# PERFORMANCE OF MLP AND RBF NEURAL NETWORKS ON ARABIC TEXT CATEGORIZATION USING SVD

*Fouzi Harrag*[*], *Aboubekeur Hamdi-Cherif*[†], *Eyas El-Qawasmeh*[‡]

**Abstract:** Text categorization is based on the idea of content-based texts clustering. An Artificial Neural Network (ANN) or simply Neural Network (NN) classifier for Arabic texts categorization is proposed. The Singular Value Decomposition (SVD) is used as preprocessor with the aim of further reducing data in terms of both size and dimensionality. Indeed, the use of SVD makes data more amenable to classification and the convergence training process faster. Specifically, the effectiveness of the Multilayer Perceptron (MLP) and the Radial Basis Function (RBF) classifiers are implemented. Experiments are conducted using an in-house corpus of Arabic texts. Precision, recall and F-measure are used to quantify categorization effectiveness. The results show that the proposed SVD-Supported MLP/RBF ANN classifier is able to achieve high effectiveness. Experimental results also show that the MLP classifier outperforms the RBF classifier and that the SVD-supported NN classifier is better than the basic NN, as far as Arabic text categorization is concerned.

Key words: *Singular value decomposition, neural network, MLP, RBF, arabic text categorization, natural language processing*

*Received: July 18, 2009*
*Revised and accepted: March 5, 2010*

## 1. Introduction

Text categorization is the process of clustering texts into one or more predefined categories based on their content. Due to the increased availability of documents in digital form and the rapid growth of online information, text categorization has become one of the key techniques for handling and organizing text data.

[*]Fouzi Harrag – Corresponding author
Computer Science Department, Farhat Abbas University, Setif, 19000, Algeria, E-mail: hfouzi2001@yahoo.fr, Tel.: +213772 298 697, +21336918898, Fax: +21336912034.

[†]A. Hamdi-Cherif
Computer College, Qassim University Buraydah, 51452, Saudi Arabia, E-mail: elhamdi62@gmail.com.

[‡]E. El-Qawasmeh
Computer Science Department, JUST, Amman, 25000, Jordan, E-mail: eyas@usa.net

Automatic text categorization has been used in many applications, such as real time sorting of files into folder hierarchies, topic identifications, dynamic task-based interests, automatic meta-data organization, text filtering and documents organization for databases and web pages [37, 39, 47]. Several methods have been used for text classification [38, 39, 46], such as: Support Vector Machine (SVM), K-Nearest Neighbor (KNN), Artificial Neural Networks, Naïve Bayes Classifier, and Decision Trees, among others.

However, a majority of Machine Learning algorithms have been tested in automatic Arabic text classification. The work in Arabic Text Classification using Neural Network is very limited. Neural network is a popular classification method. It can handle linear and nonlinear problems for text categorization. Both of linear [29] and nonlinear [25] classifier can achieve good results [48]. Neural networks have been widely applied by many researchers to classify text documents with different types of feature vectors. Wermeter [44] has used the document title as the vectors to be used for a document categorization. Lam *et al.* [25] have used the principal component analysis (PCA) method as a feature reduction technique of the input data to the neural networks.

The goal of this paper is to present and compare results obtained in Arabic text categorization using Neural Network (NN) classifier. The remainder of this paper is organized as follows: Section 2 summarizes related works in Arabic text categorization. Section 3 presents the main aspects of the MLP and RBF NNs. Section 4 provides the details of our design for the NN method to classify Arabic texts using the Singular Value Decomposition (SVD) to reduce the space of the features. Section 5 reports the results of our performance evaluation. Finally, we conclude in Section 5 with directions for future work.

## 2. Related Work in Arabic Text Categorization

Arabic is first of all a sacred language, i.e. the vehicle of the sacred revelation, namely Al-Qur'an, or Reading by Excellence, accepted by Muslims as the very Word of Allah. It is the sole liturgical language of Islam, i.e. no Islamic rituals are undertaken in any other language. It is a political language since it has been accepted in the United Nations as its official language. It is the vernacular language of 250 million Arabs scattered in 22 countries. For liturgical reasons, Arabic language is the lingua franca of more than 1.5 billion Muslims. To our knowledge, no language in the history of mankind has ever had all these features - combined. Because of its central position in the Islamic and Arab world, Arabic language is prone to play a central role in the world.

Arabic belongs to the Semitic family of languages, which includes Akkadian, Aramaic, Ethiopic, Hebrew, Phoenician, Syriac and Ugaritic. The Arabic script was derived from the Aramaic via the Nabatean cursive script. As is the case with all Semitic languages, the script is written from right to left, and this script has traditionally been represented in converted (Romanized) form in Western academic and computerized environments [20].

Classifying Arabic text is different than classifying English language because Arabic is highly inflectional and derivational language, which makes morphological analysis a very complex task. Processing text written in Arabic is a nontrivial task because of the richness of the language. A word can have many forms according to its location in a sentence; letters can have several styles according to their position in a word and their neighboring letters. The use of diacritics in Arabic makes ambiguity more difficult. For example, the root MLK with different diacritics could mean MuLK=Dominion, MaLiK=King, MaLaKa=he possesses, MaLaKun=Angel, MaaLiK=Lord, MaLlaKa=he offered.

The sizes of the feature vectors can grow very large for large documents, and therefore stemming was used as a filtering mechanism [11]. Also, in Arabic scripts some of the vowels are represented by diacritics that are usually left out in the text, which create ambiguity in that text. In addition, Arabic scripts do not use capitalization for proper nouns which are necessary in classifying documents classification [17].

Many researchers have been working on text categorization in English and other European languages. However few researchers have worked on text categorization for Arabic language. El-Kourdi *et. al.* [13] used Naive Bayes classifier for automatic Arabic document classification. The average accuracy reported was about 68.78%. Sawaf *et. al.* [38] used statistical classification methods, such as maximum entropy to classify and cluster News articles. The best classification accuracy they reported was 62.7%.

El-Halees [12] described a method based on association rules to classify Arabic documents. The classification accuracy reported was 74.41%. Duwairi [11] proposed a distance-based classifier for categorizing Arabic text. The average accuracy reported was 0.62 for the recall and 0.74 for the precision. Syiam *et. al.* [41] experimental results show that the suggested hybrid method of statistical and light stemmers is the most suitable stemming algorithm for Arabic language and gives generalization accuracy of about 98%. Mesleh [28] described a Support Vector Machines (SVMs) based text classification system for Arabic language articles. The system effectiveness for Arabic data set in term of F-measure is 88.11. Al-Harbi *et. al.* [1] evaluated the performance of two popular classification algorithms (SVM and C5.0/C4.5) on classifying Arabic text using seven Arabic corpora. The SVM average accuracy is 68.65%, while the average accuracy for the C5.0 is 78.42%.

## 3. Artificial Neural Networks

In this section, two types of NNs are briefly discussed with reference to the structures and the parameters. The main differences among these are also briefly discussed. Readers are referred to [4, 43, 19] for further details. Lately, Neural Networks (NNs) have been extensively used in real world applications, as they can be trained to approximate the responses originated from most of processing systems. This behavior can be modeled in a way that makes future estimations to similar inputs feasible and with good results. In practice, there are two kinds of NN architectures, the feedforward NNs and the feedback or recurrent ones applied in totally different problem domains [19]. In our framework, two kinds of NNs with no feedback loops are used and tested over their performance on the classification space: the Multilayer Perceptron (MLP) and the Radial Basis Function (RBF) classifiers as the most appropriate for on-line function approximation [19].

The comparison of these two NN architectures has already been studied in various areas of research, such as dynamic systems [31], channel equalization in signal processing [22], voice recognition [15], and whenever efficient, stable and low resource real time estimation is required (given that the utilized NNs have a small number of nodes).

## 3.1 Multi layer back-propagation neural network

A Multilayer Perceptron (MLP) is a common NN architecture applied in many areas of research, when solutions to diverse and difficult problems are required [6]. Typically, an MLP is a structure trained to map specific inputs to outputs through nonlinear functionality. This mapping is performed by passing inputs to outputs through one or more strongly connected hidden layers of computation nodes and a final layer of output nodes. Critical parameters affecting the NN's performance are: the number of the hidden layers, their corresponding neurons, the NN's weights and the hidden layers' transfer functions. The former are decided by the complexity of the problem and most of the time require extensive experiments to identify an adequate solution [16]. Regarding the network weights, the MLPs use the error back-propagation algorithm [35] to train their values on the supervised learning phase. For the transfer functions we can select among various different species [10].

Using neural networks, the problem of text categorization can be solved by using back-propagation. The classification decision for any document of reasonable size is based on the combined evidence from many sources. Each word is a source to classify a document [32]. In this paper, a three layer feed-forward neural network with hyperbolic tangent (*tanh*) activation function in the hidden layer, followed by a linear output layer, is employed. The neural network is trained with back-propagation algorithm. The inputs are the components of the document vector, and the outputs are the document categories. The basic idea is first to train the classifier with known labeled input/output pairs. In this phase, each document vector is associated with some document category. Secondly, during the test, after training, the classifier is used to categorize unknown documents, based on previous training, within a prescribed error. Based on the test results, additional training might be necessary. The structure of the three layered back-propagation neural network is shown in Fig. 1.

## 3.2 Radial Basis Function Neural Network (RBF-NN)

Generally speaking, a Radial Basis Function (RBF) is any real-valued function whose value depends only on the distance from some point, taken as the origin. If $x$ is a point in an $n$-dimensional vector space and $||x||$ is the Euclidian distance (or length) of the vector from the origin to that point, then an RBF can be expressed as: $\phi(x) = \phi(||x||)$. More generally, if we choose some other point $c$, called a *center*, that is different from the origin, then the RBF can alternatively be expressed as a function depending on the distance between this point $x$ and the center $c$, *i.e.* : $\phi(x,c) = \phi(||x\text{-}c||)$. The norm is usually Euclidean distance, although other distance functions are also possible, usually in conjunction with ill conditioning situations.

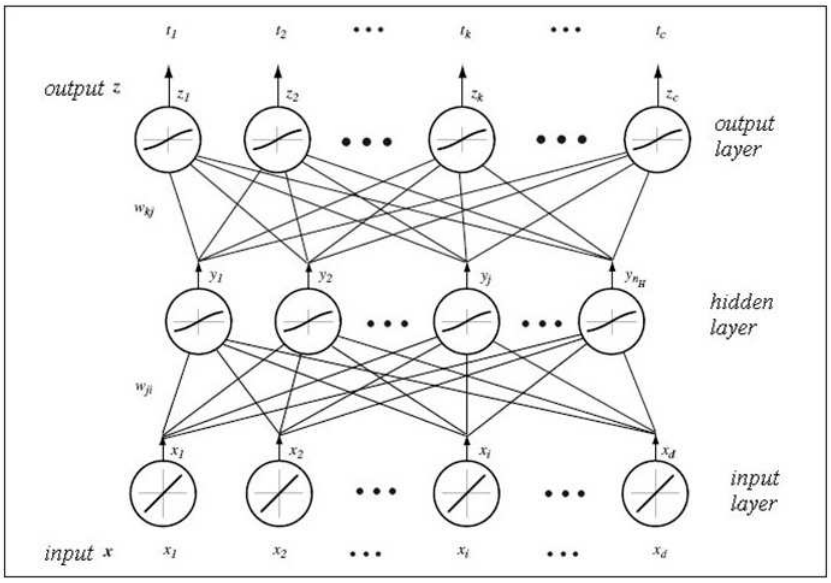


**Fig. 1** *Typical three layered back-propagation NN.*

A Radial Basis Function Neural Network (RBF-NN) is a feed-forward artificial neural network that uses radial basis functions as activation functions. It is based on linear combination of radial basis functions. It has been shown that RBF-NN had a simple structure and many excellent performances, such as best approximation, global optimum, approximation precision of the nonlinear systems and convergence rate [23]. Therefore, RBF-NN has been widely used for pattern classification, functional approximation, spline interpolation, signal processing, mixture models and many other fields.

Each RBF is a fixed two-layer NN that hides all nonlinearities in its special hidden layer and performs a linear combination in the output layer [19]. The parameters that determine the output values are the centers of the hidden RBF units and the weights of the synapses from the hidden to the output layer. The Orthogonal Least Squares (OLS) algorithm [8] is used for the selection of the RBF units' centers from the input data set, as the most appropriate and efficient method for center selection in a way that the size of the network is kept small and this process is completed with adequate complexity for on-line applications. For the network's weights, the linear Least Squares [9] (LS) algorithm is selected, satisfying similar criteria. Fig. 2 shows the structure of the RBF Neural Network [5].

The number of node in the network's input layer, hide layer and output layer is $d$, $h+1$ and $q$, respectively. From the input layer to the hide layer, each component in the Input vector transmits to each hide nodes without any change. Each hide layer's node function is the kernel function, and then the output of each node in the hide layer is:

$$\phi_i = exp(-\sum_{j=1..d}(x_{kj}-c_{ij})^2/\delta_i), i = 1, 2, \ldots, h+1, \tag{1}$$

where, $x_{ij}$ is the $j$-th component of the $k$-th input instance, $c_{ij}$ is the $j$-th component of center $c_i$ of this node, $\delta_i$ is the scaling parameter. The output value of the $n$-th unit ($1 \leq n \leq m$) of network output nodes is:

$$y_n = \sum_{j=1..h+1} \phi_i w_{nj}, i = 1, 2, \ldots, h+1, \tag{2}$$

where $w_{nj}$ is the weight of the connection of the $j$-th hide unit node to the $n$-th output node.

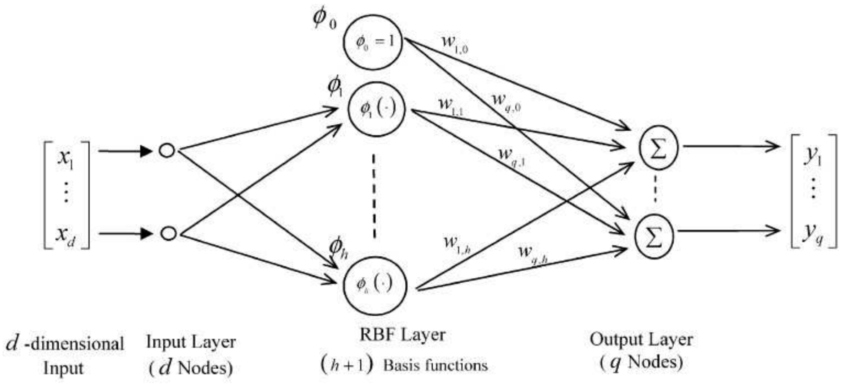


**Fig. 2** *Structure of a typical RBF neural network.*

## 3.3 MLP vs. RBF-NN

These two great NN families differ from each other in several important points. One important difference is that an MLP is most of the times a multi-layer network, whereas each RBF-NN consists of only one hidden layer with radial basis function neurons. In a typical implementation, the hidden nodes and the output nodes of a MLP-NN share a common computation and neuronal model, while the RBF-NN neurons of the RBF hidden layer play a totally different role in comparison to the output nodes, which are most of the times linear and perform a linear combination of the hidden layer neurons' responses. Additionally, the Gaussian nature of the RBF-NN neurons' activation functions contains the responses to local areas (local approximator) of the output space. On the other hand, each MLP neuron can answer with a value that belongs anywhere on the output space (global approximator). This locality that is observed on the RBF's responses leads to increased neurons numbers, and that is the reason why, theoretically, smaller MLPs are adequate for specific problems compared to the RBFs [30].

## 4. Arabic Text Categorization Model

### 4.1 Filling the term-document matrix

This section discusses how the proposed NN method is used Arabic text categorization. The processing *via* the NN method involves three basic steps, namely *data pre-processing*, *training* and *testing*, as depicted in Fig. 3. In our case, the *Feature Extraction* phase refers to data pre-processing using SVD method, detailed below. For the training dataset, once we obtain the selected features, we feed them into the neural network and generate a text classifier. For each test text, we use the classifier to verify the efficiency of NN model.

Before we embark on the description of the categorization process, we need to introduce the so-called term-document (T-D) matrix, as proposed by Salton [36]. The T-D matrix describes the frequency of terms that occur in a given collection of documents. In this matrix, each term used in one text dataset is represented by one column and each row corresponds to one text. The elements $a_{ij}$ of the T-D Matrix give the frequency of terms used in one text. There are various schemes for determining the value that each entry in the matrix should take. One such scheme is the term frequency inverse document frequency (TF-IDF).

The whole process includes several steps. First, all texts are transformed into term-document matrix (T-D matrix). Second, T-D matrix is transformed, based on SVD to reduce its dimensionality. Third, the transformed T-D matrix is divided into two parts: training set and testing set. Generally, the testing set is about one third of the whole dataset. Then, the training set is used to train the NN model to learn new patterns. Finally, the testing set is fed into NN classifier and allocated to newly learned category.

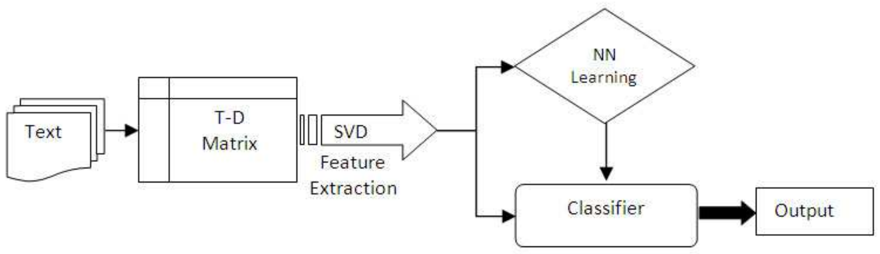


**Fig. 3** *Overview of SVD-Supported NN text classifier.*

### 4.2 Preprocessing

#### 4.2.1 Basic phases

The document in text categorization system must pass through a set of steps: document conversion, which converts different types of documents into plain text, stop words like prepositions and particles, which are considered insignificant words and must be removed. Words must be stemmed after stop words removal. Stemming is a root-finding process that consists in removing the affixes from the word [2, 3, 7, 24, 26, 41]. After applying preprocessing routines, document is passed by document indexing process, which involves creation of internal representation of the document. The first phase of the indexing process consists of the construction of the super vector containing all terms that appears in all the documents of the corpus. The second phase is term selection, which can be seen as a form of dimensionality reduction by selecting a subset of terms from the full original set of terms in the super vector according to some criteria, this subset is expected to yield the best effectiveness, or the best compromise between effectiveness and efficiency [27, 34, 45]. The third phase is term weighting in which, for every term selected in the second phase and for every document, a weight is computed, which represents how much this term contributes to the discriminative semantics of the document [36].

## 4.3 Singular value decomposition (SVD)

SVD produces a new representation space of the observations starting from the initial descriptors by preserving the proximity between the examples [33]. These new features known as "factors" or "latent variables" have several very advantageous properties: (a) their interpretation very often allows to detect patterns in the initial space; (b) a very reduced number of factors allows to restore information contained in the data; (c) the new features form an orthogonal basis, making learning algorithms, such as linear discriminant analysis, work well [18]. This process is often used in microarray data analysis [42] or text retrieval [21] fields where the initial number of descriptors is very high and where the dimensionality reduction is crucial before data analysis.

As stressed above, for SVD, the whole system is defined with the word-document co-occurrence matrix, in which each row represents a word and each column represents a document. The elements of the matrix contain the frequencies with which the words appear in the documents. Looking at this matrix, the documents appear as vectors in a high-dimensional space where the orthonormal axes represent the vocabulary words. Consider the words vectors, *i.e.* the rows in the word-document co-occurrence matrix. If we multiply the co-occurrence matrix by its transpose, we obtain the correlation matrix of the words. This matrix indicates how much words are correlated, or carry the same information. If an eigenvalue decomposition is performed on this matrix, the most significant information axes can be chosen and words can be projected in this informative subspace. Performing the eigenvalue decomposition of the correlation matrix is equivalent to performing the singular value decomposition (SVD) of the co-occurrence matrix. The most informative singular vectors are then considered defining a latent semantic subspace. The neglected less informative singular vectors are supposed to carry noise and redundant information due to style, confusion, etc. Hereafter, the different steps to compute the singular value decomposition are provided.

First, the word-document co-occurrence matrix, noted $\underline{\underline{A}}$, is computed. Next, a singular value decomposition (SVD) technique is applied to $\underline{\underline{A}}$. $\underline{\underline{A}}$ can be decomposed into the product of three other matrices:

$$\underline{\underline{A}} = \underline{\underline{U}}\underline{\underline{S}}\underline{\underline{V}}^T, \tag{3}$$

where $\underline{\underline{U}}$ and $\underline{\underline{V}}$ are the matrices of *left* and *right singular vectors* matrices and $\underline{\underline{S}}$ is the diagonal matrix of *singular values*, i.e. the non-negative square root of the eigenvalues of $\underline{\underline{A}}.\underline{\underline{A}}^T$. The first columns of $\underline{\underline{U}}$ and $\underline{\underline{V}}$ define the orthonormalized eigenvectors associated with the non-zero eigenvalues of $\underline{\underline{A}}.\underline{\underline{A}}^T$ and $\underline{\underline{A}}^T.\underline{\underline{A}}$ respectively. By choosing the $n$largest singular values the space can be furthermore reduced, eliminating some of the noise due to style or non-informative words:

$$\underline{\underline{A}}_n = \underline{\underline{U}}_n \underline{\underline{S}}_n \underline{\underline{V}}_n^T, \tag{4}$$

In this $n$-dimensional space the $i^{th}$ word $w_i$ is encoded:

$$\underline{x}_i = \underline{u}_i \underline{\underline{S}}_n / || \underline{u}_i \underline{\underline{S}}_n ||, \tag{5}$$

where $\underline{u}_i \ \underline{\underline{S}}_n$ is the $i^{th}$ row of the matrix $\underline{\underline{U}}_n \ \underline{\underline{S}}_n$ the SVD is an effective data cleaning process, by selecting the $p$ best factors, we reject negligible information contained in the data. Thus, it is possible to reconstruct an approximate version of original data from the selected factors and projection vectors.

### 4.4 Neural network training

Training is accomplished by calculating the derivative of the network's error with respect to each weight in the network when presented a particular input pattern. This derivative indicates which direction each weight should be adjusted to reduce the error. Each weight is modified by taking a small step in this direction. With a nonzero momentum factor, a fraction of the previous weight change is added to the new weight value. Notably, this accelerates learning in some cases. The patterns in the training set are traversed one-by-one. A pass through all the training patterns is called an epoch.

The training data are repetitively presented for multiple epochs, until a specified number of epochs has been reached. After each epoch, the error of the network applied to the validation set of patterns is calculated. If the current network scores the lowest error so far on the validation set, this network's weights are saved. At the conclusion of training, the network's best weights are used to calculate the network's error on the testing set.

## 5. Experiments

In order to measure the performance of our classifier, a collection of index terms from the training corpus is used. We used a specific corpus of Prophetic Traditions or "*Hadiths*' (sayings and doings of the Prophet 'Peace Be Upon Him') collected from the Prophetic Traditions Encyclopedia (*Alkutub Altissâa* – "The Nine Books") [14]. It is characterized by the specialization of its domain. It includes 453 documents distributed over 14 categories, as depicted in Tab. I. Our dataset has 5743 tokens from 14 major categories. The number of words collected from all the categories after the preprocessing step is 1065 word. These words are combined and sorted. Words of length less than 3 bytes are removed from the list. Some word ending characters are removed. Unique words are identified and arranged on the basis of their frequency of occurrences. The stop words, very high frequency words and very low frequency words are also removed. Tab. I represents the number of documents for each category for this corpus. Following the steps of the text classification model, we have removed the Arabic stop words and we have applied a light stemming process using Darwish Al-Stem Program[1]. We have used one half of the Arabic data set for testing the classifier and one half for training the text categorization classifier.

| Category name | # of training documents | |
|---|---|---|
| Faith | 23 | |
| *Qur'an* | 24 | |
| Knowledge | 22 | |
| Punishment | 22 | |
| *Al-Jihad* | 24 | |
| Good Manners | 31 | |
| Past Generations | 17 | |
| Biography | 16 | |
| Heritage | 24 | |
| Worships | 23 | |
| Behavior | 25 | |
| Food | 31 | |
| Clothes | 34 | |
| Personal States | 24 | |
| Total | 453 | |
| Number of words | Before preprocessing | After preprocessing |
| | 5743 | 1065 |

**Tab. I** *Number of documents per category for Hadith Corpus.*

## 5.1 Evaluation criteria

For text categorization task, we use standard evaluation criteria for comparing the methods and tasks with other works.

$$Precision : P = \frac{\#\text{ correct classes found}}{\#\text{ classes found}} \tag{6}$$

$$Recall : R = \frac{\#\text{ correct classes found}}{\#\text{ correct classes}} \tag{7}$$

Both quality measures are combined in the so-called F-measure:

[1] *http://www.glue.umd.edu/∼kareem/research/.*

$$F - measure : F = \frac{2 \cdot R \cdot P}{R + P}. \tag{8}$$

## 5.2 Preprocessing and input data

The goal of our experiments is to evaluate the performance of the proposed NN classifier on Arabic texts using the corpus described in Section 5.1.

After the stemming and stopping processes of the terms in each document, we represent them as the document-term frequency matrix ($Doc_j \times TF_{jk}$). $Doc_j$ which is referring to each document that exists in the corpus with $j = 1 \dots n$. Term Frequency $TF_{jk}$ is the number of how many times the distinct word $w_k$ occurs in document $Doc_j$, where $k = 1 \dots m$. The calculation of the terms weight $x_{jk}$ of each word $w_k$ is done by using a method that has been used by Salton [36] and is given by:

$$x_{jk} = TF_{jk} \times idf_k, \tag{9}$$

where the document frequency $df_k$ is the total number of documents in the corpus that contains the word $w_k$. The inverse document frequency $idf_k = log(n/df_k)$, where $n$ is the total number of documents in the corpus [40].

### 5.2.1 Input data to the neural network

After the preprocessing of the documents, a vocabulary that contains all the unique words in the corpus has been created. We have limited the number of unique words in the vocabulary to 1065 because the number of distinct words is large. Each of the words in the vocabulary represents one feature vector. Each feature vector contains the document-terms weight. The high dimensionality of feature vectors to be used as an input to the neural networks is not practical due to poor scalability and performance [40]. Therefore, the SVD has been used to reduce the original feature vectors $m = 1065$ into a small number of singular factor. In our case, we have selected the value of $d = 530$ since this parameter performs better for Arabic texts classification compared to other input parameters.

In order to have some idea of the data after the SVD transformation, we plot in Fig. 4 the first 200 factor vectors of a training dataset with 1065 features after transformation. Dimension reduction allows a data visualization of proteins in a 2-D plot, with two separated clusters (for example: C1 = *Worships* for the *Worships* category and C2 = *Behavior* for the *Worships* category, from 14 categories of our corpus). The x-coordinate is obtained by multiplying the first column of the matrix V (from SVD) by the reduced S matrix, with only the two first singular values. The y-coordinate is calculated by the multiplication of the second column of V by the reduced S matrix, with two SVD factors. The black circles are the vectors of the *Worships* category, while the light squares are the vectors of the *Behavior* category. From Fig. 4, we observe that the vectors corresponding to the categories can be distinguished relatively easily.

The loading factor graph for the accumulated proportion of eigenvalues is shown in Fig. 5. The value of $d$ contributes 77.14% of proportions from the original feature vectors.

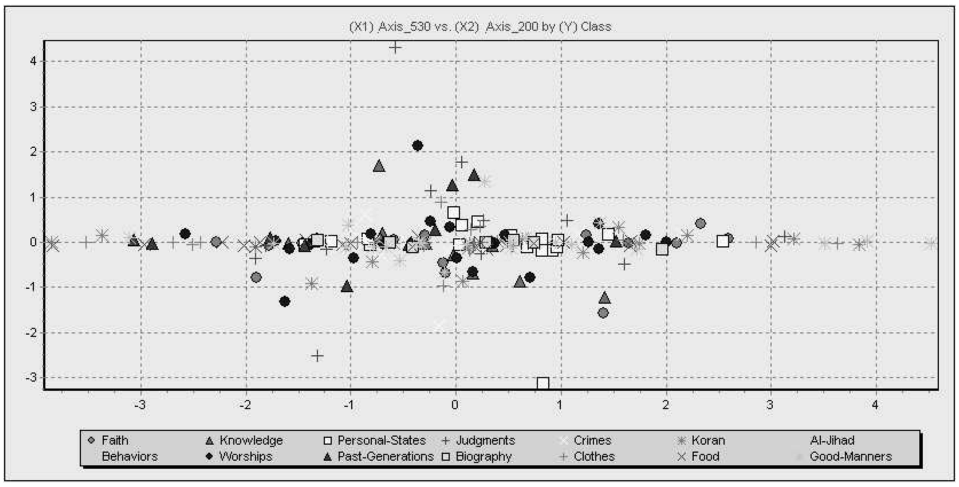


**Fig. 4** *Plotting of the first 200 factors of the 1065-features dataset.*

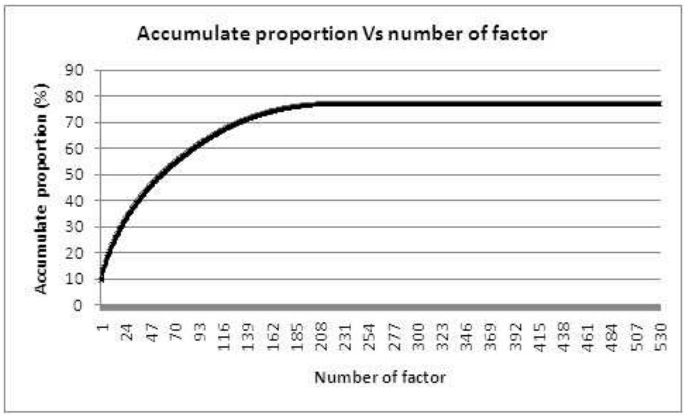


**Fig. 5** *Accumulated proportion of singular factor generated by the SVD.*

### 5.2.2 Characterization of the neural networks

The number of input layers $p$ is 1065 for the *Hadith* corpus where the number of the SVD dimensions is $d$=530. The number of hidden layers $q$ for the MLP NN is 30. The number of cluster $c$ for the RBF NN is 14. The trial-and-error approach has been used to find a suitable number of hidden layers that provide good classification accuracy based on the input data to the neural networks. The number of output layers $r$ is 14, which is based on the number of categories in the *Hadith* corpus. The MLP error back-propagation and the RBF neural networks Parameters are set as in Tab. II.

| | NN Type | |
|---|---|---|
| | **MLP** | **RBF** |
| **Architecture** | | |
| Use hidden layer | yes | yes |
| Neurons in the hidden layer | 30 | 14 |
| Cluster attribute | no | Class |
| **Learning parameters** | | |
| Learning rate (q) | 0.050 | 0.050 |
| Attribute transformation | none | none |
| Validation set proportion | 0.2 | 0.2 |
| **Stopping rule** | | |
| Number of iteration (I) | 1000 | 1000 |
| Error rate threshold | 0.001 | 0.001 |
| Verify error stagnation | no | no |

**Tab. II** *MLP and RBF neural networks parameters.*

## 5.3 Results and analyses

### 5.3.1 Learning error

In order to see how the accuracy improves with increasing number of iterations, Fig. 6 and Fig. 7 are plotted for the error rate during the learning process of the basic MLP and RBF NNs. Fig. 8 and Fig. 9 show the error rate for the SVD-Supported MLP and RBF NN. The mean square error (MSE) is measured for repeated training with increasing number of epochs.

The learning error decreases exponentially as the number of training epochs increases. We found that the error is extremely reduced after about 141 epochs for the MLP NN, and after about 14 epochs for the RBF NN, which might be considered as fast enough convergence for a practical system.

For the SVD-Supported NNs, we stop training after 179 epochs for the MLP NN, and after 159 epochs for the RBF NN. The reason why we use these values as stop condition is that, from our experiments, we find that these parameters can get the tradeoff between categorization accuracy and efficiency.

### 5.3.2 Classification results

In our experiments, we compare the performance by varying the number of dimension $k$ from 20 to 530. The NNs' input nodes number is equal to the dimension of the document vectors. For the SVD-Supported NN, the dimension ranges from 20 to 530, and for the basic NN with the vector space model, the number of vectors is 1065. The performance of our categorization system is evaluated by precision, recall and macro-averaging F-measure. The value of macro-averaging F-measure is based on the value of precision and recall.

SVD was tested for its ability in encoding important information from the high dimensional feature space into a reduced feature space of much lower dimension-

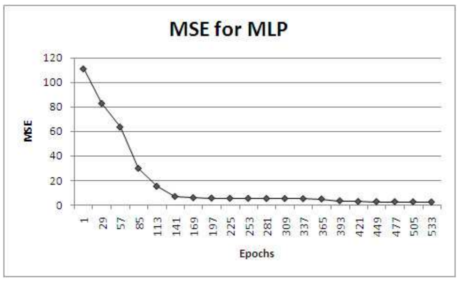


**Fig. 6** *MSE vs. Epochs for the Arabic text classification using basic MLP neural network.*

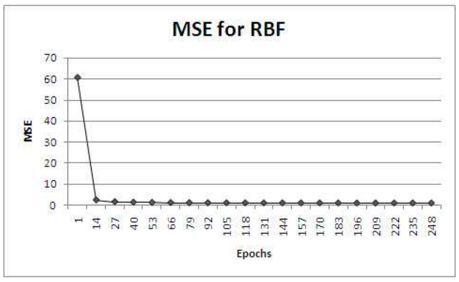


**Fig. 7** *MSE vs. Epochs for the Arabic text classification using basic RBF neural network.*

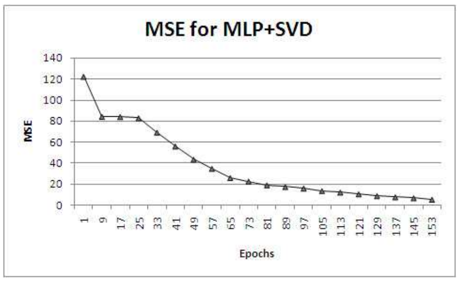


**Fig. 8** *MSE vs. Epochs for the Arabic text classification using SVD-Supported MLP NNs.*

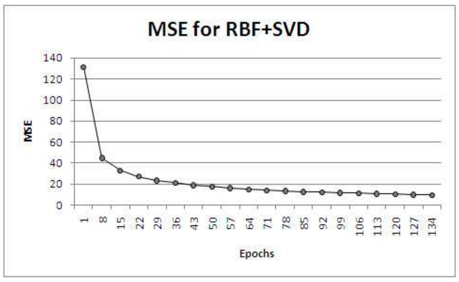


**Fig. 9** *MSE vs. Epochs for the Arabic text classification using SVD-Supported RBF NNs.*

ality. We used similar testing methodology and varied the reduced dimensionality. We plot the average recall, the average precision and the average F-measure respectively against the reduced dimensionality for the MLP and the RBF NNs.

Fig. 10 shows the comparison of the average recall for MLP and MLP using SVD, the dimension ranges from 20 to 530. The value of the average recall for MLP is 49%, however the average recall for MLP with SVD is in the range of 26%–53%. The best average recall values for MLP+SVD are 50%, 50%, 51%, 50% and 53% for the dimensions 200, 250, 300, 400 and 530, respectively.

Fig. 11 shows the comparison of the average recall for RBF and RBF using SVD. The dimension of SVD is in the range of 20 to 530. The average recall for RBF is 34% and the average recall for RBF with SVD was in the range of 12%–25%. The highest average recall value for SVD-Supported RBF is 28% for the dimensions 150 and 175.

Fig. 12 shows the comparison of the average precision for MLP and MLP using SVD. The value of the average recall for MLP is 52%. The average precision for MLP+SVD is in the range of 29%–55%. The best average precision values for MLP+SVD are 53% and 55% for the dimensions 200, 250 and 530, respectively. Fig. 13 shows the comparison of the average precision for RBF and RBF+SVD. The

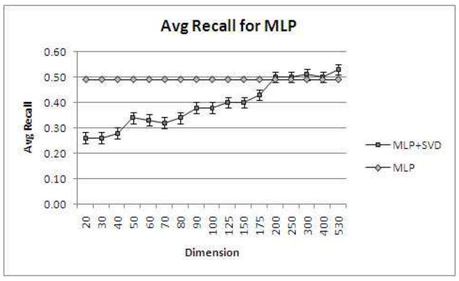


**Fig. 10** *Average Recall according to the number of dimension for MLP NN.*

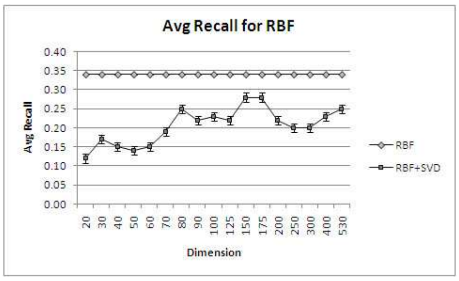


**Fig. 11** *Average Recall according to the number of dimension for RBF NN.*

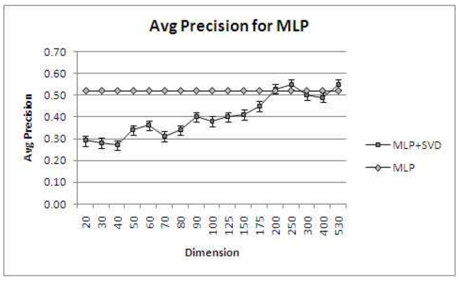


**Fig. 12** *Average Precision according to the number of dimension for MLP NN.*

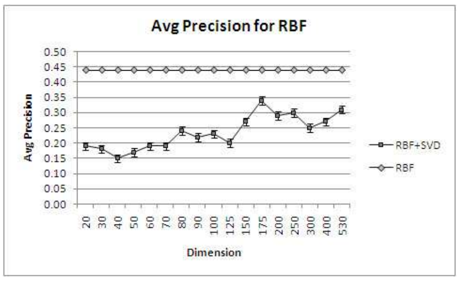


**Fig. 13** *Average Precision according to the number of dimension for RBF NN.*

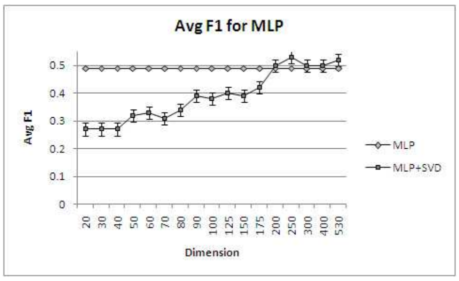


**Fig. 14** *Average F1 vs. the dimension for MLP NN.*

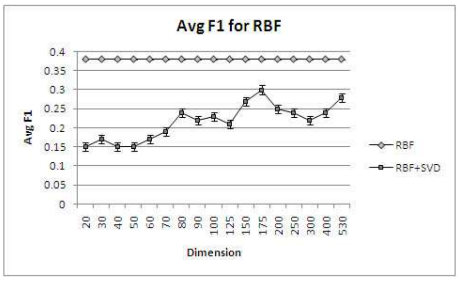
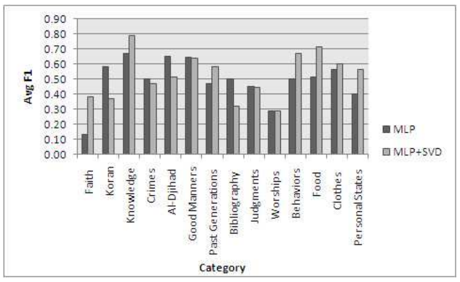


**Fig. 15** *Average F1 vs. the dimension for RBF NN.*

average precision for RBF is 44%, and the average precision values for RBF+SVD are between 19%–31%. The highest average precision value is 34% for the dimension 175.

Fig. 14 and Fig. 15 show the effect of number of input factors on the NN classification performance with a generation number of 530. In general, the average F1 measure value improved with higher number of input factors. The average F1



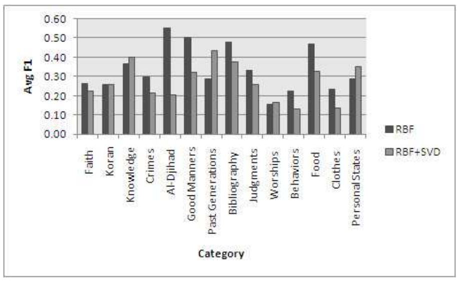


**Fig. 16** *Comparison of classification results for MLP NN model.*

**Fig. 17** *Comparison of classification results for RBF NN model.*

with all features (1065) was 49% for MLP and 38% for RBF. The average F1 with SVD (530) is between 22% and 53% for MLP, and between 15% and 30% for RBF.

Fig. 16 and Fig. 17 show the F1 measure for every category. As can be seen from Fig. 16, F1 measure reaches its highest values of 67% and 79% for the *Knowledge* (العلم) category for both MLP and MLP +SVD models, and the lowest value of 13% for the *Faith* (الإيمان) category for MLP and 29% for the *Worships* (العبادات) category for MLP+SVD. From Fig. 17 we can show that for the RBF, the highest value for the F1 measure is 55% for *Al-Jihad* (الجهاد) category, and the lowest value is 15% for the *Worships* (العبادات) category. For the RBF+SVD, the highest value is 43% for the *Past-Generations* (الأمم الماضية) category and the lowest value is 13% for the *Behaviors* (الأداب) category.

## 5.4 Discussion

From the previous experiments, we noticed that the neural network based Arabic text classifier was able to obtain reasonably average precision, recall and F1 measure values. The results show that neural networks trained by Back-propagation are effective in performing the text categorization task. For MLP neural network, we noticed that the overall recall, precision and F1 measure values for Arabic text classification increase and become more stable when we use SVD as features extraction technique. With the number of factors increasing, the accuracy rate increase gradually and at the dimension 530, MLP NN gets the best performance of 52%. For the RBF, the results show that the performance of classification is very unstable when the number of factors increases, the basic model of RBF has a better performance than RBF using SVD.

The results also showed that SVD was effective in reducing the dimensionality of the feature space from 1065 to 530, and, at the same, time maintaining average precision, recall and F1 measure values. This represented a reduction rate of 50%. Furthermore, we also find that fewer features selected by SVD can describe the characteristic of Arabic texts as well as the whole original features. It shows that adding many unimportant features does not necessarily enhance the performance of classification.

The reduced size of the vectors greatly decreases the computational (training) time in both neural network models. The computational time on SVD-Supported

MLP NN is lower than the computational time on basic MLP. For the number of 530 dimensions, the computational time of SVD-Supported MLP NN is 52.4 s. This is faster than the basic MLP NN model (191.7 s). The same remark can be made for RBF, since the computational time on SVD-Supported RBF NN is (984.5 s), compared to the basic RBF with computational time equal to (3157.9 s). Comparing the two NNs architectures, the MLP outperforms the RBF for both models (basic and with SVD). For the reasons above, it is estimated that the MLP NN is more appropriate than the RBF NN for Arabic texts categorization.

## 6. Conclusion

In this paper, we have developed Arabic text classification model using neural network and singular value decomposition (SVD) method. The learning techniques used are based on MLP or RBF NNs. It is shown that the introduction of SVD method has improved the categorization performance. As a tangible result, the reduced size of the vectors also decreased the computational time for both MLP and RBF neural networks. One of their advantages is that they require minimal computational and storage resources, which makes them ideal for Arabic texts classification. The experiments on Arabic "*Hadith*" corpus have demonstrated that the MLP NN model is effective in representing and classifying Arabic documents. This study concludes that MLP outperforms RBF for both models (basic and with SVD).

As a future work, other feature selection and reduction methods are worthwhile considering. It is also useful to use more data sets and more training documents in order to improve the learning capability. Finally, comparison with other learning algorithms employed for text categorization, such as SVM, KNN, Decision Trees and Bayesian Networks, might shed more light on the subject.

## References

[1] Al-Harbi S., Almuhareb A., Al-Thubaity A., Khorsheed M. S., Al-Rajeh A.: Automatic Arabic Text Classification, 9es Journées internationales d'Analyse statistique des Données Textuelles, JADT'08, France, 2008, pp. 77-83.

[2] Aljlayl M., Frieder O.: On Arabic Search: Improving the Retrieval Effectiveness Via a Light Stemming Approach. In: International Conference on Information and Knowledge Management, CIKM'02, ACM, McLean, VA, USA, 2002, pp. 340-347.

[3] Al-Shalabi R., Evens M.: A Computational Morphology System for Arabic. In: Workshop on Computational Approaches to Semitic Languages, COLING-ACL'98, August 1998.

[4] Bishop C. M.: Neural Networks for Pattern Recognition, Oxford University Press, Oxford, England, UK, 1995.

[5] Chathura R. D., Surendra R., Liyanage C. D.: Cloud Basis Function Neural Network: a Modified RBF Network Architecture for Holistic Facial Expression Recognition, Pattern Recognition, 41, 2008, pp. 1241-1253.

[6] Caudill M., Butler C.: Understanding Neural Networks: Computer Explorations, **1** and **2**, MIT Press, Cambridge MA, USA, 1992.

[7] Chen A., Gey F.: Building an Arabic Stemmer for Information Retrieval. In: Proceedings of the 11th Text Retrieval Conference, TREC'02, National Institute of Standards and Technology, 2002.

[8] Chen S., Cowan C., Grant P.: Orthogonal Least Squares Learning Algorithm for Radial Basis Function Networks, IEEE Transactions on Neural Networks, **2**, 2, 1991, pp. 302-309.

[9] Draper N., Smith H.: Applied Regression Analysis, 3rd Edition, John Wiley & Sons, New York, USA, 1998.

[10] Duch W., Jankowski N.: Survey of Neural Transfer Functions, Neural Computing Surveys, **2**, 1999, pp. 163-212.

[11] Duwairi R. M.: A Distance-based Classifier for Arabic Text Categorization. In: Proceedings of the International Conference on Data Mining, Las Vegas USA, 2005.

[12] El-Halees A.: Arabic Text Classification Using Maximum Entropy, The Islamic University Journal (Series of Natural Studies and Engineering), **15**, 1, 2007, pp. 157-167.

[13] El-Kourdi M., Bensaid A., Rachidi T.: Automatic Arabic Document Categorization Based on the Naïve Bayes Algorithm, 20th International Conference on Computational Linguistics, Geneva, August 2004.

[14] The Encyclopedia of the Nine Books for the Honorable Prophetic Traditions, Sakhr Company, 1997, http://www.Harf.com. Last access March 2, 2010.

[15] Finan R., Sapeluk A., Damper R.: Comparison of Multilayer and Radial Basis Function Neural Networks for Text-dependent Speaker Recognition, IEEE International Conference on Neural Networks (IJCNN'96), Washington DC, USA, 1996.

[16] Hagan M., Demuth H., Beale M.: Neural Network Design, PWS Publishing Company, Boston MA, USA, 1996.

[17] Hammo B., Abu-Salem H., Lytinen S., Evens M.: QARAB: A Question Answering System to Support the Arabic Language. Workshop on Computational Approaches to Semitic Languages. ACL 2002, Philadelphia, PA, July, 2002, pp. 55-65.

[18] Hastie T., Tibshirani R., Friedman J.: The Elements of Statistical Learning: Data Mining, Inference and Prediction. Springer, 2001.

[19] Haykin S.: Neural Networks: A Comprehensive Foundation, Prentice-Hall, Englewood Cliffs, NJ, USA, 2nd Edition, Chapter 4, MLP, pp. 156-252, Chapter 5 RBF, 1999, pp. 256-312.

[20] Hitti P.: History of the Arabs. $8^{\text{th}}$ edition. London: Macmillan, 1963.

[21] Husbands P., Simon H., Ding C.: On the Use of the Singular Value Decomposition for Text Retrieval. In: Proceedings of 1st SIAM Computational Information Retrieval Workshop, 2000.

[22] Jianping D., Sundararajan N., Saratchandran P.: Communication Channel Equalization Using Complex-Valued Minimal Radial Basis Function Neural Networks, IEEE Transactions on Neural Networks, **13**, 3, 2002, pp. 687-696.

[23] Kaminski W., Strumillo P.: Kernel Orthonormalization in Radial Basis Function Neural Networks, IEEE Transactions on Neural Networks, **8**, 5, 1997, pp. 1177-1183.

[24] Khoja S.: Stemming Arabic Text, Lancaster, UK, Computing Department, Lancaster University, 1999.

[25] Lam S. L. Y., Lee D. L.: Feature Reduction for Neural Network Based Text Categorization. In: Sixth International Conference on Database Systems for Advanced Applications (DASFAA'99), 1999, p. 195.

[26] Larkey L., Ballesteros L., Connell M. E.: Improving Stemming for Arabic Information Retrieval: Light Stemming and Co-occurrence Analysis, Proceedings of SIGIR'02, 2002, pp. 275-282.

[27] Liu T., Liu S., Chen Z., Wei-Ying M. A.: An Evaluation on Feature Selection for Text Clustering. In: Proceedings of the 12th International Conference ICML'03, Washington, DC, USA, 2003, pp. 488-495.

[28] Mesleh A. A.: Chi Square Feature Extraction Based SVMs Arabic Language Text Categorization System, Journal of Computer Science, **3**, 6, 2007, pp. 430-435.

[29] Ng H. T., Goh W. B., Low K. L.: Feature Selection, Perception Learning, and a Usability Case Study for Text Categorization. In: Proceedings of the 20th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval, 1997, pp. 67-73.

[30] Papaioannou I. V., Roussaki I. G., Anagnostou M. E.: Comparing the Performance of MLP and RBF Neural Networks Employed by Negotiating Intelligent Agents, Proceedings of the IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT'06), 2006.

[31] Park J., Harley R., Venayagamoorthy G.: Comparison of MLP and RBF Neural Networks using Deviation Signals for On-Line Identification of a Synchronous Generator, IEEE Power Engineering Society Winter Meeting, New York, USA, 2002.

[32] Rajan K., Ramalingam V., Ganesan M., Palanivel S., Palaniappan B.: Automatic Classification of Tamil Documents Using Vector Space Model and Artificial Neural Network. Expert Systems with Applications, 2009.

[33] Rakotomalala R., Mhamdi F.: Combining Feature Selection and Feature Reduction for Protein Classification. In: Proceedings of the 6th WSEAS International Conference on Simulation, Modelling and Optimization, Lisbon, Portugal, September 2006, pp. 444-451.

[34] Rogati M., Yang Y.: High-Performing Feature Selection for Text classification, CIKM'02, ACM, 2002.

[35] Rumelhart D., Hinton G., Williams R.: Learning Internal Representations by Error Propagation, Parallel Distributed Processing, MIT Press, Cambridge MA, USA 1986.

[36] Salton G., Buckley C.: Term-weighting Approaches in Automatic Text Retrieval, Information Processing and Management, **24**, 5, 1988, pp. 513-523.

[37] Sauban M., Pfahringer B.: Text Categorization Using Document Profiling. Principles of Data Mining and Knowledge Discovery, 2003.

[38] Sawaf H., Zaplo J., Ney H.: Statistical Classification Methods for Arabic News Articles, Workshop on Arabic Natural Language Processing, ACL'01, Toulouse, France, July 2001.

[39] Sebastiani F.: Machine Learning in Automated Text Categorization, ACM Computing Surveys, 2002, **34**, 1, pp. 1-47.

[40] Selamat A., Omatu S.: Neural Networks for Web Page Classification Based on Augmented PCA. In: Proceedings of the International Joint Conference on Neural Networks, 2003, pp. 1792-1797.

[41] Syiam M. M., Fayed Z. T., Habib M. B.: An Intelligent System for Arabic Text Categorization, IJICIS, **6**, 1, 2006, pp. 1-19.

[42] Wall M., Rechtsteiner A., Rocha L.: A Practical Approach to Microarray Data Analysis, chap. Singular Value Decomposition and Principal Component Analysis. Kluwer, 2003, pp. 91-109.

[43] Wasserman P. D.: Advanced Methods in Neural Computing, Van Nostrand Reinhold, New York, NY, USA, 1995.

[44] Wermeter S.: Neural Network Agents for Learning Semantic Text Classification, Information Retrieval, **3**, 2, 2000, pp. 87-103.

[45] Yang Y., Pedersen J. O.: A Comparative Study on Feature Selection in Text Categorization. In: Proceedings of ICML'97, 1997, pp. 412-420.

[46] Yang Y., Liu X.: A Re-examination of Text Categorization Methods. In: Proceedings of 22nd ACM International Conference on Research and Development in Information Retrieval, SIGIR'99, ACM Press, New York, USA, 1999, pp. 42-49.

[47] Yang Y., Slattery S., Ghani R.: A Study of Approaches to Hypertext Categorization, Journal of Intelligent Information Systems, 2002.

[48] Yu B., Zong-ben X., Cheng-hua L.: Latent Semantic Analysis for Text Categorization Using Neural Network, Knowledge-Based Systems Journal, 21, 2008, pp. 900-904.