Stability Evaluation of Combined Neural Networks

Ibtissem Ben Othman and Faouzi Ghorbel
GRIFT Research Group, CRISTAL Laboratory, School of Computer Sciences, Manouba University, Manouba, Tunisia

Keywords: Bayesian Neural Networks, Error Rate Probability Density Function, Kernel-Diffeomorphism Semi-Bounded Plug-in Algorithm, Stability, Theoretical Error Rates.

Abstract: In the industrial field, the artificial neural network classifiers are currently used and they are generally integrated of technologic systems which need efficient classifier. However, the lack of control over its mathematical formulation explains the instability of its classification results. In order to improve the prediction accuracy, most of researchers refer to the classifiers combination approach. This paper tries to illustrate the capability of an example of combined neural networks to improve the stability criterion of the single neural classifier. The stability comparison is performed by the error rate probability densities function estimated by a new variant of the kernel-diffeomorphism semi-bounded Plug-in algorithm.

1 INTRODUCTION

In high dimension spaces, due to the samples limited size, the classification in these spaces requires dimension reduction in the first step. In order to simplify this reduction, the linear methods are the most commonly used ones.

Qualified by non parametric methods, the Artificial Neural Networks (ANNs or NNs) can achieve a non-linear dimension reduction which tries to find a subspace in which the data are well presented. Thus, the ANNs have become a regular method to provide a solution for the non-linear dimension reduction and classification, where statistical techniques have traditionally been used.

The traditional statistical approaches are based on the Bayesian decision rule, which presents the ideal classification technique. However, a dimension reduction is often required in the first step because of the limited samples size. The favorite linear technique for this purpose is the Fisher Linear Discriminate Analysis (LDA) which tries to find efficient discrimination directions.

A recent review of various experimental comparison studies between neural and statistical approaches is presented by Paliwal and Kumar in (Paliwal, 2009).

In order to compare the neural and statistical classifiers, most of researchers try to compare their prediction accuracy while forgetting the NNs instability criterion. In (Othman, 2013) and (Othman, 2014), we have proven the instability of network classifier results compared to the statistical methods. The stability 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 a new variant of the kernel-diffeomorphism semi-bounded Plug-in algorithm since pdf support information is known.

Many techniques proved their effectiveness in improving the classifiers performance. Combining several classifiers is becoming an active research area. Thus, the combination approach for the neural networks may improve their performance and stability.

The present work will be organized as follows: the next section summaries the neural classifiers including the multilayer perceptron neural network. Here we deal with the combined neural network approach. In section 3, we lead a comparative study between the combined and single neural networks stressing their stability degree. This 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 will present our works conclusions.

Ben Othman I. and Ghorbel F..
Stability Evaluation of Combined Neural Networks.
DOI: 10.5220/0005077402030209
In *Proceedings of the International Conference on Neural Computation Theory and Applications* (NCTA-2014), pages 203-209
ISBN: 978-989-758-054-3

2 ARTIFICIAL NEURAL NETWORKS

In pattern recognition, the extractor-classifier neural network is the most studied and used neural models. These mixed neural networks present a combination of the features extractors NNs and the classifiers NNs. Although the hidden layers are capable to reduce the data dimension in a non-linear way, the output layer makes the last decision by applying a non-linear separation to the extracted primitives. An interesting example is the feedforward Multi-Layer Perceptron (MLP) that uses the back-propagation algorithm.

The main duty of this supervised algorithm is to reduce the mean squared error (MSE) between the ANN outputs and the known target values:

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

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.

Based on the results from (Steven, 1991) and (Lepage, 2003), 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.

2.1 Neural Networks Limitations

Although the effectiveness and significant progress of ANNs in several applications, and especially the classification process, they present several limits. First, the neural classifiers produce a black box model in terms of only crisp outputs, and hence cannot be mathematically interpreted as in statistical approaches. Second, 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 threshold function to binaries the obtained outputs. Third, the NNs have a complex architecture that the task of designing the optimal model for such application is far from easy.

Unlike the simple linear classifiers which may underfit the data, the architecture complexity of NNs tends to overfit the data and causes the model instability. Breiman proved, in (Breiman, 1996), the instability of ANNs classification results. Therefore, a large variance in its prediction results can be introduced after small changes in the training sets. Thus, a good model should find the equilibrium between the under-fitting and the over-fitting processes.

Indeed, researches kept looking for suitable methods to solve these related problems. The cross validation method, mentioned in (Morgan, 1990) and (Weiss, 1991), presents the classical solution. German and al introduced, in (Geman, 1992), the *bias plus variance* decomposition of the prediction error, which presents an interesting solution for the over-fitting problem. Intending to reduce the over-fitting effect of NNs, a probabilistic interpretation of NNs learning methods has been proposed by Mackay, in (Mackay, 1992) and (Mackay, 1995), thereby using Bayesian techniques. In (Othman, 2014), we have proved that the Bayesian NN is most stable and performs better than the conventional NN. The performance and stability classification may also be improved by combining several neural classifiers (Miller, 1998), (Zhang, 2000), (Hansen, 1990) and (Morgan, 1990).

2.2 Combined Neural Networks

Many studies show that combining several classifiers significantly improves their performances with respect to each individual classifier. Several combined methods have proved their effectiveness to improve the individual classifier performance. Referring to the implementation order criterion, the classifiers combination approaches could be classified into sequential, parallel and hybrid. However, among these different combination architectures, the parallel architecture resulted in the most significant work. Its simplicity of implementation, its ability to exploit the combining classifiers taking into account (or not) the behavior of each classifier and its proven efficiency in many classification problems show its success, including in the sequential approach for which knowledge of the behavior of each classifier is necessary in order to obtain a pattern of effective cooperation.

To implement the combined NNs, the outputs of the first level networks were combined to a second level neural network. The combined NN model used in the present study is shown in Figure 1. Two MLPs, with the same structure, and having each one hidden layer, were used in the first level. Their outputs constitute the second level network inputs. A third MLP was used in the second level having also one hidden layer. In the first and second level, the back-propagation training algorithm was used. The number of hidden nodes in each level was determined considering the classification accuracies. In the hidden and output layers, the sigmoïd activation function was used.

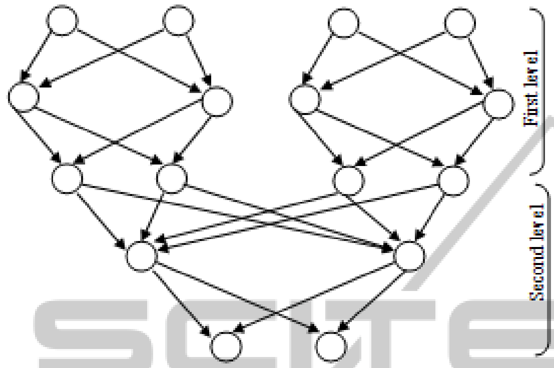


Figure 1: Combined neural networks topology.

3 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.

3.1 Non-parametric Density Estimation

The first step before comparing is to train the two classifiers, and then we proceed by measuring the error rate produced by each classifier with each one of *N* independent test sets. Let's consider $(X_i)_{1\leq i\leq N}$ the *N* generated error rates of a given classifier (Bayes or ANN). These error rates $(X_i)_{1\leq i\leq N}$ are random variables which have the same probability density function (pdf), $f_X(x)$. These $(X_i)_{1\leq i\leq N}$ are supposed 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 (Fukunaga, 1990) and (Ghorbel, 2012), 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:

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

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

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

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:

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

where;

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

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

3.2 Conventional Plug-in Algorithm

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 (Ghorbel, 2012). It applies directly a double derivation of the kernel estimator analytical expression in order to approximate the function *J(f)*.

3.3 Kernel-diffeomorphism Semi-bounded Plug-in Algorithm

The set of observed error rates $(X_i)_{1\leq i\leq N}$ 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 (Saoudi, 1997) and (Saoudi, 1994). 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 (Troudi, 2013) and (Ghorbel, 2012)

For complexity and convergence reasons, we propose in this paper a new variant of the kernel-diffeomorphism 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:

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

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

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

The use of this new variant of the kernel-diffeomorphism 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.

4 SIMULATIONS

The neural and statistical approaches were first compared experimentally on the multivariate Gaussian mixture classification problem. Three types of classifiers are applied to evaluate their performances and stability: Fisher-Bayes, single MLP and combined MLPs. For the combination approach, three MLPs were combined using the parallel topology discussed in the section 2.2. With the same train set (including 1000 samples for each class), we look to find the optimal transformation by the mean of the well known Fisher criterion which realizes the dimension reduction before applying the Bayesian rule, and then to fix the optimal NN model parameters for both single and combined MLPs.

After the training phase, we generate 100 independent supervised 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. In order to compare the stability degree, the error rate probability densities, retained for the statistical and neural approaches, are estimated using the new version of the kernel-diffeomorphism semi-bounded Plug-in algorithm discussed in the previous section.

Figure 2 shows the estimated error rate probability densities generated for the different classifiers on a mixture of two homoscedastic and heteroscedastic Gaussians. It illustrates the results of two homoscedastic Gaussians (Fig.2.a and Fig.2.b), a simple case of two heteroscedastic Gaussians (Fig.2.c), two heteroscedastic superposed Gaussians (Fig.2.d and Fig.2.e) and two truncated ones (Fig.2.f). The stability and performance of the classifiers are also analyzed by presenting their error rate means and variances in table 1.

By analyzing the results shown in the three first cases in figure 2 and table 1, the statistical classifier (Fisher-Bayes) admits the smallest error rate mean that proves its performance against the single neural classifiers for these simple cases. 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. For these complex cases, the Fisher LDA fails to find the optimal projection subspace. Whereas, the neural classifiers perform well due to their non linear reduction dimension capability. We deduce then the efficiency of these models. Although, the single MLP remains the least stable classifier that presents the greatest variance and thus the widest curve for the most cases. However, the combination approach for NN improves its stability and performance (except the fifth case where the variances are too close).

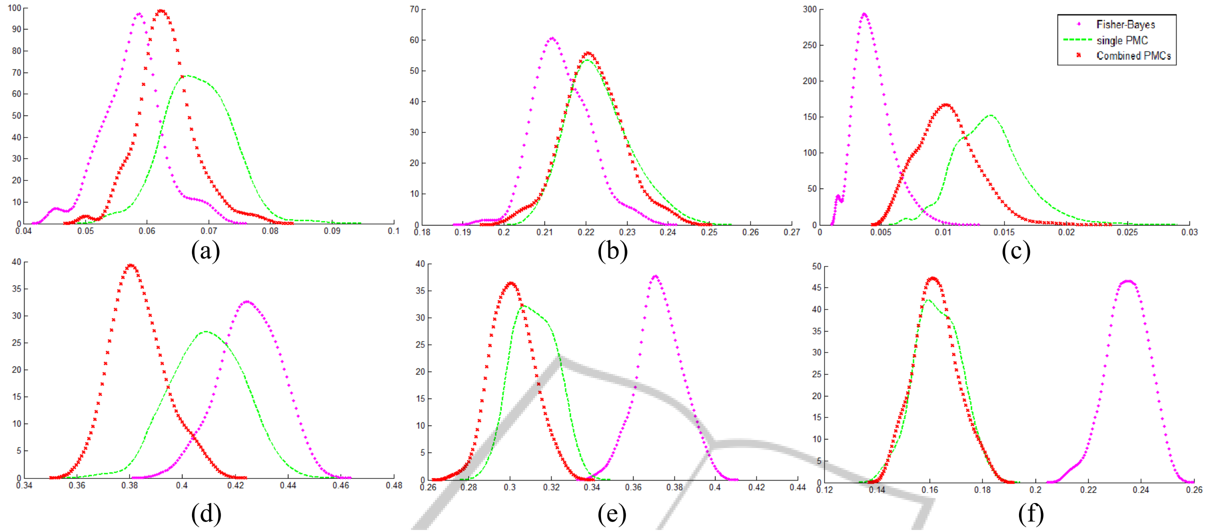


Figure 2: Error rate probability density function of Fisher-Bayes classifier (in pink(*)), single MLP (in green(--)) and combined MLPs (in red(x)).

Table 1: Comparison results of Fisher-Bayes, single MLP and combined MLPs.

Low (mean/variance) ==> Better (performance/stability)

Cases	**Distributions**		**Fisher-Bayes**		**Single MLP**		**Combined MLPs**	
	Gaussian 1	***Gaussian 2***	***Mean***	***Variance***	***Mean***	***Variance***	***Mean***	***Variance***
a	$\mu_1=(1,..,1),\sum_1=I_{10}$	$\mu_2=(2,..,2),\sum_2=I_{10}$	0.0577	0.0234	0.0682	0.0267	0.0630	0.0212
b	$\mu_1=(1.5,..,1.5),\sum_1=I_{10}$	$\mu_2=(2,..,2),\sum_2=I_{10}$	0.2142	0.0424	0.2231	0.0559	0.2217	0.0518
c	$\mu_1=(0,..,0),\sum_1=I_{10}$	$\mu_2=(2,..,2),\sum_2=2*I_{10}$	0.0044	0.0020	0.0138	0.0074	0.0104	0.0056
d	$\mu_1=(1,..,1),\sum_1=2*I_{10}$	$\mu_2=(1,..,1),\sum_2=3*I_{10}$	0.4253	0.1215	0.4093	0.1594	0.3834	0.1025
e	$\mu_1=(0,..,0),\sum_1=I_{10}$	$\mu_2=(0,..,0),\sum_2=2*I_{10}$	0.3730	0.0113	0.3119	0.0954	0.3015	0.0966
f	$\mu_1=(0,0,0)$ $\sum_1=$[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]	0.2347	0.0578	0.1630	0.0657	0.1623	0.0661

5 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 made of about 60.000 training samples and 10.000 test ones. The images resolution is 28x28 pixels.

For the training and test sets, we select randomly, from the MNIST training and test sets respectively, single digit images (the both sets contain 1000 images for the 10 digit classes). Random sampling images are shown in Fig.3.

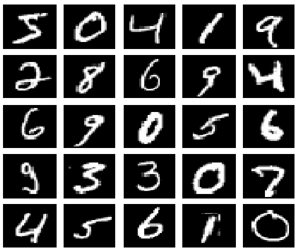


Figure 3: Random sample images of MNIST database. (wordpress).

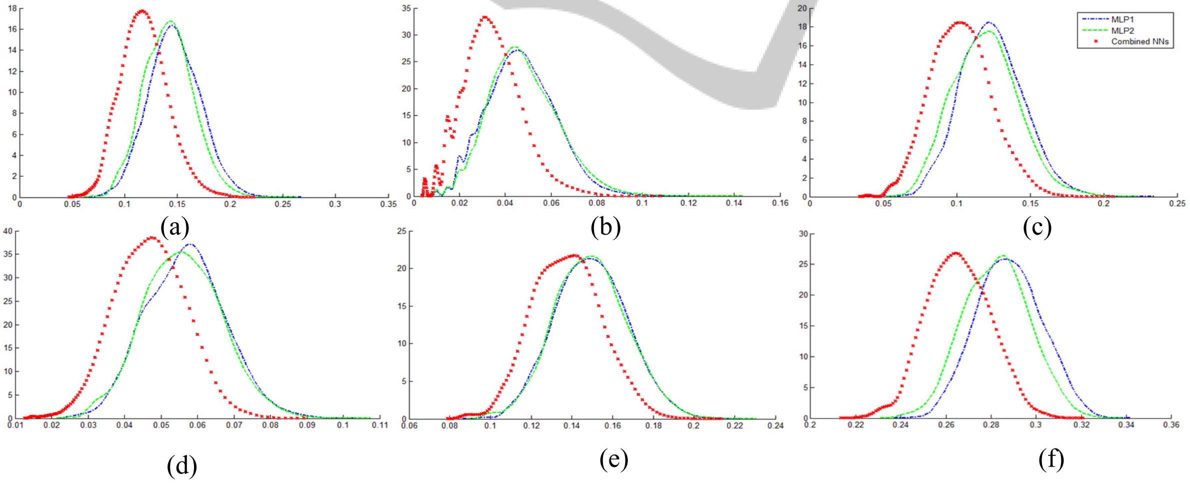
The most difficult step in handwritten digit recognition is to choose the suitable features. The chosen features must necessarily verify a non-exhaustive set of criteria such as stability, completeness, fast computation, powerful discrimination and invariance under the geometrical transformations. The invariant descriptors family proposed by Ghorbel in (Ghorbel, 1998) satisfies the various criteria cited above. Thus, each image will be described by this type of descriptor. We select a high descriptors size ($D = 14$).

The principal goal of the train set is to fix the parameters of the optimal NN model for both single and combined multilayer perceptron. Thus, we have used two single MLPs 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 this 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 kernel-diffeomorphism semi-bounded Plug-in algorithm for Ghorbel descriptors. In table 3, we summarize the MCR means and variances obtained for the two types of descriptors using the two single classifiers and the combined one. The results show the performance and stability of the combined MLP against the two single classifiers. Thus, we can approve that the stability and performance of the MLP increases with the parallel combination approach.

6 CONCLUSIONS

This paper provided a novel approach to comparing single and combined neural networks. This new criterion is performed by using a new variant of the kernel-diffeomorphism semi-bounded Plug-in algorithm.



(a) (b) (c)

(d) (e) (f)

Figure 4: Error rate probability density function of single MLP1 (in blue(-.)), single MLP2 (in green(--)) and combined MLPs (in red(x)) for Ghorbel descriptors.

Table 2: Comparison results of the single and combined MLPs on the MNIST database for Ghorbel descriptors

Low (mean/variance) ==> Better (performance/stability)

Digit classes	**MLP1**		**MLP2**		**Combined MLPs**	
	Mean	**Variance**	**Mean**	**Variance**	**Mean**	**Variance**
a (2 and 5)	0.1470	0.5779	0.1415	0.5197	0.1185	0.4741
b (4 and 7)	0.0470	0.2007	0.0480	0.2170	0.0345	0.1466
c (6 and 9)	0.1240	0.4437	0.1195	0.4740	0.1030	0.4005
d (0,1,2 and 3)	0.0567	0.1116	0.0560	0.1098	0.0470	0.0914
e (4,5,6 and 7)	0.1495	0.2996	0.1493	0.3095	0.1375	0.2758
f (0..9)	0.2880	0.1983	0.2824	0.2029	0.2655	0.1884

The comparative study demonstrated that the statistical classifier is more stable than the neural networks. However, the combination approach of NN showed improvements in its performance and stability.

Future works will be directed towards the stability evaluation of other classifiers such as support vector machine and CART decision trees. Another interesting point would be also to test other classifiers combination strategies.

REFERENCES

Breiman, L., 1996. Bagging predictors, *Machine learning*, vol. 24, no. 2, pp. 123-140.

Fukunaga, K., 1990. *Introduction to Statistical Pattern Recognition*, Academic Press, second edition.

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

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

Ghorbel, F., and al., 2012. *Récentes avancées en Reconnaissance de Formes Statistique*, Art-pi edition, Tunis, www.arts-pi.org.tn.

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

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

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

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

Mackay, D.J.C., 1995. *Bayesian methods for neural networks: theory and applications*.

Miller, D.W., 1998. *Fitting frequency distributions, Book Resource*. Second edition.

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

Othman, I.B., Ghorbel, F., 2013, *A New criterion for Comparing Neural Networks and Bayesian Classifier*, ICCAT' 2013, Tunisia.

Othman, I. B., Ghorbel, F., 2014. The Use of the Modified Semi-bounded Plug-in Algorithm to Compare Neural and Bayesian Classifiers Stability, Neural Networks and Fuzzy Systems, Venice, Italy.

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

Saoudi, S., Ghorbel, F., Hillion, A., 1994. 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.

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

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

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

Weiss, S.M., Kulilowski, C.A., 1991. Computer Systems that Learn. San Mateo, CA: Morgan Kaufmann.

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

www.wordpress.com.