will be required for MLP with the back-propagation algorithm.

A. Neural Networks limitations

The ANN produces a black box model in terms of only crisp outputs, and hence cannot be mathematically interpreted as in statistical approaches. The most common representation mode of the output layer in pattern recognition is the "local representation" one. In this representation, each output neuron represents one of the classes to which samples can belong [14]. The MLP desired outputs provide a value close to 1 for the class neuron of the considered object and values close to zero for the other nodes. However, the ANNs outputs values are not similar to those desired, and they can even be negative values. For a discrete representation {0,1} of output neurons, Jolliffe normalizes, in [9], the obtained outputs, and considers the new normalized values as a posterior probability. In [7], the authors have used the Softmax transfer function, which ensures that the ANN outputs are homogeneous to a posterior probability. Till today, the quality of this approximation has never been proved. However, users of these networks are based on this approximation as a thresholding function to binarize the obtained outputs. This non proved approximation, the black box nature, the lack of control over its mathematical formulation and the non fixed architecture of the optimal NN model explain the instability of its classification results as against the statistical ones. Thus, after the training phase, small changes in the test samples can introduce a large variance in its prediction results. During the training phase, the NN classifier might learn the data very well in order to reach best results. As a consequence, this can lead to the NN instability; the overfitting may create a high variance while testing the new data. Indeed, the overfitting related problems got the attention of the literature and researches kept looking for suitable solutions. The classical one is the cross validation method [18], [21]. Combining several neural classifiers is another solution which may improve the performance and stability classification [8], [5], [13], [18]. The *bias plus variance* decomposition of the prediction error, introduced by German and al in [20], presents a solution for the overfitting problem. In order to reduce the overfitting effect of NNs, Mackay has proposed, in (Mackay, 1992), a probabilistic interpretation of neural networks learning methods, thereby using Bayesian techniques.

B. Bayesian Neural Networks

The Bayesian approach for NNs was originally developed by Mackay in [3] and reviewed by Bishop in [2] and Mackay in [4]. This approach has been devoted to improving the conventional NN learning methods while adding a penalty term to the classical error function. The resulting function is defined by:

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

The penalty term $E_w(w) = \frac{1}{2}\sum_{i=1}^{m} w_i^2$ (where m is the total number of parameters) controls the model complexity. The main idea is to find the optimal value of the regularization coefficient μ that gives the best tradeoff between the overfitting and the underfitting problems. This optimal value can be found by using probabilistic interpretation of NN learning which controls automatically its complexity.

The Bayesian approach assigns a probability density function (pdf) to each NN parameter w_i (weights, biases, number of neurons, NN outputs, etc). 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:

$$p(w|t,x) = \frac{p(t,x|w)\,p(w)}{p(t,x)} \qquad (3)$$

III. Approaches Comparison

Most of the current research almost exclusively uses the criterion of prediction accuracy, while evaluating classifiers and comparing their performance, but ignoring the fact that the NNs present unstable results. In this paper, a new approach to comparing neural and statistical classifiers is proposed. In addition to the prediction accuracy criterion, a stability comparison based on estimating classifiers error rate probability density functions is presented.

A. Error rate density estimation

We start first by training the two classifiers to be compared, and then measure the error rate produced by each classifier on each one of N independent test sets. Let $(X_i)_{1\le i\le N}$ be the N generated error rates of a given classifier (Bayes or ANN). These error rates $(X_i)_{1\le i\le N}$ are random variables having the same probability density function (pdf), $f_X(x)$. The $(X_i)_{1\le i\le N}$ are assumed to be independent and identically distributed.

We suggest to estimate the pdf of the error rates for each classifier using the kernel method proposed in [10] and [25], where the involved smoothing parameters h_N are estimated by optimizing an approximation of the integrated mean square error (IMSE). The kernel estimator of the probability density is defined as follows:

$$\hat{f}_N(x) = \frac{1}{Nh_N}\sum_{i=1}^{N} K\left(\frac{x - X_i}{h_N}\right) \qquad (4)$$

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

$$K(x) = \frac{1}{\sqrt{2\pi}}\exp(-\frac{1}{2}x^2) \qquad (5)$$

The choice of the optimal smoothing parameter h_N^* is very important. Moreover, Researchers have introduced different methods that minimize the integrated mean square error ($IMSE \approx \frac{M(K)}{Nh_N} + \frac{J(f(X))h_N^4}{4}$) to define the optimal bandwidth. The smoothing parameter h_N^* becomes as follows:

$$h_N^* = N^{-1/5}[J(f_X)]^{-1/5}[M(K)]^{+1/5} \qquad (6)$$

where;

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

and

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

The goodness of estimation depends on choosing an optimal value for the smoothing parameter. Calculating its optimal value, with a direct resolution of the equation (4), seems very difficult. We opt for the recursive resolution: The *Plug-in* algorithm. Actually, a fast variant of known conventional Plug-in algorithm has been developed [24]. It applies directly a double derivation of the kernel estimator analytical expression in order to approximate the function $J(f)$.

B. Modified semi-bounded Plug-in algorithm

The set of observed error rates $(X_i)_{1\le i\le N}$ of each classifier is a set of positive values. In this case, the kernel density estimation method is not that attractive. When estimating the probability densities, which are defined in a bounded or semi-bounded space $U \subset \Re^d$, we will encounter convergence problems at the edges : the Gibbs phenomenon. Several authors have tried to solve this issue and presented some methods to estimate the probability densities under topological constraints on the support. The orthogonal functions method and the kernel diffeomorphism method are two interesting solutions [22], [23]. The kernel diffeomorphism method is based on a suitable variable change by a C1-diffeomorphism. Although, it is important to maximize the value of the smoothing parameter in order to ensure a good estimation quality. The optimization of the smoothing parameter is performed by the Plug-in diffeomorphism algorithm which is a generalization of the conventional Plug-in algorithm [16].

For complexity and convergence reasons, we propose in this paper 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 define new classification quality measure, a sequence of three steps is performed:

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

$$\hat{f}_Y(y) = \frac{1}{Nh_N^*}\sum_{i=1}^{N} K\left(\frac{y - Y_i}{h_N^*}\right) \qquad (7)$$

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

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

The modified semi-bounded Plug-in algorithm produces a sufficient precision for the densities estimation and the stability aspects. It tends to be a good criterion for the stability comparison of the different classifiers and reduction algorithms.

C. 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. 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, 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. This criterion is basic for any stability and performance analysis of each classifier.

IV. Simulations

For the simulation phase, we propose a binary classification adapted to a mixture of two different distributions. We generate one train set including 1000 samples for each class. With this train set, 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 MLP and Bayesian NN.

In order to analyze the stability of the different approaches, 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.

The comparison between the statistical and neural classifiers used in the present work (PCA-Bayes, Fisher-Bayes, MLP and Bayesian NN) is first summarized by the Gaussian mixture classification problem. Figure 1 shows the estimated error rate probability densities generated for the different classifiers on a mixture of two homoscedastic Gaussians (Figure1.a), two heteroscedastic Gaussians (Figure1.b), two superposed Gaussians (Figure1.c) and two truncated ones (Figure1.d). The stability and performance of the classifiers are also analyzed by presenting their error rate means and variances in table 1.

The first two cases ((a) and (b)) in figure 1 and table 1, show that the statistical classifiers (ACP-Bayes and Fisher-Bayes) are more efficient than the neural ones (they admit the smallest error rate means). However for the last complex cases of the two heteroscedastic superposed Gaussians (c) and the truncated ones (d), the error rate probability density functions of the neural models are on the left. We conclude that these models are the most efficient. Although, the neural approach remains the least stable classifier that presents the greatest variance and thus the widest curve. For these two complex cases, the linear reduction dimension methods (PCA and Fisher) fail to find the optimal projection subspace. Whereas, the Bayesian and classical NN perform well due to their non linear reduction dimension capability.

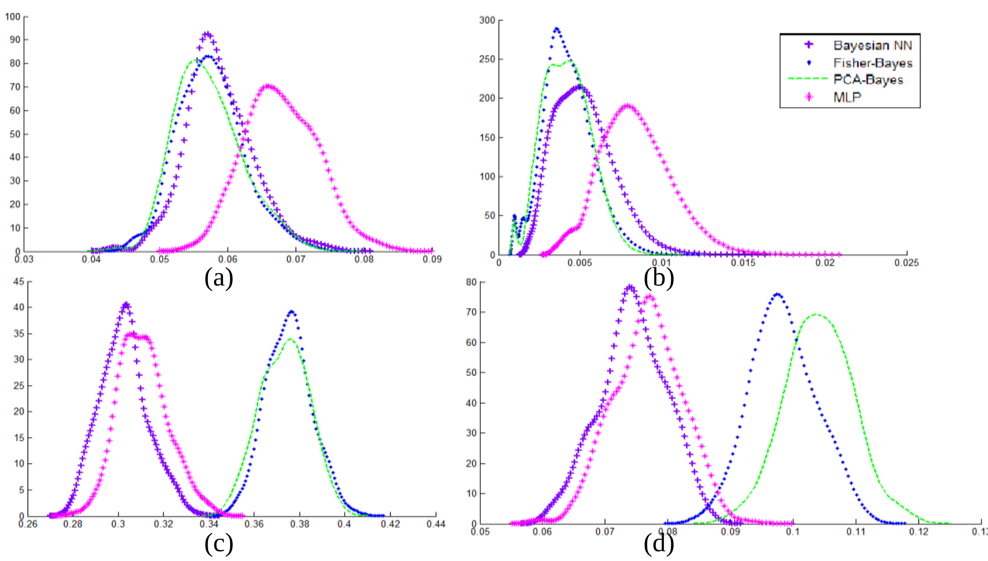

Figure 1: Error rate densities of PCA-Bayes (in green(--)), Fisher-Bayes (in blue(..)), MLP (in pink(*)) and Bayesian NN (in purple(+)).

Table 1: 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	$\mu_1=(1,..,1),\sum_1=$Identity	$\mu_2=(2,..,2),\sum_2=$Identity	0.0572	0.2287	0.0575	0.2223	0.0679	0.2697	0.0587	0.2284
b	$\mu_1=(0,..,0),\sum_1=$Identity	$\mu_2=(2,..,2),\sum_2=2*$Identity	0.0042	0.1721	0.0043	0.1972	0.0084	0.4231	0.0052	0.2590
c	$\mu_1=(0,..,0),\sum_1=$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
d	$\mu_1=(0,0,0)$ $\sum_1=$[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

DOI: 10.46300/91016.2022.9.2 Volume 9, 2022

To analyze the classifiers stability, the comparison study is often illustrated by a mixture of univariate distributions according to the Pearson System. This system presents a set of eight families of distributions, including Gaussian, Gamma and Beta ones. The Pearson system distributions are generally qualified by their four parameters; mean (μ), variance (σ^2), skewness (β_1) and kurtosis (β_2). Further details about the Pearson system can be found in [5] and [17].

The classifiers performances and stabilities are compared in the sense of their error rate means and variances in table 2. Figure 2 shows the estimated error rate densities generated for the different classifiers (the figure cases (a,b,…,f) correspond to the cases in table 2).

By analyzing the figures and the table above, we note that the neural approach admit the greatest variances, we can confirm that it is less stable than the statistical one.

Using the simulations concerning the different kinds of stochastic distributions, we illustrate the better stability of the Bayesian classifier against the neural one. In addition, the statistical approaches are proved to perform better than the neural networks in the simple classification problems. However, the results prove that the neural classifier performs better if the classification task tends to become complex (the truncated and the superposed distributions for the Gaussian simulations and the Pearson System distributions). Although the Bayesian NN provides a better performance and is relatively more stable than the classical NN, it remains less stable than the Bayesian classifier. Thus, we can confirm that the Bayesian approach for ANNs improves the stability and performance of the conventional NN.

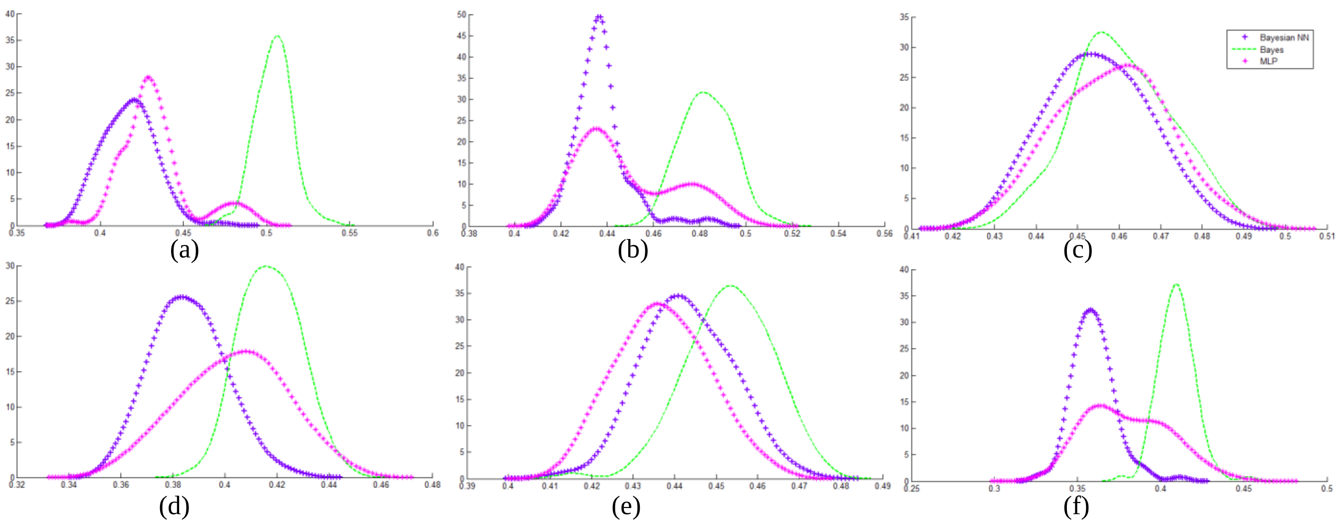

Figure 2: Error rate densities of Bayesian classifier (in green(..)), MLP (in pink(*)) and Bayesian NN (in purple(+)).

Table 2: Comparison results of ANN and Bayesian classifier for the Pearson system distributions.

Cases	**Distributions**										**Bayes**		**MLP**		**Bayesian NN**	
	Distribution 1					***Distribution 2***										
	Type	μ	σ^2	β_1	β_2	***Type***	μ	σ^2	β_1	β_2	***Mean***	***Variance***	***Mean***	***Variance***	***Mean***	***Variance***
a	8	40	100	0	3	3	60	100	1.26	4.9	0.5044	0.1154	0.4328	0.4460	0.4172	0.2274
b	1	10	100	0.63	3.28	3	30	100	1.26	4.9	0.4831	0.1151	0.4491	0.4501	0.4374	0.1369
c	8	40	100	0	3	2	60	100	0	2.34	0.4607	0.1403	0.4581	0.1687	0.4548	0.1414
d	6	40	100	1.32	5.35	2	60	100	0	2.34	0.4174	0.1213	0.4030	0.3912	0.3860	0.1950
e	4	40	100	1.71	7.3	6	60	100	1.32	5.35	0.4531	0.1043	0.4372	0.1162	0.4430	0.1077
f	4	40	100	1.71	7.3	2	60	100	0	2.34	0.4096	0.1184	0.3802	0.5564	0.3597	0.1643

V. 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. This database contains 60,000 training images and 10,000 test ones.

For the training and test sets, we select randomly, from the MNIST training and test sets respectively, single digit images from '0 'to '3'. Each class contains 1000 images for the both sets. Some images are shown in Fig.3.

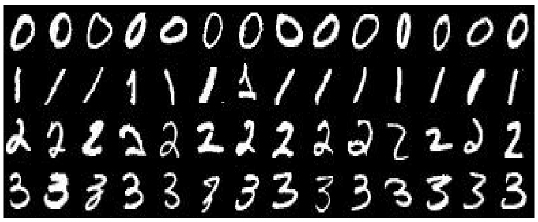

Figure 3: Sample images of MNIST database.

The most difficult step in handwritten digit recognition is to choose the suitable features. The chosen features must necessarily verify a non-exhaustive set of criteria such as stability, completeness, fast computation, powerful discrimination and invariance under the geometrical transformations. The invariant descriptors family proposed by Ghorbel in [6] satisfies the various criteria cited above. Thus, each image will be described by this type of invariants and Fourier descriptors (FD). The selected descriptors size is high (D = 14). In order to apply Bayesian rule, dimension reduction becomes necessary. The transformation matrices are estimated for both PCA and LDA methods from the training set, which transform the data to the appropriate dimensions subspace (two dimensions in our study). For the neural approach, we have used a MLP and a Bayesian NN with three layers having, respectively, 14, 10 and 4 neurons. In order to compare the classifiers' stability, we evaluate the classifiers' performance for 100 times using the k-folds cross validation algorithm (k=10 in our study). The misclassification rate (MCR) of each classifier is calculated on the test sets selected by the CV algorithm from the MNIST test set (N=400 images for each class). Figures 4.a and 4.b represent the MCR probability estimation using the four classifiers (PCA-Bayes, Fisher-Bayes MLP and Bayesian NN) for Fourier descriptors and Ghorbel descriptors, respectively.

In order to obtain meaningful comparison between the different types of classifiers, we evaluate their performances and stability degrees. Figure 4 shows the error rate probability densities estimated using the modified semi-bounded Plug-in algorithm. This algorithm is qualified by its sufficient precision on the stability aspects. In table 3, we summarize the MCR means and variances obtained for the two types of descriptors using the four classifiers. We note that these classifiers give the best results for Ghorbel descriptors. The MLP shows performance against the Bayesian classifier, but the superiority of its error rate variances proves that it is less stable than the statistical approaches. Although the Bayesian NN provides a better performance and is relatively more stable than the classical NN. Thus, we can confirm that the Bayesian approach for ANNs improves the stability and performance of the conventional NN.

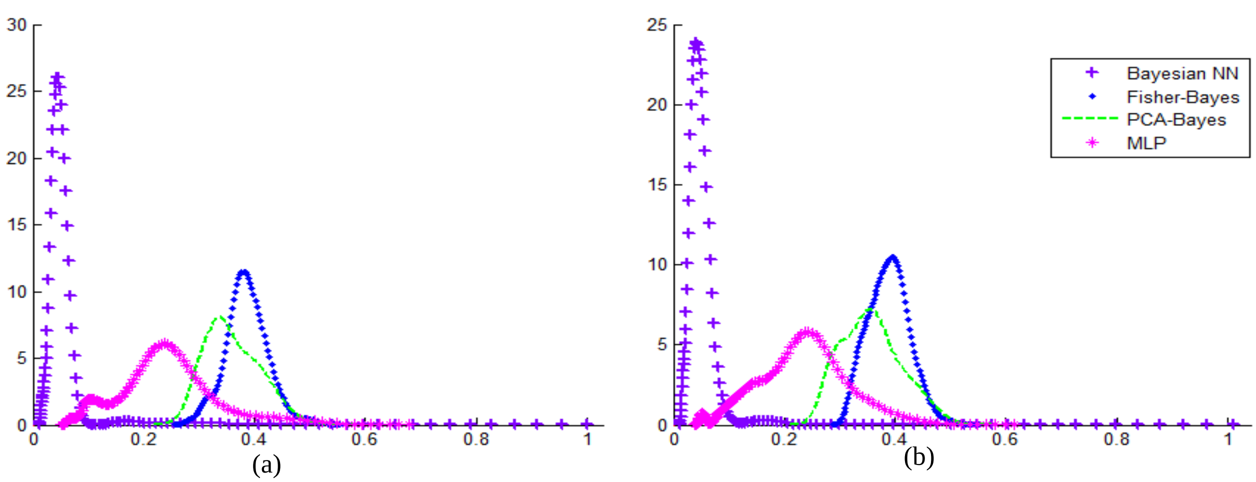

Figure 4: Error rate densities of PCA-Bayes (in green(--)), Fisher-Bayes (in blue(..)), MLP (in pink(*)) and Bayesian NN (in purple(+)) for Fourier descriptors (in the left) and Ghorbel descriptors (in the right).

Table 3: Comparison results of neural and statistical classifiers on the MNIST database.

	Fourier Descriptors		Ghorbel Descriptors	
	Mean	**Variance**	**Mean**	**Variance**
PCA-Bayes	0.3599	0.0022	0.3572	0.0026
Fisher-Bayes	0.3855	0.0014	0.3840	0.0012
MLP	0.2469	0.0072	0.2385	0.0057
Bayesian NN	0.0747	0.0037	0.0646	0.0011

VI. Conclusions

In this paper, a new criterion to comparing neural and Bayesian classifiers was proposed. In fact, a stability comparison based on estimating classifiers error rate probability density functions was presented.

The stochastic simulations demonstrated the superiority of the statistical approaches stability compared to the neural networks stability. In addition, the Bayesian approach for modeling NNs enhances their performance and stability. This study has provided a new conception to compare the stability results of the neural networks and other classifiers kinds. Another interesting point would be also to combine the classifiers to improve their stabilities.

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] MACKAY, David JC, et al. 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.