Journal of Information Assurance and Security.
ISSN 1554-1010 Volume 12 (2017) pp. 001-017

An Adaptation Module with Dynamic Radial Basis Function Neural Network Using Significance Concept for Writer Adaptation

Lobna Haddad[1,5], **Tarek M. Hamdani**[2,5], **Wael Ouarda**[3,5] **Adel M. Alimi**[3,5] **and Ajith Abraham**[4,6]

[1]Higher Institute of Technological Studies of Rades (ISET), Tunisia
lobna.haddad@ieee.org

[2]Taibah University, College Of Science And arts at Al-Ula, al-Madinah al-Munawwarah, KSA
tarek.hamdani@ieee.org

[3]University of Sfax, National Engineering School of Sfax (ENIS) BP 1173, Sfax, 3038, Tunisia
wael.ouarda@ieee.org, adel.alimi@ieee.org

[4] Machine Intelligence Research Labs (MIR Labs), WA, USA

[5] REGIM-Lab: REsearch Groups in Intelligent Machines, Sfax, Tunisia

[6]IT4Innovations, VSB-Technical University of Ostrava, Czech Republic
ajith.abraham@ieee.org

Abstract: **In this paper, we focused on the handwriting-based biometric to personalize the hand held-devices, mainly used by one person, to recognize effectively its new writing style. For this end, we plug-in an adaptation module (AM) with a writer-independent recognition system (WIRS) to generate a writer-dependent recognition system. The WIRS response is then adapted considering the new writing style. The AM applied a sequential learning algorithm named GARBF-AM using a significance concept for writer adaptation based on a Radial Basis Function (RBF) neural network. The proposed GARBF-AM algorithm defines a new Growing and Adjustment algorithm named GARBF-AM. This algorithm can dynamically insert new hidden neurons under predefined conditions on the significance of both the new input and the nearest neuron. Otherwise, our algorithm adjusts the nearest and the desired contributor neurons parameters. For experiments, two writer dependent datasets are used. The first is LaViola dataset and the second is MEnv-REGIM that is created considering different physical positions of the writer (sitting, standing, walking, going up/down stairs and by car). The experimental results based on the two datasets show that the performance of the generic WIRS has improved significantly when integrating GARBF-AM algorithm. The comparative study highlights the benefits of the using the GARBF-AM against the well known OAM algorithm.**
Keywords: handwriting-based biometric, Incremental learning RBFNN, Writer adaptation, Information Security, Information assurance

I. Introduction

Biometrics emerged from its robustness to recognize and identify person using his physiological and behavioral characteristic. The physiological biometrics are fingerprints, iris scans, retina scans, hand geometry, and face recognition, ear recognition, DNA. The behavioral biometrics are dynamic keystroke, gait, signature, handwriting.Nowadays, various applications use efficiently biometric characteristics. The use of these methods handles the problem of the lost of some passwords, keys which can be also stolen. For this purpose, researchers used biometric information due to its availability when the person is present. Systems based on physiological biometrics require a specific technology that limits their use by common person. The choice between all these modalities depends essentially on traits that will be acquired from sensors application. Consequently, especially marked applications for hand held devices, the behavioral biometrics seems the more appropriate. Our application consists of the personalization of the hand held-devices, mainly used by one person, to recognize effectively its writing. Referring to [39], three main categories of handwriting-based biometric approaches can be identified: forensic verification, user authentication and handwriting recognition. For this we are focusing on the handwriting recognition approach to reach our goal.

The hand-held devices have made our life easier and tidier, making them a need in our everyday routines. Such devices as the PDAs, Smartphones, tablets have made the human machine interaction convivial and rapid. Since the natural method of human communication is based on writing or speaking, we have to go beyond the use of the common keyboard to interact with these devices. Consequently, the setting of a writer-independent recognition system turns out to be essential but not sufficient. These new devices have become so close to the user they that incite the researchers not only to propose a recognition system that is learned on a writer-independent database to ensure generalization but also try to upgrade the system in such a way that it adapts itself during the use of the device to a specific writer style has become a need.

Handwriting recognition systems can be divided into two categories which are Writer-Independent Recognition Systems (WIRS) and Writer-Dependent Recognition Systems (WDRS). A WIRS is trained with a dataset collected from a wide number of writers to include as many writing styles as possible to ensure that the recognition system will perform well. Conversely, the aim of the WDRS is not to consider all the possible writing styles but to obtain a higher recognition rate for an individual new writing style. In this paper we focused on the development of a WDRS starting from a WIRS and carrying out a new writer adaptation approach that its application is not limited to the used recognizer in our experiments but can be applied to any writer-independent recognition system.

There has been a huge amount of research in the field of writer adaptation during the last two decades. All the achieved works propose systems that depend on either the type of recognition system or the nature of adaptation process. Pattern based systems reorganize the database prototypes defining a data management process (addition, modification and deletion) to improve the response quality. Its worthy to know that this first group of systems includes the prototype based systems that can be adapted to a new writing style by reorganizing the standard prototype set or also using a new writer-dependent prototype set. In [17], the authors propose an adaptation method which uses not only misclassified patterns but also correctly-classified patterns as learning samples. The system in [16] learns new writing styles, by activating new prototypes and inactivating erroneous ones to automatically transform a writer-independent database into a writer-dependent database of very high quality and compactness. In [18], the used classifier is based on a Fuzzy Inference System which automatically fits the handwriting style of the writer who uses the system.

Some other parametric systems update the internal parameters of the classifier to permanently change the classifier structure. This second group of systems includes mainly the recognition systems that adapt their response by modifying their parameters values. For the SVM adaptation, [31] used a biased regularization for personalization as a principle way of trading off user-dependent versus user-independent information. Since the proposed method is a modification of standard SVMs, [20] achieved an adaptation by re-learning the different SVMs using virtual examples. The system in [21] applies an SVM based multiple kernel learning where support vectors are adapted to better model the decision boundary of a specific writer. For the Hidden Markov Model (HMM) adaptation, three techniques were generally used namely the expectation maximization (EM) retraining, the maximum a posteriori (MAP) adaptation and the maximum likelihood linear regression (MLLR) technique [30, 22, 32, 33, 19]. We also find a writer adaption method based on an incremental linear discriminant analysis (ILDA) [11] where the writer adaptation is performed by updating the LDA transformation matrix and the classifier prototypes in the discriminative feature space. using, the maximum a posteriori (MAP) adaptation and the maximum likelihood linear regression (MLLR) technique. The two subsequently described system groups are based on a permanent change of the system behaviour. This choice is based on the fact that devices are personal and mono-user. This fact limits the system reliability for more than one user on the same system, or for more than one session for the same user. Helpfully, such systems give the user the possibility to reset their recognition system to start a new adaptation session but do not generally give the possibility to save specific adaptation contexts or to manage multiple sessions.

Few further systems take up an independent adaptation module to adjust their response without modifying their classifiers internal parameters. This type of group includes systems adapting themselves without modifying the writer-independent system parameters values. Some systems used an adaptation module based on Radial Basis Function Neural Network (RBF-NN) with a sequential learning algorithm [4, 12, 13, 14]. The adaptation module is placed on the top of a recognition system. Moreover, [10] used a Style Transfer Mapping where the data of different writers are projected onto a style-free space, and the writer-independent classifier needs no change to classify the transformed data and can achieve a significantly higher accuracy. The system in [23] developed a representative symbol recognizer that uses a set of binary classifiers based on AdaBoost as part of an allpairs recognition algorithm. AdaBoost takes a series of weak or base classifiers and calls them repeatedly in a series of rounds on the training data to generate a sequence of weak hypotheses. Each weak hypothesis has an associated weight that is updated after each round, based on its performance on the training set.

The aim of this paper was to define a distinct adaptation module separately from the initial handwriting recognition system. This is very useful to ensure the portability of the adaptation system, in such a way that it can be added to any existing system and after that adapts its response to the new context. A context oriented approach can also be useful to generalize both multi-session and multi-user options and to give more portability and reusability to the adaptation system.

This paper presents the development of a new sequential learning algorithm using the neuron significance concept for a writer adaptation that uses an adaptation module based on Radial Basis Function (RBF) neural network. The neuron significance is defined as the contribution made by that neuron to the network output averaged over all the input data received so far. The proposed adaptation module defines a new Growing and Adjustment algorithm named GARBF-AM. First, we estimate the significance of the new added neuron which is the contribution made by that neuron to the network output averaged over a certain number of already received observations. Second, we used a new formula to es-

timate the nearest neuron significance which is based on the contribution of the nearest neuron to the output layer for the current input. The nearest neuron significance is always used for neuron pruning [6, 7, 8] but will be used, in our algorithm, as a criterion for network growth. Moreover, compared to our previous work, we added a new step for the network growth where we resize the width of the nearest unit to minimize the overlap between the new added unit and the nearest unit. This algorithm can dynamically insert new hidden neurons under predefined conditions on the significance of both the new input and the nearest neuron. Otherwise, our algorithm adjusts the nearest and the desired contributor neurons parameters. The module adaptation is associated to writer-independent recognition systems (WIRS) to generate a configurable writer-dependent recognition system (WDRS). The original WIRS response is then adapted considering the new writing style.
The remaining of the paper is organized as follows; in Section 2, we depicted different methods for neural network online learning methods. The third Section was devoted to the description of our adaptation system and the GARBF-AM algorithm. In the fourth Section the experimental results that validate the consistency and the performance amelioration of a writer recognition system were reported. The last section presented some concluding remarks and suggested some interesting perspectives.

II. Related work

The first suggested incremental learning algorithm of RBF-NN was that of Platt [1] named Resource Allocating Network (RAN). The RAN algorithm allows a sequential learning of the RBF-NN that initially contains no hidden units, and can add hidden units in the RBF-NN to extend the approximation ability when errors classification are reported. In fact, this algorithm is made up of two actions depending on how the network performs on a presented pattern. If the network performs poorly, a new unit is allocated satisfying some growth criteria. If the network performs well, the existing network parameters are updated using standard Least mean squares (LMS) gradient descent algorithm. Subsequently, an enhancement of the RAN in which the extended Kalman filter (EKF) algorithm is used instead of the LMS algorithm [2]. The MRAN algorithm [5] combines the growth criterion of the RAN with a pruning strategy based on the relative contribution of each hidden unit to the overall network output. The resulting network leads toward a minimal topology for the RBFNN. The MRAN has recently been used in [24] combined with a growing gaussian mixture model (GGMM) for classification problems. Besides in [27], the algorithm was used to avoid the catastrophic interference in incremental learning between Resource Allocating Network and the Long Term Memory (RAN-LTM). In RAN-LTM, not only a new training sample but also some memory items stored in Long-Term Memory are trained based on a gradient descent algorithm. On the other hand, [3] presents a new algorithm which uses accumulated error information to determine where to insert new units. The diameter of the localized units is chosen relying on the mutual distances of the units. In [6], the generalized growing and pruning (GGAP) training algorithm for RBF-NN is applied. GGAP is a RAN algorithm but introducing a formula for computing the significance of the network units. So, the growing and pruning strategy is based on linking the required learning accuracy with the significance of nearest or new units. [7, 8] present improved GAP-RBF for enhancing its performance in both accuracy and speed and the resulting algorithm is referred to as Fast GAP-RBF. Then, the significance of the network units is estimated by the recently received M training samples. [28] used the idea to exploit a memory that corresponds to representative input-output pairs. These pairs are selected from the training data, and they are learned with newly given training data to avoid forgetting. The sequential learning of RBF-NN presented in [29] is applied for the parameterization of freeform surfaces from larger, noisy and unoriented point clouds. The algorithm allows neurons to be dynamically inserted and fully adjusted (e.g. their locations, widths and weights), according to mapping residuals and data point novelty associated to the underlying geometry. Pseudo-neurons, exhibiting very limited contributions, can be removed through a pruning procedure. Additionally, a neighborhood extended Kalman filter (NEKF) was developed to significantly accelerate parameterization. Added to that, the system in [9] used a self-adaptive error based control parameters to alter the training data sequence, evolve the network architecture, and learn the network parameters. In addition, the algorithm removes the training samples which are similar to the stored knowledge in the network.

III. Writer adaptation through sequential learning RBF-NN

In order to achieve a writer adaptation that can be applied to any system independently of the implemented classifiers type, we opted to use a module to adapt the on-line handwriting recognition system (OHRS). The Adaptation Module (AM) is based on the Radial Basis Function Neural Network (RBF-NN) because it is considered as the most convenient network in sequential learning by reason of its simplicity, local learning, robustness, optimal approximation [34]. The architecture of the on-line handwriting recognition system with writer adaptation (WDRS), presented in Fig. 1, is made up of the Writer-independent Recognition System (WIRS) and the Adaptation Module (AM). The AM is added below the WIRS, and its role is to examine the writer-independent output (WI_output) and produce a more correct output vector close to the desired response of the user. In this way, the AM adds to WI_output of the recognition system an adaptation vector (A) to produce a writer-dependent output (WD_output) using Eq.(1).

$$WD\_output = WI\_output + A \quad (1)$$

Where A is the adaptation module output. $A = (A_1, ..., A_L) \in \mathbb{R}^L$.

The AM, presented in Fig. 2, is an RBF-NN with N hidden neurons. The output (A) of the adaptation module for an observation (I, D) is calculated using Eq.(2, 3).

$$A = \sum_{n=1}^{N} W_n \phi_n(I) \quad (2)$$

$$\phi_n(I) = exp\left(-\frac{\|I - C_n\|^2}{\sigma_n^2}\right) \quad (3)$$

The used notations in the equations are presented below:
I: Input observation which is the output of the WIRS, $I = (I_1, ..., I_L) \in \mathbb{R}^L$ where L is the dimension of an input observation,
N: Number of neurons in a hidden layer, $(n = 1, ..., N)$,
σ_n: Width of the nth hidden neuron,
ϕ_n: The response of the nth hidden neuron of an input vector I,
C_n: Center of the nth hidden neuron, $Cn = (C_{n,1}, ..., C_{n,L}) \in \mathbb{R}^L$,
W_n: Weight connecting the nth hidden neuron to the ouput layer,
D: Desired output corresponding to the input I. In our experiments the target vector (D) is 1 for the neuron corresponding to the correct class and 0 otherwise.

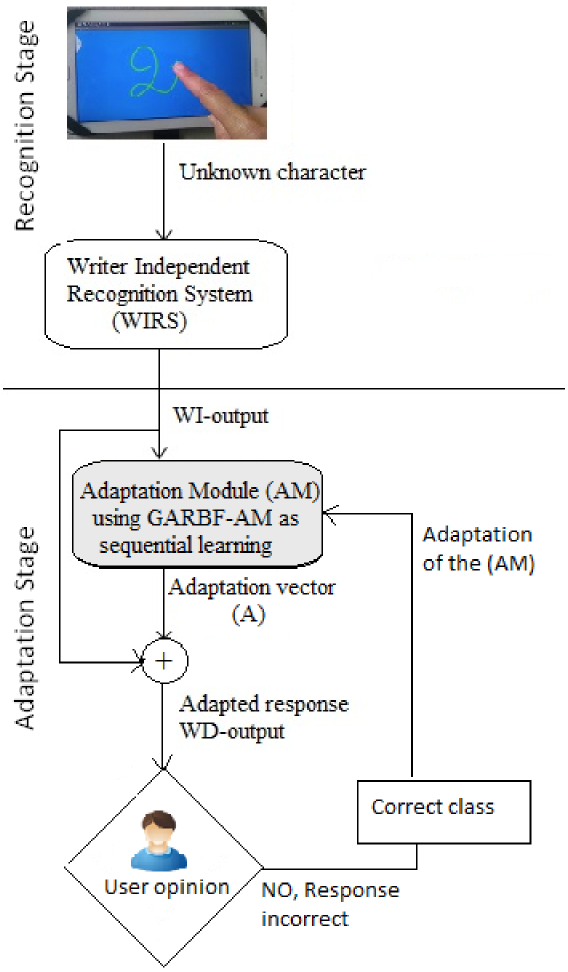


Figure. 1: Architecture of Adaptive recognition System

In this section, we present the adaptive sequential learning GARBF-AM algorithm (section III-A), definition and estimation of the neurons's significance (section III-B), network growth strategy (section III-C) and network adjustment strategy (section III-D).

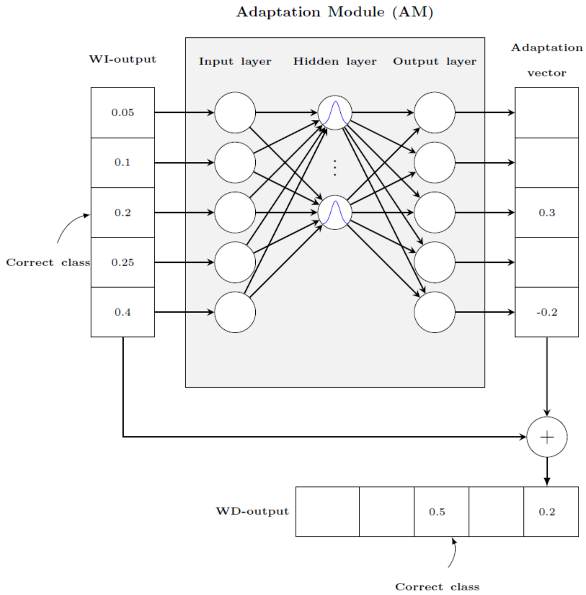


Figure. 2: Architecture of the adaptation module (AM)

A. The Adaptive Sequential Learning of the Adaptation Module (AM)

At the beginning the Adaptation Module (AM), presented in Fig. 2, contains no hidden neurons. After each misclassification, we applied an incremental learning algorithm, named GARBF-AM later on, so that the AM learns to correct the mistakes caused by the WIRS. The GARBF-AM is a supervised and incremental algorithm, working in two phases: the growing and the adjustment. The adaptation steps are summarized in algorithm 1.

Algorithm 1 GARBF-AM Algorithm: Adaptive Sequential Learning

For each observation (I, D)
Compute the overall writer-dependent recognition system output using Eq.(1)
Calculate the Significance of the new input and the nearest neuron using Eq.(4, 5) (section III-B)
Apply the criteria for adding or adjusting neurons
if cr1 and (cr2 or cr3) **then**
    Call Growing Algorithm (section III-C)
else
    Call Adjustment Algorithm (section III-D)
end if

In algorithm 1, we point out the two adaptation strategies using $cr1$, $cr2$ and $cr3$ as criteria (detailed in section III-C) to decide the novelty of the new input. As the adapted system learns to correct the errors made by the WIRS, the GARBF-AM algorithm allocates new neurons only for novel errors that the WDRS has not seen before. Otherwise, the algorithm adjusts parameters of hidden neurons. Consequently, the number of hidden neurons grows sub-linearly with the number of errors.

B. Definition and Estimation of Neurons's Significance

The salient properties of a writer adaptation are its speed and efficiency, using a few examples of a specific writer, without WIRS performance degradation [4]. These properties can be satisfied by leading to an optimal positioning of the basis functions (hidden neurons number and centers location) by growing its architecture incrementally [36]. In our work, we are focused on neuron's significance concept applied to incremental learning algorithms for RBF networks.
The neurons significance was proposed by [6] and used and improved by these later works [7, 8]. According to [6] (GGAP-RBF algorithm), the significance gives a measure of the information content in the neuron about the function to be learned and is defined as the contribution made by that neuron to the network output averaged over all the input data received so far. This definition is applied successfully in the fields of function approximation and classification problems. To reduce the computational complexity, [7] (FGAP-RBF algorithm) proposed a simplified formula to estimate the significance of neurons. In this case, the neuron significance can be estimated using a certain limited number of recently received training samples. The neuron significance was usually used for classification problems.
Our algorithm GARBF-AM (algorithm 1) is developed aiming at overcoming the writer adaptation problem by using a simple significance estimation of two neurons (the added new neuron and the nearest neuron). These two significance values will be used separately on two novelty conditions ($cr2$ and $cr3$) to optimize the growing cases. Bearing in mind the observation (I, D), we find the nearest unit to it from the existing units in the RBF-NN. After that, we estimate the significance of the intentionally added new unit and the significance of the nearest unit. These two significant units are detailled below.

i) The significance of an intentionally added new unit

GARBF-AM learning algorithm allocates new hidden neurons and adjust the nearest neuron parameters. An observation should give rise to a new hidden neuron if it is novel. We use the significance of the new added neuron ($Esig_{novelty}(N+1)$), which is calculated using the Eq.(4), to make the decision about the novelty of a new observation (I, D).

$$Esig_{novelty}(N+1) = \frac{\|er\|}{M} \sum_{s=B}^{K} exp\left(-\frac{\|I_s - I\|^2}{\kappa^2 \|I_s - C_{nearest}\|^2}\right) \quad (4)$$

Where: K: is the total number of inputs already seen,
I: is the current received input ,
I_s: is an input from the recently received inputs that should be remerbred, $C_{nearest}$: is the center of the hidden neuron nearest to the current input I,
M: is the number of recently received inputs and has to be remembered, $B = K - M + 1$,
$er = D - WD\_output$: is the error produced by I,
κ: is an overlap factor that determines the overlap of the responses of the hidden neurons in the input space.
From a statistical viewpoint, the significance of the added neuron is the average information content of a neuron over all inputs seen so far, and also the contribution of that neuron to the overall performance of the RBF network [6]. To reduce the complexity load of the learning algorithm and avoid the difficulty of knowing the input distribution, the significance of the new neuron is estimated using only a memory containing the M recently received inputs [7]. If the current input is far from most of the recently received inputs Ms, the significance of the new neuron will be high enough to be allocated to the neural network. The value of M and its impact on the adapted system performance is discussed in section IV-A.2.

ii) The significance of the nearest unit

The significance of the nearest neuron is its contribution to the output layer. To this end, we compute the product of the norm of the weights vector by the output of the nearest neuron. In our work, the significance of the nearest neuron will be used in the network growth. We provide a new formula Eq.(5) to calculate this significance using only the current input. For the current observation, if the significance of the nearest neuron is not high enough, that it means that it is insignificant for the current observation. This case should give rise to a new hidden neuron.

$$Esig_{nr}(C_{nearest}) = \|W_{nearest}\| \times \phi_{nearest} \quad (5)$$

Where $W_{nearest}$ is the weights between output layer and nearest unit, $\phi_{nearest}$ is the output of the nearest unit.
Contrastly, in the already achieved research [6, 7, 8] calculate the significance of the nearest neuron after adjusting its parameters. If the nearest neuron becomes insignificant it should be removed from the network. In the same way as the significance of the new added neuron, the nerarest neuron significance is estimated using the M recently received inputs.
In fact, the removal of a hidden neuron is relevant in classification problems but doesn't have the same impact on a writer adaptation problem. In a writer adaptation context, the network learns a new writing style with a small number of samples and should remember an already seen example. Removing a hidden neuron will deeply increase the forgetting and the error classification.

C. Network Growth

Basically, the RBF-NN begins with no hidden neurons. The training inputs are sequentially exposed to the system and the user has to report the misclassification and specifiy the correct class. If it's the first time an error is mentioned, a new RBF unit is automatically allocated. Otherwise,we study the novelty of the current input by estimating its significance using Eq.(4). Then, we estimate the significance of the nearest unit compared to the current input applying Eq.(5). To perform a writer adaptation with a small number of RBF units, we used the following growing criteria ($cr1$, $cr2$ and $cr3$) :

$$\begin{cases} \|I - C_{nearest}\| > d_{min} & cr1 \\ (Esig_{novelty}(N+1) > e_{1min}) & cr2 \\ (Esig_{nr}(C_{nearest}) < e_{2min}) & cr3 \end{cases}$$

where d_{min} is a threshold corresponding to the minimal distance, $C_{nearest}$ is the center of the nearest unit to the input I and e_{1min} and e_{2min} are the desired approximation accuracy.

- The criterion $cr1$: It is the basic criterion that is always used for a growth network [1, 4, 6, 7, 8, 24]. It allows to test if the current input is far from the existing units. This criterion guarantees a well balanced neurons distribution because neurons are inserted ensuring at least a threshold minimal distance from each other.
- The criterion $cr2$: It verifies if the current input is written with a novel style since $Esig_{novelty}(N+1)$ is greater than the constrained approximation accuracy e_{1min}.
- The criterion $cr3$: It checks if the nearest unit is insignificant to the current input when $Esig_{nr}(C_{nearest})$ is less than an approximation accuracy e_{2min}.

Algorithm 2 GARBF-AM Growing Case Algorithm

Allocate a new hidden unit (N+1) with:

1. The input becomes the center of the new unit.
$$C_{N+1} = I \tag{6}$$
2. The weight values of connections between the new unit and the output layer correspond to the desired output.
$$W_{N+1} = D_{N+1} \tag{7}$$
3. To avoid the overlap of different regions of the RBF units, the width of the new unit is fixed to the distance between the input and the unit which is the nearest to it.
$$\sigma_{N+1} = d_{(I,C_{nearest})} \tag{8}$$
4. Resize the width of the nearest unit to minimize the overlap between the new added unit and the nearest unit.
$$\sigma_{nearest} = min(\sigma_{nearest}, d_{(I,C_{nearest})}) \tag{9}$$

Therefore, in the case of satisfactory growing criteria : $cr1$ and ($cr2$ or $cr3$), a new hidden unit will be allocated relying on the steps described in algorithm 2.

The objective of our system is the learning of a new writing style of only one user. The user can write the same character intermittently and the writer-independent recognition system responds similarly each time, even if it is an error classification. The adaptive system should remember the same errors to correct it without forgetting the already learnt. This compromise is workable by using the memory to store the M last characters and estimating the novelty of the current one. By focusing on the criteria of GARBF-AM algorithm, $cr1$ ensures a well separated and scattered neuron distribution to ovoid the adding of superfluous neurons. $cr2$ enforces that a neuron is generated only if the error made by the adapted system is novel. The combination of these two criteria strictly limits the network growth ensuring a minimal number of allocated hidden neurons (proved in section IV-A.3, IV-B.2). Consequently, having a little number of neurons straightly affects the recognition rate of the adaptive system. To have the optimal number of hidden neurons that minimize the recognition error rate, we used $cr3$ which considers the nearest hidden neuron significance. The significance of the nearest neuron is its contribution to the adaptive system response. Then, more this significance is unimportant means that the neuron is insignificant to the current input. The impact of each of these three criteria is discussed in section IV.

D. Network Adjustment

When an observation is presented and the growing criteria are not satisfied, the network adjustment needs to check the desired contributor (Dc) neuron in addition to the nearest unit. The Dc is the neuron that contributes relatively much to the erroneous writer-dependent output. So, to find the Dc unit we used Eq.10 where o is the desired maximum output position.
$$Dc = Max_j(\phi_j \times W_{jo}) \tag{10}$$
Thus, we update only the parameters (center and weights) of either the nearest neuron or the two neurons: $C_{nearest}$ and Dc. These two cases are distinguished according to the distance value, $d_{(C_{nearest},Dc)}$, between both the nearest and the desired contributor units. Basically, only the nearest unit is adjusted, but if $d_{(C_{nearest},Dc)}$ is lower than the threshold minimal distance d_{min} then the Dc unit is also updated. The adjustment case is described in the algorithm 3.

Algorithm 3 GARBF-AM Adjustment Case Algorithm

if Growing criteria are not satisfied **then**
  Adjust parameters of the nearest unit using Eq.(12, 13)
  if $d_{(C_{nearest},Dc)} < d_{min}$ **then**
    Adjust parameters of the Desired Contributor unit using Eq.(12, 13)
  end if
end if

The research achieved in the field of sequential learning of RBF-NN generally used either the standard least mean square (LMS) gradient descent or the Extended Kalman Filter (EKF) algorithm. Therefore, having an adaptation time and memory size constraints, we opted for the standard LMS gradient descent to minimize the error each time no new unit is allocated. The error, made by the adapted system and should be corrected, is calculed as follows:
$$Er = \|D - WD\_output\|^2 . \tag{11}$$
Where D is the desired output corresponding to the current input I, WD_output is the writer-dependent output.

Whenever a new unit is not allocated, we adjusted the center position of a hidden neuron using Eq.(12).
$$\Delta C_n = 2\frac{\alpha}{\sigma_n}(I - C_n)\phi_n[(D - WD\_output) \times W_n] \tag{12}$$
In addition, we adjusted the weights of a hidden neuron using Eq.(13).
$$\Delta W_n = \alpha(D - WD\_output)\phi_n \tag{13}$$
Where C_n, σ_n, W_n and ϕ_n of a neuron n respectively indicate center position, width, weights to the output layer. WD_output the reponse of the adapted system corresponding to the input I. α is a learning rate.

IV. Experimental Results

To test the performance of our Writer Adaptation strategy applying the (GARBF-AM) algorithm as sequential learning of the adaptation module, it was connected to the output of a writer-independent recognition system for alphanumerical characters. This system is developed using a generic toolkit (LipiTk) whose aim is to facilitate the development of online handwriting recognition engines [15] [1]. The IRONOFF handwriting database [35] was used to train the recognizer.
In principle, the number of inputs in the adaptation module must be the same as the number of outputs response of the writer-independant recognizer. However, the two datasets used for the experiment include 36 class characters (digits and lower case letters). Thus, the adaptation module consists of 36 inputs and outputs.
In this section, we present the results that demonstrate the performance of our writer-dependent recognition system using a Benchmarking dataset (section IV-A) and a Multi-Environment dataset (section IV-B).

A. *Writer adaptation using a Benchmarking dataset*

1) *LaViola dataset description*

To test the efficiency of our writer adaptation system, we used a benchmarking dataset named LaViola. The LaViola dataset [23] contains samples of handwritten digits (0-9),characters (a-z) and mathematical symbols written by 11 people taken with an (HP) Compaq tc1100 Tablet PC. This dataset involves two sets for training. Each training dataset contains few training samples (10 per class and per writer). In this case we have 720 examples per writer. The results on this dataset have been reported in [23, 25]. The average recognition rate without adaptation using the writer-independent alphanumerical recognition system is 80%.

2) *Analysis of the algorithm parameters*

To get the upper performance of our algorithm GARBF-AM, we have to choose the right values of its parameters. The algorithm works with four weighty parameters which are: the memory size M, the threshold d_{min} and the desired approximation accuracy e_{1min} and e_{2min}. The values of d_{min}, e_{1min} and e_{2min} are important for the topology updating procedure. The common parameters of GARBF-AM are fixed for the two writer-dependent datasets as: the threshold d_{min}=0.2, the learning rate α=0.02 and approximation accuracy e_{1min}=0.2 and e_{2min}=0.25, κ=0.8. Also we used the Euclidian distance to caculate the distance between unit centers and inputs. Moreover, to show the effectiveness of the GARBF-AM we have resorted to the cumulative classification errors made during the real interactive use of the tactile apparatus. In this section, we analized the impact of the memory factor M on the adapted system performance using LaViola dataset as shown in Fig. 3. This analysis helps us to determine the best value of M that optimizes the adaptation performance and the network size. We set the parameter M from 2 to 40, and the average neurons allocated and the average cumulative errors reached by all the writers (w1...w11) dataset is shown in Fig. 3. Clearly, it is observed from Fig. 3 that increasing M can significantly reduce the average number of hidden neurons for the writer-dependent dataset; meanwhile, the average cumulative errors is increased in accordance with the increase of the parameter M.

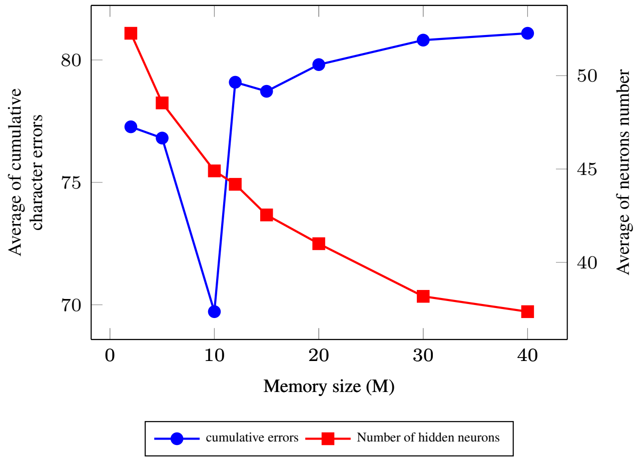


Figure. 3: The effect of memory size on adaptation performance using the average of cumulative errors and neurons number obtained from all writers of LaViola dataset

These results prove that when increasing the memory size, a new input has less opportunities to be novel compared to the M examples. In these cases the growing criterion (cr2) will not be satisfied which leads to the decrease the number of the allocated hidden neurons. Having a reduced number of the resulting hidden neurons slows down learning which increases the cumulative errors. From this experiment, we deduce that when M is 10 we can reach the lowest error rate.

3) *Results*

In this section we present the quantitative results using both of the GARBF-AM and OAM [4] algorithms. Compared to OAM, GARBF-AM resizes the width of the nearest neuron when a new hidden neuron is allocated and updates the parameters of the desired contributor neuron in an adjustment case. For evaluation,we looked at the computation load taken by these two algorithms to process each set of data. It's obvious that the time taken to process a new input varies according to the network size. Because the reponse time is crucial in adaptation for handwriting recognition, we displayed not only the recognition rate, but also the total number of hidden neurons allocated in the adaptation process using both the OAM and the GARBF-AM algorithms. Moreover, we showed the results that demonstrate the performance of the proposed sequential algorithm GARBF-AM taking into account its growing criteria. We should remind that our algorithm is based on three criteria which are $cr1$, $cr2$ and $cr3$ (Section III-C). To study the impact of the nearest significance criterion ($cr3$) on adaptation performance, we defined two variants of our algorithm. The first was defined by several systems [6, 7, 8] and uses only two criteria ($cr1$ and $cr2$). We call it here Restricted GARBF-AM. The second is GARBF-AM that uses the following combination of the three criteria ($cr1$ and ($cr2$ or $cr3$)). The results of these two variants are compared to those obtained using the OAM

[1] available at http://lipitk.sourceforge.net

algorithm which uses only $cr1$ as growing criterion.
In this section we present the results of three studies. We focus, in the first study, on the performance evaluation of the GARBF-AM by referring to the recognition rate and the error rate reduction using each of the three algorithms (GARBF-AM, OAM and Restricted GARBF-AM). Moreover, we compute precision and recall to analyze the overall performance of the adapted system. The second study consists of the statistical comparison of accurate adaptation of the GARBF-AM over the other two algorithms: OAM and Restricted GARBF-AM. Finally, the last study presents the adaptation efficiency of the proposed GARBF-AM.

- *Performance evaluation of GARBF-AM using accuracy:*

To study the performance of our algorithm, we report, in Table 1, the recognition rate and the number of hidden neurons allocated in the adaptation module using OAM, Restricted GARBF-AM and GARBF-AM. From the results shown in Table 1, we observe that the Restricted GARBF-AM decreases significantly the number of hidden neurons, but this was achieved at the expense of decreasing the recognition rate for the majority of writers. Furthermore, the GARBF-AM outperforms the system by increasing slightly the number of hidden units (average 43 neurons) compared to Restricted GARBF-AM (average 37 neurons). Taking writer w3 as an example, the recognition rate without adaptation is 80.83% and is increased using OAM, Restricted GARBF-AM and GARBF-AM by 9.62%, 12.37% and 14.78% respectively. On the other hand, OAM allocates a high number of hidden neurons (Nb neur=52) which decreased when we applied Restricted GARBF-AM and GARBF-AM by 44.23% and 32.69% respectively. This example shows that GARBF-AM allocates an optimal number of hidden neurons to reach a best recognition rate. Moreover, compared to the writer-independent recognition system, the GARBF-AM achieves the highest percentage recognition rate improvement for writer w1 (21.93%) and the lowest percentage improvement for writer w11 (6.88%). We consider this improvement a result of writer's handwriting style that is well represented with the different hidden neurons allocated in the adaptation module.
In Fig. 4 (a) we plot, for writer w10, the baseline cumulative error without adaptation and the cumulative character errors from the time when the adaptation started to give an estimated instantaneous error rate.The goal is to compare the strategy of adding hidden neurons of each used method to understand its impact on the cumlative errors vs. the number of used hidden units. OAM, using only $cr1$ as a growing criterion, generates a high number of neurons in the adaptation module. Restricted GARBF-AM, using a growing criteria which is true in such cases, more adjustment of the existing neurons will be achieved than adding new ones. In this case the slope of cumulative hidden neuron decreases considerably but generates the increase of the slope of cumulative error. By introducing the $cr3$, the GARBF-AM growing condition is broader and the algorithm adds just the necessary number of neurons to be more efficient.
Table 2 reports the error rate reduction using OAM and GARBF-AM algorithms. It is clearly observed that our writer adaptation algorithm GARBF-AM reduces the error rate reduction compared to the OAM algorithm. Using GARBF-AM learning algorithm and taking writer w3 as an example, the recognition rate is improved from 80.83% to 92.78% indicating a very high error reduction of 62.34%. Similarly, for writer w11 the recognition rate is improved from 86.80% to 92.78% resulting a lower error reduction of 45.30%. Further-

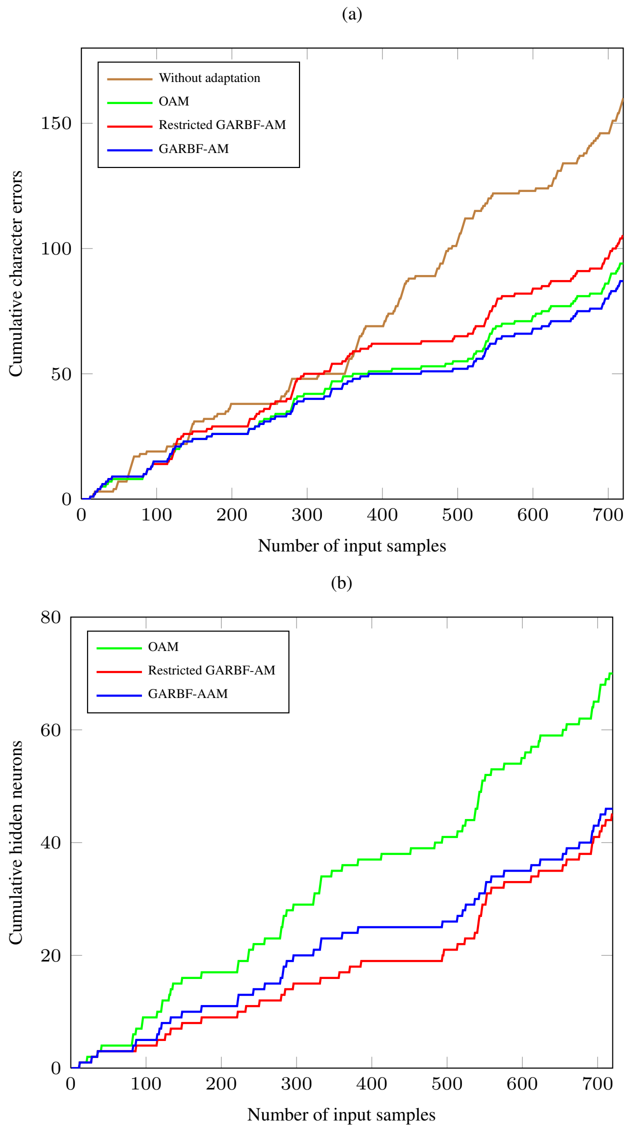


Figure. 4: Effectiveness of writer adaptation for writer w10 from LaViola dataset (a) cumulative character errors during adaptation against without adaptation (b) cumulative hidden units allocated during adaptation

Table 2: Error rate reduction on LaViola dataset using OAM and GARBF-AM

writer	Error rate (%) without adaptation	Error rate reduction (%)	
		OAM	GARBF-AM
w1	26.81	55.97	59.60
w2	25.14	36.47	44.19
w3	19.17	44.94	62.34
w4	21.81	38.87	50.34
w5	17.09	46.36	55.30
w6	19.45	35.73	52.85
w7	18.48	45.89	52.65
w8	16.67	38.35	50.87
w9	19.87	41.98	55.26
w10	22.23	39.40	45.30
w11	13.20	35.82	45.30
Average	20.00	41.79	52.33

Table 1: Recognition rate (RR) and number of hidden neurons (Nb neur) on LaViola dataset using OAM, Restricted GARBF-AM and GARBF-AM

writer	RR (%) without adaptation	OAM		Restricted GARBF-AM		GARBF-AM	
		RR (%)	Nb neur	RR (%)	Nb neur	RR (%)	Nb neur
w1	73.19	88.18	64	85.56	35	89.17	52
w2	74.86	84.00	76	84.72	52	85.97	63
w3	80.83	89.44	52	90.83	29	92.78	35
w4	78.19	86.67	58	85.42	36	89.17	44
w5	82.91	90.83	48	92.08	32	92.36	36
w6	80.55	87.50	61	88.33	39	90.83	43
w7	81.52	90.00	48	89.58	28	91.25	34
w8	83.33	89.72	48	91.81	27	91.81	42
w9	80.13	88.40	58	88.19	52	91.11	43
w10	77.77	86.53	71	87.50	45	88.19	48
w11	86.80	91.53	43	90.14	26	92.78	29
Average	80.00	88.43	57	88.56	37	90.94	43

more, GARBF-AM carried out for writer w1 the lowest error rate reduction of 8.38% compared to OAM algorithm. These results confirm that our algorithm is very useful and effective for the performance improvement of the writer-independent recognition system.

- *Performance evaluation of GARBF-AM using recall and precision:*

To conclude the performance evaluation, we present additional results with LaViola dataset using other performance measures which are recall, precision and F-score. Since we are in a multi-class context precision and recall are defined as follows. Precision for a class cl is the number of items correctly labeled as belonging to the positive class divided by the total number of elements labeled as belonging to the positive class. Recall for a class cl is the number of items correctly labeled as belonging to the positive class divided by the total number of elements that actually belong to the positive class. Precision and recall for a class cl are calculated using equations Eq.14.

$$P_{cl} = \frac{TP_{cl}}{TP_{cl} + FP_{cl}}, \quad R_{cl} = \frac{TP_{cl}}{TP_{cl} + FN_{cl}} \tag{14}$$

$$P = \frac{\sum_{cl=1}^{L} P_{cl}}{L}, \quad R = \frac{\sum_{cl=1}^{L} R_{cl}}{L} \tag{15}$$

When we average the values P_{cl} and R_{cl} we get the overall precision and recall (Eq.15). The F-score is calculated using Eq. 16.

$$F = \frac{2 * P * R}{P + R} \tag{16}$$

The Fig. 5 show the overall performance of the recognition system with and without adaptation using recall and precision measures. From Fig. 5, for all writers, we state that the precision is slightly larger to equal compared to recall. This proves that our adaptation system is precise as well as it is performing. Also, the curvatures (without and with adaptation) have almost the same slopes which show clearly that the writer adaptation improves the racall without the deterioration of the precision. The most higher enhancement is made for writer w1, for this reason we analyzed deeply its confusion matrix in the both cases without and with adaptation. We remind that every writer wrote 20 times each character. From the confusion matrix without adaptation we remark that the WIRS incorrectly labelled the majority of characters really belonging to a class1 to another class2. Which means the

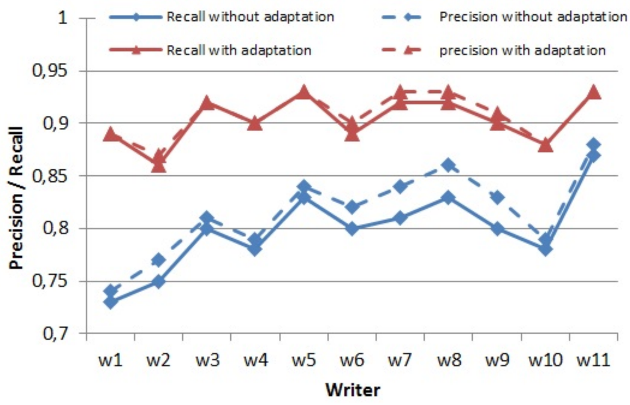


Figure. 5: Overall system recall and precision without and with adaptation per writer on LaViola dataset

classifier is somehow confused between class1 and class2. To carry out this analyze, we extract the following confusion between classes, shown in Table 3.

Table 3: Extracted confusion matrix of writer w1 using LaViola dataset

Without Adaptation										
	'f	'g'	'o'	'p'	'0'	'7'	'8'	'9'	P_{cl}	R_{cl}
'f'	1	0	0	19	0	0	0	0	0.5	0.05
'g'	0	1	0	0	0	0	0	19	1	0.05
'7'	0	0	9	0	1	0	10	0	0	0
With Adaptation										
	'f'	'g'	'o'	'p'	'0'	'7'	'8'	'9'	P_{cl}	R_{cl}
'f'	18	0	0	2	0	0	0	0	1	0.9
'g'	0	17	0	0	0	0	0	3	0.94	0.85
'7'	0	0	0	0	4	13	3	0	0.86	0.65

The striking confusion between classes made by the WIRS on the writing style of writer w1 are the pairs ('f','p'), ('g','9') and ('7','8'). From the Table 3 we make out the ability of the adapted system to correct the majority of errors. So, without adaptation and taking the character '7' as an example, the precision and the recall are $P_{cl} = 0$ and $R_{cl} = 0$. With adaptation the recognition system performance was improved to reach a precision of $P_{cl} = 0.86$ and recall of $R_{cl} = 0.65$. Which means that for precision, out of the times character '7' was predicted, 86% of the time the system was in fact correct. And for recall, it means that out of all the times character '7' should have been predicted only 65% of the characters were correctly predicted. In the same way, we plotted in Fig 6 the F-score reached by each writer's writing style in both cases without and with adaptation. We state that the overall performance of the recognition system is improved using GARBF-AM where the F-score is moved up at least by 0.62% for writer w11 and at the most by 2.10% for writer w1.

Figure. 6: Overall system F-score without and with adaptation per writer on LaViola dataset

- *Statistical comparison using a binomial test:*

We conducted in this study a statistical comparison of the GARBF-AM algorithm over the OAM and Restricted GARBF-AM algorithms. To this end, we used the binomial test [37, 38] using eq.(17). To explain this formula, we take as an example the binomial test between GARBF-AM and OAM.

$$E = \sum_{j=S}^{K_d} \frac{K_d!}{j!(K_d - j)!} p^j q^{K_d - j} \tag{17}$$

Where:
E: The p-value (probability of S success in K_d trials) using a binomial distribution,
K_d: The number characters for which the proposed GARBF-AM and OAM produce different results,
S (Success): The number of times GARBF-AM predicts the class label correctly rather than OAM,
F (Failure): The number of times OAM predicts the class label correctly rather than GARBF-AM,
p and q are the probability of success for GARBF-AM and OAM. In our case we assume that no difference between the two algorithms then $p = q = 0.5$.
The probability of S success in K_d trials (p-value) using a binomial distribution is reported in Table 4. First, we achieve the binomial test between the proposed algorithm GARBF-AM and the OAM algorithm. For writer w3, from the 720 test examples, the GARBF-AM and the OAM differ only in 22 characters. Among the 22 characters, GARBF-AM classifies accurately 17 characters (S=17) and OAM classifies accurately 5 characters (F=5).
The propability (E) for this case is $8.5\ 10^{-3}$. From this result, we can say that the proposed GARBF-AM is better than OAM with a high confidence for writer w3. Likewise for the other writers, the proposed GARBF-AM is marginally better than OAM. Similarly, we conducted the binomial test for Restricted GARBF-AM and results are given in Table 4. From

Table 4: Performance comparison using binomial test on LaViola dataset

Algorithms: GARBF-AM and OAM							
Writer	Binomial test			Writer	Binomial test		
	S	F	E		S	F	E
w1	7	5	0.38	w7	11	7	0.24
w2	16	12	0.28	w8	13	6	0.08
w3	17	5	8.5 e^{-3}	w9	12	9	0.33
w4	9	8	0.5	w10	17	7	3.2 e^{-2}
w5	10	6	0.22	w11	8	10	0.75
w6	11	8	0.32				
Algorithms: GARBF-AM and Restricted GARBF-AM							
Writer	Binomial test			Writer	Binomial test		
	S	F	E		S	F	E
w1	19	8	2.6 e^{-2}	w7	8	9	0.68
w2	23	10	1.7 e^{-2}	w8	25	11	1.44 e^{-2}
w3	5	0	0.03	w9	16	6	2.62 e^{-3}
w4	8	6	0.39	w10	24	6	7.15 e^{-4}
w5	10	1	5.9 e^{-3}	w11	9	2	3.27 e^{-2}
w6	13	6	8.35 e^{-2}				

the result, we can say that the performance of the GARBF-AM is better than Restricted GARBF-AM with high confidence.

- *Efficiency evaluation of GARBF-AM algorithm:*

We explored in this part another area of interest analyzing how the adaptation algorithm GARBF-AM reacts to errors made by the writer-independent recognition system (WIRS). In this experiment, for each writer of LaViola dataset we dealt with four pieces of information that are illustrated in Table 5:

⇨ Performance deterioration (Perf-D): is the number of correct classifications by the WIRS and that becomes incorrect during adaptation.

⇨ Performance improvment (Perf-I): is the number of incorrect classifications by the WIRS and that becomes correct during adaptation.

⇨ Persistent error (Pers-E): is the number of incorrect classifications by the WIRS and that remains incorrect during adaptation.

⇨ Persistent correct (Pers-C): is the number of correct classifications by the WIRS and that remains correct during adaptation.

Table 5: Efficiency analysis of GARBF-AM using the LaViola dataset

Writer	Pers-C	Perf-I	Perf-D	Pers-E	FCR (%)	TCR (%)
w1	517	125	10	68	1.90	64.77
w2	518	101	21	80	3.90	55.80
w3	575	93	7	45	1.20	67.39
w4	553	89	10	68	1.78	56.69
w5	585	80	12	43	2.01	65.04
w6	572	82	8	58	1.38	58.57
w7	574	83	13	50	2.21	62.41
w8	589	72	11	48	1.83	60.00
w9	564	92	13	51	2.25	64.34
w10	546	89	14	71	2.50	55.63
w11	615	53	10	42	1.60	55.79

The performance of a writer adaptation is estimated by its ability to identify and adapt unreliable WIRS responses. From Table 5, it can be seen that the performance deterioration (Perf-D) is too small compared to the performance Improvment (Perf-I). Moreover, from the persistent correct (Pers-C) information, we ascertain that the module adaptation using GARBF-AM algorithm keeps the efficiency of the WIRS without any performance degradation. On the other hand, to obtain a deeper analysis of the errors with adaptation, we calculated the false classified rate (FCR) given by eq.18 and the true classified rate (TCR) given by eq.19.

$$FCR = \frac{Perf\_D}{Pers\_C + Perf\_D} \tag{18}$$

$$TCR = \frac{Perf\_I}{Pers\_E + Perf\_I} \tag{19}$$

From Table 5, we can see that using GARBF-AM, the degradation of the recognition system performance is meaningless compared to its significant increase. The average FCR (2.05%) is meaningless compared to the average TCR (60.58%).

B. Writer adaptation using Multi-Environment dataset

The handwriting is the many spontaneous movements through which we can observe the ever-changing environment of the writer. During the process of writing, the words that we shape show how we feel (excitement, fear, anxiety, irritability or anger) and how we are (standing, sitting, lay down on the sofa, going up/down stairs, on train, by car, ...). Because, handheld devices can be used especially while the user is settled (sitting, standing, lay down on the sofa), while he is in mobile settings (walking, going up/down stairs) or when he is in a car, on a train or subway, our work is limited to the physical positions of the writer. However, the physical positions (environments) affect the users writing style with different degrees. The developed writer-dependent recognition systems used the written data while the user is sitting. To perform the writer adaptation we need to consider these environments to increase the performance of the writer-independent systems. In this section we describe the multi-environment dataset REGIM-MEnv which contains handwritten samples written in different environments (section IV-B.1) and we evaluate the performance of the proposed writer adaptation system using REGIM-MEnv dataset (section IV-B.2).

1) REGIM-MEnv dataset description

The REGIM-MEnv is a multi-environment writer-dependent dataset consisting of 36 different characters (a-z and 0-9), handwritten on a Samsung N5100 GALAXY Note 8.0. We collected some handwriting samples from five writers (three females and two males) using the Android application ISIgraphy [26] which is developped for the generation of online handwriting sample databases on touchscreen based devices. The REGIM-MEnv dataset contains five environments. Two mobile environments (walking, going up/down stairs) and three stationary environments (sitting at a desk, standing and in a car). Without any guidance or constraint, each person was asked to write ten times each character in each environment. The total number of characters contained in the dataset is 1800 per writer.

When the user is in mobile environments or standing, he/she holds a device with the nondominant hand and write characters using the other hand. This situation is exhausting. For this reason and to have a real dataset, the writer was asked to write during different periods in a day, at most four characters ten times each period. Fig. 7 displays the effect of the writers environment on his handwriting style. The ISIgraphy application is designed well enough to store handwritten data samples in large scales in user-given file names for specific users. Each file is made up of successive columns of data comprising x, y coordinates along the pen trajectory, pen pressure and time stamps. To carry out our experimental evaluation we have to convert files generated by ISIgraphy to UNIPEN format to be tested by the writer-independent recognition system.

2) Results

To test the effectiveness of the proposed writer-dependent recognition system with the multi-environment dataset REGIM-MEnv, we used the same parameters value as those

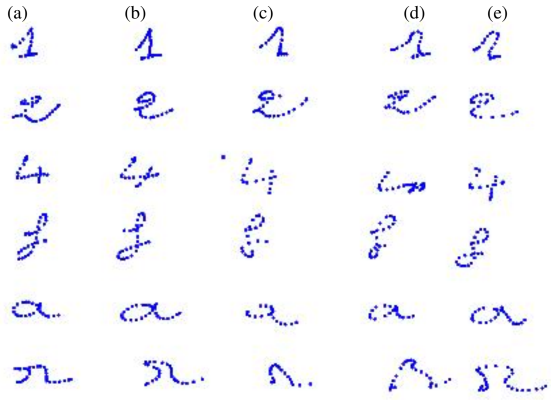


Figure. 7: Example characters written by writer w10 in each environment : (a) sitting, (b) standing, (c) walking, (e) going up/down stairs , (d) in a car

used for testing LaViola dataset (Section IV-A.2). In this section, we conducted the three studies as previously described (section IV-A.3) and we presented the experimental results on the multi-environment dataset using both OAM, Restricted GARBF-AM and GARBF-AM algorithms for sequential learning of the adaptation module.

- *Performance evaluation of GARBF-AM using accuracy:*

The performance comparison is displayed in Table 6. This table provides the recognition rate of the writer-independent recognition system (without adaptation) and the writer-dependent recognition system. Taking writer w4 as an example, the recognition rate without adaptation is 87.44% and was increased using OAM, Restricted GARBF-AM and GARBF-AM by 7.79%, 8.89% and 9.72%, respectively. Moreover, compared to OAM the Restricted GARBF-AM and GARBF-AM reduced the number of hidden neurons by 20.68% and 27.58%, repectively.

For more details, using GARBF-AM learning algorithm the minimal percentage of increase of the recognition rate is for writer w2 (6.68%) and the best percentage is for writer w5 (8.52%). Moreover, the proposed GARBF-AM decreased the number of hidden neurons by 4% for writer w2 and increased it slightly (1.85%) for writer w5. These results confirm that GARBF-AM algorithm, compared to OAM, performs the classification accuracy and reduces the number of hidden units. An exception made for the writer w5 where OAM and GARBF-AM reached almost the same performance. Furthermore, Fig. 8 shows the cumulative errors without and with adaptation and the total number of hidden neurons allocated depending on the used algorithm, for writer w1. From Fig. 8 we observe that GARBF-AM reduces the cumulative errors by allocating an optimal number of hidden neurons.

In addition, from Table 7, we can see that GARBF-AM increases the average error rate reduction from 55.93% to 59.96% compared to the OAM. Taking writer w4 as an example, the error rate reduction was improved using GARBF-AM compared to OAM from 58.83% to 67.68% resulting in a higher error reduction of 15.03%. For writers w1, w2, w3 the error rate was reduced by 23.38%, 2.44% and 2.88%, respectively. From this performance study, we can assume that the GARBF-AM outperforms the writer-independent recognition system. The hidden units allocated in the adaptation module represent well the possible diversity of writing styles of each writer by environment.

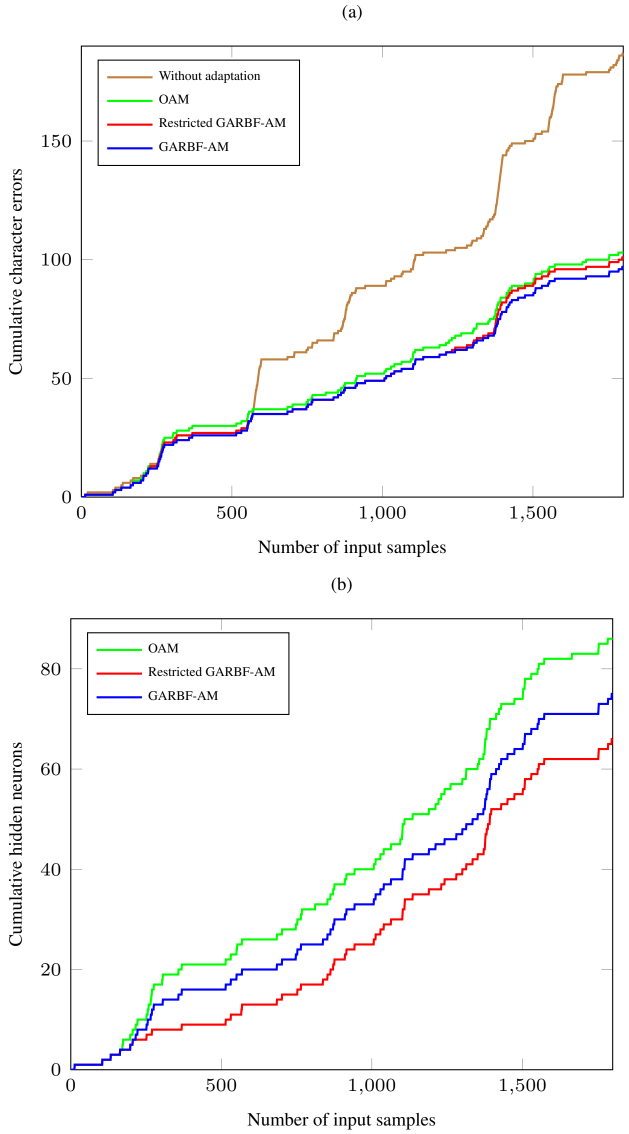


Figure. 8: Effectiveness of writer adaptation for writer w1 from REGIM-MEnv dataset (a) cumulative character errors during adaptation against without adaptation (b) cumulative hidden units allocated during adaptation

Table 7: Error rate reduction on REGIM-MEnv dataset using OAM and GARBF-AM

writer	Error rate (%) without adaptation	Error rate reduction (%)	
		OAM	GARBF-AM
w1	12.18	44.85	55.34
w2	10.00	61.30	62.80
w3	13.17	58.23	59.91
w4	12.56	58.83	67.68
w5	14.77	54.46	54.10
Average	12.53	55.53	59.96

Table 6: Recognition rate (RR) and number of hidden neurons (Nb Neur) on REGIM-MEnv dataset using OAM, Restricted GARBF-AM and GARBF-AM

writer	RR (%) without adaptation	OAM		Restricted GARBF-AM		GARBF-AM	
		RR (%)	Nb neur	RR (%)	Nb neur	RR (%)	Nb neur
w1	89.61	94.27	86	94.50	66	94.56	69
w2	90.00	96.13	50	96.22	40	96.28	46
w3	86.83	94.50	64	94.56	46	94.72	49
w4	87.44	94.83	58	95.22	46	95.94	42
w5	85.22	93.27	54	92.83	51	93.22	51
Average	87.82	94.60	63	94.66	50	94.94	51

- *Performance evaluation of GARBF-AM using recall and precision:*

Apart from using accuracy to judge the performance of the adapted system in the multi-environment context, it is always important to look at the confusion matrix to analyze the results by computing precision, recall and F-score using Eq.14, Eq.15 and Eq.16. The Fig. 9 show the overall performance of the recognition system on REGIM-MEnv dataset with and without adaptation using recall and precision measures. From Fig. 9, for all writers, we state that the precision and recall are improved when we adapt the response of the WIRS using the GARBF-AM. The highest amelioration is reached for writer w5. The precision is moved up by 10.58% and the recall by 9.41%. This case was further analyzed by exposing to the view an extract from the matrix confusion of writer w5 across the five environments. We remind that every writer wrote 10 times each caracter in each environment; so up to 50 characters. The objective of this analyze is to show and study how the GARBF-AM reacts in front of the confusion between classse made by the WIRS. The Table 8 reports the striking confusion between classes which are the following four pairs ('l','e'), ('n','m'), ('r','n') and ('1','l').

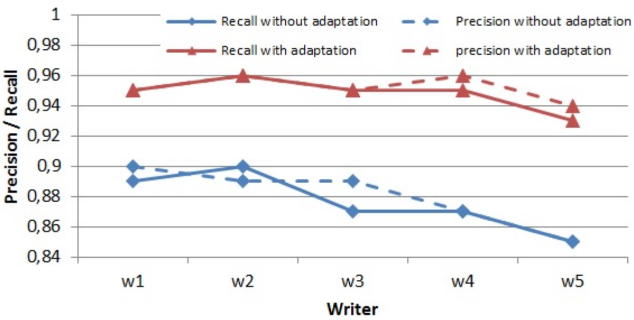


Figure. 9: Overall system recall and precision without and with adaptation per writer on REGIM-MEnv dataset

From the Table 8 we state the adapted system's ability to increase its performance by handling perfectly the various class confusions to correct the majority of errors. So, without adaptation and taking character '1' as an example, the precision and the recall are $P_{cl} = 0$ and $P_{cl} = 0$. With adaptation the recognition system performance is improved to reach a precision of $P_{cl} = 0.93$ and $P_{cl} = 0.82$. Which means that for precision, out of the times character '1' was predicted, 93% of the time the system was in fact correct. And for recall, it means that out of all the times character '1' should have been predicted only 82% of the characters were correctly predicted. In the same way, we plotted in Fig 10 the F-score reached by each writer's writing style in both cases without and with adaptation. We state that the overall performance of the recognition system is improved using GARBF-AM where the F-score is moved up at least by 6.14% for writer w1 and at the most by 9.99% for writer w5.

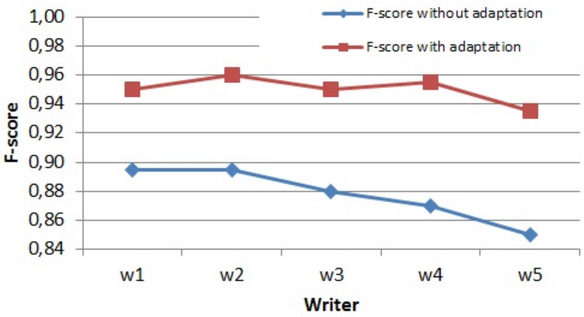


Figure. 10: Overall system F-score without and with adaptation per writer on REGIM-MEnv dataset

- *Statistical comparison using binomial test:*

We conducted in this study a statistical comparison of the GARBF-AM algorithm over the OAM and Restricted GARBF-AM algorithms using the binomial test and applying eq.(17). The results are reported in Table 9. For writer w4, from 1800 characters, the GARBF-AM and OAM differ only in 30 characters. Out of the 30 characters GARBF-AM classifies 25 characters correctly ($S = 25$). OAM, however, classifies 5 characters accurately ($F = 5$). For this case, the probability is $E = 1.62\ 10^{-4}$.
For the same writer w4, the propability using the binomial test between GARBF-AM and Restricted GARBF-AM is $E = 6.4\ 10^{-3}$. These results prove that the proposed algorithm is better than other algorithms with high confidence. Similarly, we conducted the binomial test for the other writers and we concluded that the performance of GARBF-AM is better than OAM and Restricted GARBF-AM.

Table 9: Performance comparison using binomial test on REGIM-MEnv dataset

Algorithms: GARBF-AM and OAM				Algorithms: GARBF-AM and Restricted GARBF-AM			
Writer	Binomial test			Writer	Binomial test		
	S	F	E		S	F	E
w1	17	12	0.22	w1	13	10	0.33
w2	5	9	0.9	w2	9	9	0.59
w3	15	11	0.27	w3	13	10	0.33
w4	25	5	$1.62\ e^{-4}$	w4	14	3	$6.4\ e^{-3}$
w5	8	10	0.75	w5	24	11	0.02

Table 8: Extracted confusion matrix of writer w5 using REGIM-MEnv dataset

Without Adaptation														
	'a'	'c'	'e'	'i'	'l'	'm'	'n'	'p'	'q'	'r'	'x'	'y'	'1'	P_{cl}	R_{cl}
'l'	0	0	36	0	14	0	0	0	0	0	0	0	0	0.22	0.28
'n'	0	0	0	0	0	38	12	0	0	0	0	0	0	0.19	0.24
'r'	0	0	0	0	0	0	36	0	0	10	0	0	4	0.83	0.20
'1'	1	0	0	0	44	0	0	2	2	0	1	0	0	0	0
With Adaptation using GARBF-AM															
	'a'	'c'	'e'	'i'	'l'	'm'	'n'	'p'	'q'	'r'	'x'	'y'	'1'	P_{cl}	R_{cl}
'l'	0	0	17	0	33	0	0	0	0	0	0	0	0	0.91	0.66
'n'	0	0	0	0	0	21	29	0	0	0	0	0	0	0.55	0.42
'r'	0	0	0	1	0	0	15	0	0	33	0	0	1	0.91	0.66
'1'	0	2	0	0	3	0	0	0	2	0	0	2	41	0.93	0.82

- *Efficiency evaluation of GARBF-AM algorithm:*

Table 10 reflects the deep analysis on REGIM-MEnv dataset for each environment (sitting, standing, walking, going up/down stairs and by car). Each writer wrote 360 character in each environment. The behaviour of the adaptation to each input character was analysed to check four parameters : persistent correct (Pers-C), performance improvment (Perf-I), performance deterioration (Perf-D) and persistent error (Pers-E). Also, we calculated the FCR (eq.18) and the TCR (eq.19).
From Table 10 we can see that there is no degradation of the writer-independent recognition system performance with our adaptation algorithm when referring to Perf-D information. In a sitting environment for writer w1, we reach the maximum Perf-D (8 characters) which generates an FCR of 2.44%. Considering the 5 environments, using Perf-I information, the average number of characters that are correctly classified after adaptation and are initially misclassified by the writer-independent recognition sytem is 20, 24, 29, 30 and 29 respectively for w1, w2, w3, w4, and w5. On the other hand, referring to Perf-D information, the average number of characters that are misclassified after adaptation and are initially correctly calssified by the writer-independent recognition system are 3, 2, 2, 2 and 2 respectively for w1, w2, w3, w4, and w5. From these results, we conclude that the proposed GARBF-AM outperforms the writer-independent recognition system without performance degradation. The performance of the adaptation system depending on Pers-E information, indicates how many characters were incorrect during adaptation. We remark that, for writer w5, the Pers-E rose a little bit which explains why the recognition rate decrease slightly compared to OAM (Table 6).

V. Conclusions

In this paper, a sequential learning algorithm called GARBF-AM for writer adaptation was presented. The principle of our work was to use an architecture of writer adaptation that can be applied to any recognition system. The writer-dependent recognition system incorporates an adaptation module (AM) at the output of any writer-independent recognition system. The AM, based on RBF-NN, operates with GARBF-AM and converts the writer-independent output into writer-dependent output. The GARBF-AM consists of two strategies which are growing and adjustment. The growing criteria use the idea of significance of hidden neurons. The adjustment consists of the update of two specific units (nearest and desired contributor) parameters using the standard LMS gradient descent to decrease the error each time no new unit is allocated. The writer adaptation performance was tested using the LaViola dataset and a multi-environment dataset REGIM-MEnv. The comparison with other sequential learning algorithms clearly shows that the proposed GARBF-AM, using the neuron significance concept in the growth criteria, improves the performance of the writer-dependent recognition system.

Acknowledgments

The authors would like to acknowledge the financial support of this work by grants from General Direction of Scientific Research (DGRST), Tunisia, under the ARUB program.

References

[1] J. Platt. A Resource-Allocating Network for Function Interpolation. *Neural computation*, 3 (2), pp. 213-225, 1991.

[2] V. Kadirkamanathan and M. Niranjan. A Function Estimation Approach to Sequential Learning with Neural Networks. *Neural computation*, 5 (6), pp. 954-975, 1993.

[3] B. Fritzke. Fast learning with incremental RBF Networks. *Neural Processing Letters*, 1 (1), pp.2-5, 1994.

[4] J. Platt and N. P. Matic. A Constructive RBF Network for Writer Adaptation. *Advances in Neural Information Processing Systems*, 9 (1), pp. 765-771, 1997.

[5] L. Yingwei, N. Sundararajan and P. Saratchandran. A sequental learning scheme for function approximation using minimal radial basis function (RBF) neural networks. *Neural Computation*, 9 (2), pp. 461-478, 1997.

[6] G. B Huang, P. Saratch and N. Sundararajan. A generalized growing and pruning rbf (GGAP-RBF) neural network for function approximation. *Neural Networks*, 16 (1), pp. 57-67, 2005.

[7] R. Zhanga, G. B Huang, N. Sundararajanb and P. Saratchandranb. Improved GAP-RBF network for classification problems. *Neurocomputing*, 70 (16-18), pp. 3011-3018, 2007.

[8] M. Bortman and M. Aladjem. A growing and pruning method for radial basis function networks. *Neural Networks*, 20 (6), pp. 1039-1045, 2009.

Table 10: Performance analysis of GARBF-AM using the REGIM-MEnv dataset

writer	Env	Pers-C	Perf-I	Perf-D	Pers-E	FCR (%)	TCR (%)
w1	Sitting	320	11	8	21	2.44	34.38
	Standing	315	16	3	26	0.94	38.10
	Walking	316	21	2	21	0.63	50.00
	Going up/down stairs	319	28	2	11	0.62	71.79
	By car	327	26	1	6	0.30	81.25
w2	Sitting	305	21	6	28	1.93	42.86
	Standing	321	24	2	13	0.62	64.86
	Walking	324	30	0	6	0	83.33
	Going up/down stairs	330	24	2	4	0.60	85.71
	By car	328	25	1	6	0.30	80.65
w3	Sitting	306	19	6	29	1.92	39.58
	Standing	309	32	1	18	0.32	64.00
	Walking	314	31	2	13	0.63	70.45
	Going up/down stairs	317	33	1	9	0.31	78.57
	By car	307	34	0	19	0	64.15
w4	Sitting	313	18	7	22	2.19	45.00
	Standing	311	33	1	15	0.32	68.75
	Walking	313	21	3	23	0.95	47.73
	Going up/down stairs	306	46	0	8	0	85.19
	By car	320	33	0	7	0	82.50
w5	Sitting	317	16	8	19	2.46	45.71
	Standing	309	26	0	25	0	50.98
	Walking	293	36	3	28	1.01	56.25
	Going up/down stairs	299	35	0	26	0	57.38
	By car	305	35	0	20	0	63.64

[9] S. Suresh, K. Dong and H. J. Kimb. A sequential learning algorithm for self-adaptive resource allocation network classifier. *Neurocomputing*, 73 (16-18), pp. 3012-3019, 2010.

[10] X. Y. Zhang and C. L. Liu. Writer Adaptation with Style Transfer Mapping. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 99 (1), pp. 1-15, 2012.

[11] L. Jin, K. Ding and Z. Huang. Incremental learning of LDA model for Chinese writer adaptation. *Neurocomputing*, 73 (10-12), pp. 1614-1623, 2010.

[12] L. Haddad, T. M. Hamdani, M. Kherallah, A. M. Alimi : Improvement of On-line Recognition Systems Using a RBF-Neural Network Based Writer Adaptation Module. In *Proceedings of the International Conference on Document Analysis and Recognition*, pp. 284-288, 2011.

[13] L. Haddad, T. M. Hamdani and A. M. Alimi : Improved Neural Based Writer Adaptation For On-line Recognition Systems. In *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, pp. 1175-1180, 2013.

[14] L. Haddad, T. M. Hamdani and A. M. Alimi : OHRS-MEWA: On-line Handwriting Recognition System with Multi-Environment Writer Adaptation. In *Proceedings of the International Conference on Frontiers in Handwriting Recognition*, pp. 335-340, 2014.

[15] S. Madhvanath, D. Vijayasenan and T. M. Kadiresan. LipiTk: A Generic Toolkit for Online Handwriting Recognition. In *Proceedings of the ACM Special Interest Group on Computer Graphics (SIGGRAPH)*, 2007.

[16] L. Prevost and L. Oudot. Self-Supervised Adaptation for Online Script Text Recognition. *Letters on Computer Vision and Image Analysis*, 5 (1), pp. 87-97, 2005.

[17] A. Nakamura : A Method to Accelerate Writer Adaptation for On-Line Handwriting Recognition of a Large Character Set. In *Proceedings of the International Workshop on Frontiers in Handwriting Recognition*, pp. 426-431, 2004.

[18] H. Mouchere, E. Anquetil, and N. Ragot. Writer Style Adaptation in On-line Handwriting Recognizers by a Fuzzy Mechanism Approach : The ADAPT Method. *International Journal of Pattern Recognition and Artificial Intelligence*, 21 (1), pp. 99-116, 2007.

[19] M. Liwicki, A. Schlapbach, and H. Bunke : Writer-Dependent Recognition of Handwritten Whiteboard Notes in Smart Meeting Room Environments. In *Proceedings of the International Conference on Development and Application Systems*, pp. 151-157, 2008.

[20] H. Miyao and M. Maruyama : Writer Adaptation for Online Handwriting Recognition System Using Virtual Examples. In *Proceedings of the International conference on Document Analysis and Recognition*, pp. 1156-1160, 2009.

[21] N. C. Tewari and A. M. Namboodiri : Learning and Adaptation for Improving Handwritten Character Recognizers. In *Proceedings of the International conference on Document Analysis and Recognition*, pp. 86-90, 2009.

[22] H. Cao, S. Prasad, S. Saleem, and P. Natarajan : Unsupervised HMM Adaptation Using Page Style Clustering. In *Proceedings of the International conference on*

Document Analysis and Recognition, pp. 1091-1095, 2009.

[23] J. LaViola and R. Zeleznik. Apractical approach for writer-dependent symbol recognition using a writer-independent symbol recognizer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29 (11), pp. 1917-1926, 2007.

[24] A. Bouchachia, M. Prossegger and H. Duman : Semi-Supervised Incremental Learning. In *Proceedings of the IEEE International Conference on Fuzzy Systems*, pp. 1-6, 2010.

[25] A. Delaye and E. Anquetil, HBF49 feature set: A first unified baseline for online symbol recognition. *IEEE Transactions on Pattern Recognition*, 46 (1), pp. 117-130, 2013.

[26] A. Das and U. Bhattacharya : ISIgraphy: A Tool for Online Handwriting Sample Database Generation. In *Proceedings of the National Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics*, pp.1-4, 2013.

[27] K. Okamato, S. Ozawa and S. Abei: A Fast Incremental Learning Algorithm of RBF Networks with Long-Term Memory. In *Proceedings of the International Joint Conference on Neural Networks*, pp. 102-107, 2003.

[28] S. Ozawa : An autonomous incremental learning algorithm of ressource Allocating Network for online pattern recognition. In *Proceedings of theInternational Joint Conference on Neural Networks*, pp. 1-8, 2010.

[29] Qinggang Meng, Baihua Li, Horst Holstein and Yonghuai Liu. Parameterization of point-cloud freeform surfaces using adaptive sequential learning RBF networks. *Pattern Recognition*, 46 (8), pp. 2361-2375, 2013.

[30] A. Brakensiek, A. Kosmala, A and G. Rigoll : Comparing adaptation techniques for on-line handwriting recognition. In *Proceedings of the International Conference on Document Analysis and Recognition*, pp. 486-490, 2001.

[31] W. Kienzle and K. Chellapilla : Personalized handwriting recognition via biased regularization. In *Proceedings of the International Conference on Machine Learning*, pp. 457-464, 2006.

[32] K. Chellapilla, P. Simard and A. Abdulkader : Allograph based writer adaptation for handwritten character recognition. In *Proceedings of the International Workshop on Frontiers in Handwriting Recognition*, 2006.

[33] M. Szummer and C. M. Bishop : Discriminative Writer Adaptation. In *Proceedings of the International Workshop on Frontiers in Handwriting Recognition*, pp. 293298, 2006.

[34] Li Jun and T. Duckett: Some Practical Aspects on Incremental Training of RBF Network for Robot Behavior Learning. In *Proceedings of the World Congress on Intelligent Control and Automation*, pp. 2004-2006, 2008.

[35] C. Viard-Gaudin, P-M. Lallican, S. Knerr and P. Binter: The IRESTE On/Off (IRONOFF) Dual Handwriting Database. In *Proceedings of the International Conference on Document Analysis and Recognition*, pp. 455-458, 1999.

[36] C. Panchapakesan, M. Palaniswami, D. Ralph and C. Manzie. Effects of moving the center's in an RBF network. *IEEE Transactions on Neural Networks*, 13 (6), pp. 12991307, 2002.

[37] S. Suresh, N. Sundararajan and P. Saratchandran. A sequential multi-category classifier using radial basis function networks. *Neurocomputing*, 71 (79), pp. 13451358, 2008.

[38] S.L. Salzberg. On comparing classifiers: pitfalls to avoid and a recommended approach. *Data Mining and Knowledge Discovery*, 1 (3), pp. 317328, 1997.

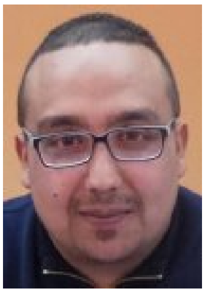
[39] C. Vielhauer and R. Steinmetz. Handwriting: Feature Correlation Analysis for Biometric Hashes. *Journal on Applied Signal Processing*, 4, pp. 542-558, 2004.

[40] A. K. Jain, A. Ross and S. Prabhakar. An Introduction to Biometric Recognition. *IEEE Transactions on Circuits and Systems for Video Technology*, 14 (1), pp. 4-20, 2004.

Author Biographies

Lobna Haddad was born in Sfax, Tunisia, in 1979. She received the Graduate degree in management computing and the Masters degree in information systems and new technologies from the Faculty of Economics and Management of Sfax , Tunisia in 2001 and 2004, respectively. She has been pursuing the Ph.D. degree with the Research Groups in Intelligent Machines (REGIM-Lab). Her research interests include pattern recognition, incremental learning, and especially classifier adaptation. She is a student member of the IEEE.

Tarek M. Hamdani (M01, SM'12) was born in Tunis, Tunisia, in 1979. He received the Ph.D. degree in 2011 and his M.S. degree in 2003, in Computer Science Engineering from the University of the Sfax, Tunisia. He is currently pursuing the PostDoc degree in Computer Science Engineering in the National School of Engineers, Sfax, Tunisia. He is teaching computer science at the Taibah University since 2012, and was teaching at the University of Sfax and the University of Monastir since 2003. He focuses his research on intelligent pattern recognition, learning, and analysis of large scale complex systems. His research interest includes applications of intelligent methods, neural networks, fuzzy logic, and genetic algorithms to pattern recognition. He is a Reviewer of the Pattern Recognition Letters, Neurocomputing, Neural processing Letter, TNN, and the Soft Computing journal. He is an IEEE member of CIS and SMC societies.



Wael Ouarda Computer Science Engineer, University of Sfax. He is a researcher and a member of the REsearch Groups in Intelligent Machines (REGIM-Lab). His research interests include Computer Vision and Image and video analysis. These research activities are centered around Video Events Detection and Pattern Recognition. He is a Graduate member of IEEE. He is member of the organization committee of several International Conferences.

Adel M. Alimi (IEEE Student Member91, Member96, Senior Member00). He graduated in Electrical Engineering in 1990. He obtained a PhD and then an HDR both in Electrical & Computer Engineering in 1995 and 2000 respectively. He is full Professor in Electrical Engineering at the University of Sfax, ENIS, since 2006. He is founder and director of the REGIM-Lab. on intelligent Machines. Prof. Alimi was advisor of 52 PhD Thesis. He is the holder of 15 Tunisian patents. He served also as Expert evaluator for the European Agency for Research since 2009. Prof. Alimi research interests include iBrain (Evolutionary computing, Swarm intelligence, Artificial immune systems, Fuzzy Sets, Uncertainty analysis, Fractals, Support vector machines, Artificial neural networks, Case Based, Reasoning, Wavelets, Hybrid intelligent systems, Nature inspired computing techniques, Machine learning, Ambient intelligence, Hardware implementations, Multi-Agent Systems, iRobotics, Multi-Robots, Autonomous Robots) & iPerception (Classification, Classifiers Combination, Features Extraction, High Dimension Classification Problems, Handwriting Recognition, Handwriting Modeling, Motor Control, Perception, OCR, Arabic Script, Historical Documents Analysis, iDocument, iVision, Remote Sensing, Affective Computing) & iData (Big Data, iWeb, Biometry, Data Mining, Web Mining, Cloud Computing). Prof. Alimi served as associate editor and member of the editorial board of several international scientific journals (IEEE Transactions on Neural Networks and Learning Systems, IEEE Transactions on Fuzzy Systems, Pattern Recognition Letters, NeuroComputing, Neural Processing Letters, International Journal of Image and Graphics, Neural Computing and Applications, International Journal of Robotics and Automation, International Journal of Systems Science, EURASIP Journal on Advances in Signal Processing, Journal of Universal Computer Science). He was guest editor of several special issues of international journals (e.g. Fuzzy Sets & Systems, Soft Computing, Journal of Decision Systems, Integrated Computer Aided Engineering, Systems Analysis Modelling and Simulations).

Ajith Abraham is the Director of Machine Intelligence Research Labs (MIR Labs), a Not-for-Profit Scientific Network for Innovation and Research Excellence connecting Industry and Academia. The Network with Head quarters in Seattle, USA has currently more than 1,000 scientific members from over 100 countries. He also works as a Research Professor in VSB-Technical University of Ostrava, Czech Republic.
As an Investigator / Co-Investigator, he has won research grants worth over 100+ Million US$ from Australia, USA, EU, Italy, Czech Republic, France, Malaysia and China. Dr. Abraham works in a multi-disciplinary environment involving machine intelligence, cyber-physical systems, Internet of things, network security, sensor networks, Web intelligence, Web services, data mining and applied to various real world problems. In these areas he has authored / coauthored more than 1,000+ research publications out of which there are 100+ books covering various aspects of Computer Science. One of his books was translated to Japanese and few other articles were translated to Russian and Chinese. About 800+ publications are indexed by Scopus and over 450+ are indexed by Thomson ISI Web of Science. Some of the articles are available in the ScienceDirect Top 25 hottest articles. He has 700+ co-authors originating from 40+ countries. Dr. Abraham has more than 21,000+ academic citations (h-index of 72 as per google scholar). He has given more than 100 plenary lectures and conference tutorials (in 20+ countries). For his research, he has won seven best paper awards at prestigious International conferences held in Belgium, Canada Bahrain, Czech Republic, China and India. Since 2008, Dr. Abraham is the Chair of IEEE Systems Man and Cybernetics Society Technical Committee on Soft Computing (which has over 200+ members) and served as a Distinguished Lecturer of IEEE Computer Society representing Europe (2011-2013). Under his direct academic supervision, 12 students received Ph.D. degrees and is currently supervising 10 Ph.D. students in different Universities in Europe, USA, Africa and India. He has examined over 100 Ph.D. theses. Currently Dr. Abraham is the editor-in-chief of Engineering Applications of Artificial Intelligence (EAAI) and serves/served the editorial board of over 15 International Journals indexed by Thomson ISI. He is actively involved in the organization of several academic conferences, and some of them are now annual events. Dr. Abraham received Ph.D. degree in Computer Science from Monash University, Melbourne, Australia (2001) and a Master of Science Degree from Nanyang Technological University, Singapore (1998). More information at: http://www.softcomputing.net