# Ham- Spam Filtering Using Kernel PCA

Issam Dagher, Rima Antoun

***Abstract*—** Electronic mails have become one of the most important ways of communication. Email filtering is a very important task. The objective of this paper is to study the Kernel Principal Component Analysis classifier implemented for email filtering process (Ham vs. spam emails). Different experiments were done using a public corpus extracted from the University of California-Irvine Machine Learning Repository. Different training and test sets were used. A comparison with PCA, Support Vector Machine and Bayes detector was done to prove its superior behavior.

***Keywords*—** Ham, Spam, Kernel, PCA, SVM, Bayes.

## I. Introduction

Document classification approach consists of assigning a document to one of predefined categories based on its content. This methodology is implemented in several applications such as email filtering.

Nowadays, Electronic mails have become one of the most important ways of communication. Unfortunately, as the importance of messaging increases, the number of spam messages sent to users also increases. In fact, spam emails are identical messages sent to many users. Spam emails have different functions. Some of them serve for advertising issues, others are responsible of spreading computer viruses and there exist spam messages intended to steal the user financial identities. In addition to their direct disadvantages, spam messages yields to waste the network bandwidth [1]. As the spam messages are getting to be more severe, classifying messages and filtering the spam ones has become an essential need to protect users from their risks.

Different Classification methods exist already. Principal Component Analysis [2], Support Vector Machine (SVM) and Bayes detector are the most famous ones.

The aim of this research is to increase the accuracy of the PCA Classifier by changing the way of selecting representative features. Principal component analysis (PCA) consists of finding the principal components (PCs) of each class. The PCs of a class form what is called PCA basis. When a new message is received, it will be projected into the new computed basis. Detection of the document class depends on the reconstruction error. This is known as Document Reconstruction (DR) Process.

Principal Component Analysis [4] idea was launched in 1901 by the mathematician Karl Pearson in his famous paper "On lines and planes of closest fit to systems of points in space". Pearson developed the idea of representing a system of points in a high dimensioned plane by a lower best fitting one [2]. Later, in 1933, Harold Hotelling introduces PCA as a way to" understand the structure of large numbers of correlated multivariate observations" and a method "to measure the strength of relationship between two dependent sets of multivariate observations". Jolliffe discussed the statistical properties of PCs based on covariance matrix analysis [2].

Before applying PCA, a set of preprocessing steps must be done. Vocabulary pruning [8] is one of them. It consists of removing non discriminative terms and extracting the relevant ones. In [3], the vocabulary terms are selected using Matlab. Selecting the features using mutual information technique was done in [11].

In this paper, four scenarios were implemented. Scenario 1 consists of different representative features for each class. The features of scenario 2 are result of pruning the overall samples together. Scenario 3 consists of selecting the common terms between classes. Scenario 4 is an updated version of scenario 3 where characteristic terms are added to the set of common features.

These scenarios were implemented using a public corpus extracted from the Machine Learning Repository of the University of California-Irvine. The corpus contains Ham and Spam samples. These samples are used for training and testing sets.

At the end of the study, a comparison is made between PCA, SVM and Bayes detector. This comparison proves the very good performance of the PCA classifier.

The outline of the paper is as follows: Section 2 introduces the Principal Component Analysis and the document reconstruction approaches. In Section 3, general overviews of the Support Vector machine and Bayes detector are given. Section 4 consists of a detailed description of PCA classifier including pruning [8], TF-IDF [9] and SVD [12] techniques. Section 5 discusses the different scenarios and their results in addition to the comparison of PCA with SVM and Bayes detector [13]. Section 6 presents the conclusion deduced after comparing the whole scenarios.

.

## II. PCA

### *A. PCA Architecture*

Principal Component Analysis is a feature representing

Issam Dagher is an associate professor at the University of Balamand. Department of Computer Engineering. dagheri@balamand.edu.lb

Rima Antoun is a graduate student at the University of Balamand.

method with dimensionality reduction [10] effect that converts a set of observations data to uncorrelated vectors called Principal Components (PCs). These PCs form the orthogonal axes of a new reduced space that "optimally describes the highest variance of the data" [5]. In document classification, Principal Component Analysis is used for class selection. It consists of extracting its PCs.

Let matrix **X** be the TF-IDF matrix [14] of a class. The TF-IDF matrix is also known as the representative model [15] of this class. Matrix **X** is a dxn matrix where each of the n columns is a d-dimensional vector model representing a document in the class set. The Principal Components of this class are the eigenvectors of the Covariance Matrix (Co) of **X** given by:

$$Co = \frac{1}{n} MM^T$$

Where

$$M = X - \mu$$

**μ** is the mean message vector defined by:

$$\mu = \frac{1}{n} \sum_{i=1}^{n} X_i$$

The mean message vector should be subtracted from each column of X.

In the figure below, the graph to the left represent a set of observations as a function of Feature 1 and Feature 2. The figure to the right represents the extraction of the principal components: the orthogonal basis of the new reduced space.

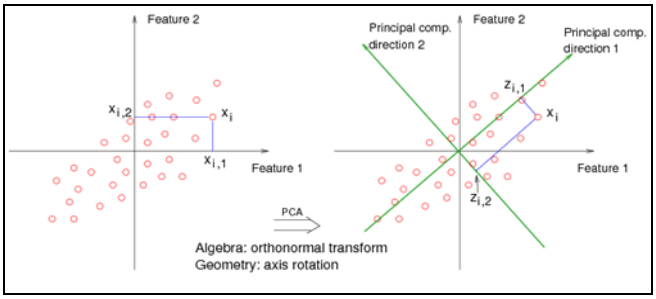


Figure 1: Extraction of PCs of a Set of Observations

Different applications are based on PCA. Some examples are document classification, email filtering, fingerprints enhancement and Face Recognition.

In this paper we have used the Singular Value Decomposition (SVD) method in order to compute the eigenvectors of the covariance matrix.

B. Document Reconstruction

Document Reconstruction is used for document classification. It consists of projecting an incoming document in the PCA basis of each category. The next step is to attempt to recuperate the original message by reconstructing the resulted vector using the same PCs. Then reconstruction errors consisting of the difference between reconstructed messages and original ones are computed and the testing document is assigned to the class with the least error.

Let **W** be the projection matrix of a class found using PCA. **W** matrix is used to project data of the original space into PCA basis.

Thus, the projection of the document **z** in the set of PCs is defined as:

$$p = W^T z$$

The reconstruction of the original document is the projection of the column vector **p** in the initial high dimensional space. It is given by:

$$z' = Wp = WW^T z$$

The reconstruction error is found based on the Euclidean distance. It is equal to the absolute value of the difference between the original document vector **z** and the recovered document **z'.**

$$L_p = | z - z' |^2$$

C. *SVD*

After preparing data and computing the representative matrix for each class, PCA method is applied on each of the two matrices in order to find the PCA basis of each class. However, finding PCs by computing the eigenvalues and eigenvectors of the covariance matrix is a complicated time-consuming method. Thus, a restrictive and much simpler technique is used instead: The Singular Value Decomposition (SVD).Singular Value Decomposition is a matrix factorization method. SVD theorem states that an (mxn) matrix **M** can be seen as the product of three matrices: an orthogonal (mxm) matrix **U**, an (mxn) diagonal matrix **S** and a transpose of another orthogonal (nxn) matrix **V.** So matrix **M** can be expressed as:

$$M = USV^T$$

The columns of **U** matrix are called the left singular vectors and correspond to the eigenvectors of $MM^T$ matrix. Concerning the **V** matrix, its columns are the eigenvectors of the $M^T M$ matrix. These columns are the right singular vectors. The next step consists of finding the projection matrix of each class. Since the projection matrix is composed of the first k PCs, the next mission consists of finding the value of k. In fact, choosing the number of PCs is a subjective task. The method used in this study takes into consideration the cumulative sum of singular values. Let S be the diagonal matrix resultant from applying SVD on matrix M. the purpose is to find the number of singular values which the ratio of their cumulative over the sum of all of them is greater than a threshold value. The threshold value is chosen to be 0.8.

## III. KERNEL PCA

### A. Derivation

It extends the conventional principal component analysis (PCA) to a high dimensional feature space using the kernel trick. It extracts nonlinear principal components without expensive computations. This is done by mapping every point x to some nonlinear feature space $\phi(x)$.

The mean and the covariance of the data in the feature space are given by

$$\mu = \frac{1}{n}\sum_{i=1}^{n}\phi(x_i)$$

$$C = \frac{1}{n}\sum_{i=1}^{n}\phi(x_i)\phi(x_i)^T$$

Given that the Eigenvector equation is $Cv = \lambda v$

$$Cv = \frac{1}{n}\sum_{i=1}^{n}\phi(x_i)\phi(x_i)^T v = \lambda v$$

$$\Rightarrow v = \frac{1}{n\lambda}\sum_{i=1}^{n}\phi(x_i)\phi(x_i)^T v = \frac{1}{n\lambda}\sum_{i=1}^{n}(\phi(x_i)v)\phi(x_i)^T$$

$$v = \sum_{j=1}^{n}\alpha_j\phi(x_j)$$

$$\Rightarrow Cv = \frac{1}{n}\sum_{i=1}^{n}\phi(x_i)\phi(x_i)^T\sum_{j=1}^{n}\alpha_j\phi(x_j) = \lambda\sum_{i=1}^{n}\alpha_i\phi(x_i) \Rightarrow$$

$$\frac{1}{n}\sum_{i=1}^{n}\phi(x_i)\sum_{j=1}^{n}\alpha_j K(x_i,x_j) = \lambda\sum_{i=1}^{n}\alpha_i\phi(x_i)$$

$$\Rightarrow K\alpha = n\lambda\alpha \Rightarrow \lambda n\alpha^T\alpha = 1$$

For a point x, its projection onto the principal component is :

$$\phi^{\mathrm{T}}(x).v_j = \sum_{i=1}^{n}\alpha_{ji}.K(x,x_i)$$

In general $\varphi(x_i)$ may not be zero mean. Centerd features :

$$\phi(x_k) = \phi(x_i) - \frac{1}{\mathrm{n}}\sum_{k=1}^{n}\phi(x_k) \Rightarrow)$$

$$K = K - 2I_{\frac{1}{n}}.K + I_{\frac{1}{n}}.K.I_{\frac{1}{n}}$$

where $I_{\frac{1}{n}}$ is a matrix with all elements are $\frac{1}{\mathrm{n}}$

### B. Summary of the Kernel PCA

1. Pick a kernel (Gaussian kerenel)

$$K(X1, X2) = \exp(-\frac{(X_1 - X_2)^2}{2\sigma^2})$$

2. Construct the normalized kernel matrix

$$K = K - 2I_{\frac{1}{n}}.K + I_{\frac{1}{n}}.K.I_{\frac{1}{n}}$$

3. Solve an eigenvalue problem :

$$K\alpha = \lambda\alpha$$

4. For any data point we can represent it by

$$y_j = \sum_{i=1}^{n}\alpha_{ji}K(x,x_i) \quad j = 1,\cdots d$$

### C. Kernel PCA Reconstruction

The PCA reconstruction error is given by

$$L_P = |\, z - z^{'}\,|^2 = |\, z - WW^T\,|^2$$
$$= |\, z\,|^2 - 2z^TWW^Tz + z^TWW^TWW^Tz$$
$$= |\, z\,|^2 - z^TWW^Tz = |\, z\,|^2 - |\, W^Tz\,|^2$$

The Kernel PCA reconstruction error is given by:

$$L_{KP} = |\, \phi(z) - WW^T\phi(z)\,|^2$$
$$= |\, \phi(z)\,|^2 - |\, W^T\phi(z)\,|^2$$

Let V be the eigenvectors of K and S be a diagonalized matrix containing the eigenvalues of K. The projection of z on the principal space is given by $S^{-1}V^Tk$ ($k$ is a column vector with coordinates $k_i = k(x,x_i)$). The Kernel PCA reconstruction error can be computed as follows:

$$L_{KP} = k(x,x) - |\, S^{-1}V^Tk\,|^2$$

## IV. SVM AND BAYES

### A. Support Vector Machine

Support Vector machine Classifier (SVM) [6] is another technique that can be used for email filtering. As the PCA, the role of SVM [7] is to discriminate between the training samples of ham and spam classes. Then, it classifies the incoming messages by assigning it to one of the classes. The preprocessing setup is the same as the one of PCA classifier. Vocabulary Pruning and TF-IDF matrix computing are essential. Training samples of both classes must be mapped into the same space which implies having the same vocabulary

set. The training samples of each class must be denoted by either "1" or "0". Then, the next step is to find an optimal hyperplane. The optimal hyperplane is the hyperplane with the largest margin between the 2 classes [8]. This hyperplane could be linear or nonlinear. The type of the hyperplane depends on data distribution. After mapping data in the required space and finding the optimal hyperplane, Classification of testing samples can be done. When a new message arrives, it will be mapped in the same space and the assignment of this message into a class will be based on which side of the gap the message fall on.

B. Bayes Detector

Bayes Detector [13] is a classification tool used for document classification. It is a probabilistic classifier on Bayes theorem. Given that a document is denoted by D and the classes are denoted by A and B, Bayes classifier assign document D to class A if and only if p(A/D) > p(B/D). Otherwise, document D is assigned to class B.

The first step of Bayes classification consists of extracting Ham and Spam datasets. Ham dataset contains a huge number of ham characteristic words whereas spam datasets are formed of spam class characteristics. Next, the features probabilities are calculated. The words probabilities are given by:

$$p(f_i) = p(f_i \setminus S).P(S) + p(f_i \setminus H)P(H)$$

$p(f_i|S)$=frequency of spam training samples containing the ith feature.

$p(f_i|H)$=frequency of Ham training samples containing the ith featuure.

p(S)=probability of Spam message.

P(H)=probability of Ham message.

Once a new message M arrives, Bayes classifier extracts the spam and ham characteristic terms and computes the probability of this message being ham or spam. These probabilities are found using Bayes theorem:

$$P(H \setminus f_1 f_2 \cdots f_i) = \frac{P(f_1 f_2 \cdots f_i \setminus H)P(H)}{P(f_1 f_2 \cdots f_i)}$$

$$P(S \setminus f_1 f_2 \cdots f_i) = \frac{P(f_1 f_2 \cdots f_i \setminus S)P(S)}{P(f_1 f_2 \cdots f_i)}$$

## V. PCA CLASSIFIER

A. Training Set

The training set consists of a collection samples used as a reference for testing process. For example, in email classification the training sets are predefined ham and spam messages. These training sets undergo a preprocessing procedure before applying PCA method.

Vocabulary Pruning is the first step in preprocessing data in order to make it ready for classification. It is one of the core methodologies of dimensionality reduction. It consists of removing some vocabulary terms from a data collection before finding its model representation. Vocabulary pruning is applied in order to remove the least discriminative words in the analysis.

In document classification, Documents are represented as a function of the vocabulary terms. Accordingly, the model representation is a dxn matrix where d corresponds to the number of vocabulary words and n to the number of documents.

In this research, model representation matrices are computed for ham and spam classes by applying TF-IDF method on the training set of each class. Thus, each document in the training set is represented by a column vector with dimension equal to the number of words. Therefore, by pruning vocabulary terms, the dimension will be reduced.

Vocabulary pruning consists of removing two types of terms: high frequency terms and Singletons. High Frequency terms are words that occur frequently in a set. They are words with frequency greater than a threshold value. On the other hand, Singletons are words that occur only once in a set. High frequency terms and singletons are non-consistent words for text categorization.

In fact, high consistent words are called characteristic features. Characteristic features of a class refer to discriminative terms for this class. Most probably, they are high frequency terms. However, they are not removed. For example in the spam training set, the term "call" occurs 419 times and the term "free' occurs 260 times. These words are high frequency terms. However, they are not removed because they characterize the spam class.

Furthermore, pronouns, prepositions and stop-words are eliminated. These terms cannot be considered as characteristic terms of any class and thus should be removed from the sets.

Once the set of vocabularies is found, the TF-IDF matrix is computed. The TF-IDF matrix also referred to as model representation matrix. It is a powerful mathematical representation of a collection of documents. It is called "representation matrix" since it serves to represent the collection as a set of observations in a space. The axes of the spaces correspond to the words, and the documents are represented as points in this system.

TF-IDF value is the product of the Term Frequency (TF) and the inverse document frequency (IDF). It represents a composite weight for each term in each document. TF-IDF is given by:

$$tfidf\ (i, j) = tf\ (i, j).idf\ (i)$$

The term frequency values offer information about the number of occurrences of vocabulary terms in each documents of a collection. It is a "powerful representation of the overall data collection and its overall distribution of words over documents" [5]. The TF of a word i in document j is the number of occurrence of this word in that document

$$tf\ (i, j) = \#\text{ of occurence of word i in document j}$$

The inverse document frequency is given by:

$$idf(i) = \log(\frac{D}{d})$$

Where D is the total number of documents in the corpus and d is the number of documents in which the $i^{th}$ word occurs.

IDF is used for reducing the negative effects caused by function words. Function words occur in almost every document in a collection. Thus, they should not be considered as indicators of document class. In practice, the more documents a vocabulary term occurs in, the less will be the IDF value of that word and thus the TF_IDF weight of that word in each document.

In this paper, after finding the set of vocabulary using pruning technique, and thus the axes of the representation, TF-IDF matrix is computed. Having m training ham messages and n training spam messages, the TF-IDF matrix will be a dx(m+n) matrix. The first m columns of this matrix correspond to the vector models representing the documents of the Ham class and the last n ones will be the representative vectors of the documents of the Spam Class. The extended matrix will be then splitted into two: a (dxm) H matrix representative model of the Ham class and a (dxn) S matrix representative model of the Spam class. The next step consists of applying PCA technique.

B. Testing Set

After Finding the PCA basis and projection matrices of each class, incoming messages are tested using Document Reconstruction [16]. The objective of document reconstruction is to assign the new message to the correct [17].The document reconstruction procedure [18,19] consists of 5 steps:

1. Finding the vector model **z** of the incoming message.
2. Projecting **z** into Ham basis and Spam

$$PCA: \quad p_H = W_H^T z$$

$$p_S = W_S^T z$$

$$KernelPCA: \; y_H = \sum_{i=1}^{n} \alpha_{ji} K(z, z_{iH})$$

$$y_S = \sum_{i=1}^{n} \alpha_{ji} K(z, z_{iS})$$

3. Computing the error for both reconstructions

$$PCA:$$

$$L_{PH} = |z - z_H|^2 = |z|^2 - |W_H{}^T z|^2$$

$$L_{PS} = |z - z_S|^2 = |z|^2 - |W_S{}^T z|^2$$

$$KernelPCA:$$

$$L_{KPH} = |z - y_H|^2 = k(z,z) - |S_H{}^{-1} V_H{}^T k_H|^2$$

$$L_{kPS} = |z - z_S|^2 = k(z,z) - |S_S{}^{-1} V_S{}^T k_S|^2$$

4. PCA: Compare $L_{pH}$ and $L_{pS}$

Kernel PCA: Compare $L_{kpH}$ and $L_{kpS}$

And assign the tested message to the class with the smallest reconstruction error.

C. Data preprocessing

Spam and ham samples are seen as one collection. Pruning technique is applied on the whole collection and a set of vocabulary terms is extracted. Characteristic words are kept as well. Next, the TF-IDF matrix of the whole collection is found then splitted into two to extract the representation matrices for each class.

1. Find the pruned vocabulary terms of the collection
2. Compute the TF-IDF matrix with respect to the set of vocabularies: T (dx(m+n))
3. Split T matrix into 2: H (dxm) matrix that consists of the first m columns of T. S (dxn) matrix that consists of the remaining n columns.
4. Apply PCA and kernel PCA on each of the H and S matrices.

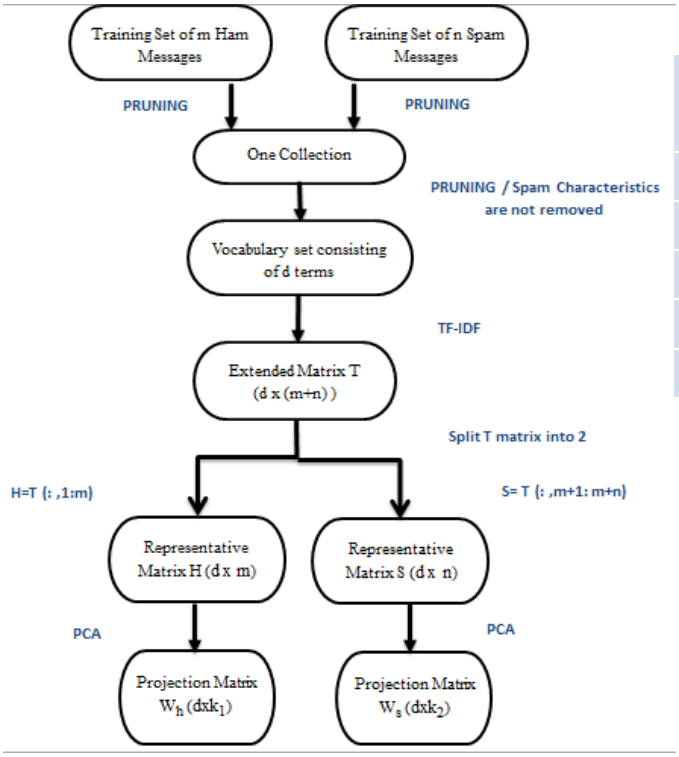


Figure 2: PCA Diagram

## VI. EXPERIMENTS AND RESULTS

### A. Corpus

The source of the email corpus used is the University of California-Irvine Machine Learning Repository (https://archive.ics.uci.edu/ml/datasets/SMS+Spam+Collection ). It consists of 4827 Ham messages and 747 Spam messages. Each Ham message starts with the term "ham" and Spam message with "spam" term. For example:

- ham What you doing?how are you?
- spam FreeMsg: Txt: CALL to No: 86888 & claim your reward of 3 hours talk time to use from your phone now! ubscribe6GBP/ mnth inc 3hrs 16 stop?txtStop

Different number of training and testing messages is chosen (Table 1). We have used Matlab on an Intel Pentium CPU 3.4GHz.

Table 1: Distribution of Ham and Spam sample for Each Trial

| | **Training Set** | | **Testing Set** | |
|---|---|---|---|---|
| | Ham | Spam | Ham | Spam |
| **Trial 1** | 100 | 100 | 50 | 50 |
| **Trial 2** | 200 | 200 | 100 | 100 |
| **Trial 3** | 570 | 218 | 232 | 57 |
| **Trial 4** | 1128 | 197 | 451 | 68 |
| **Trial 5** | 4355 | 667 | 476 | 82 |

### B. Comparison of PCA with SVM and Bayes Classifiers

Support Vector Machine [20] and Bayes classifier were implemented and compared to the Scenario 2 PCA Classifier. The results are shown in the following tables.

Table 2: Results of KPCA, PCA, SVM and Bayes Detector for Trial 1

| **Method** | Time | Features | Accuracy |
|---|---|---|---|
| **KPCA** | 4 sec | 486 | 98% |
| **PCA** | 3 sec | 486 | 96% |
| **SVM** | 6 sec | 486 | 92% |
| **Bayes** | 5 sec | 486 | 94% |

Table 3: Results of KPCA, PCA, SVM and Bayes Detector for Trial 2

| **Method** | Time | Features | Accuracy |
|---|---|---|---|
| **KPCA** | 6sec | 821 | 96% |
| **PCA** | 5 sec | 821 | 94.5 |
| **SVM** | 18 sec | 821 | 91% |
| **Bayes** | 7 sec | 821 | 92% |

Table 4: Results of KPCA, PCA, SVM and Bayes Detector for Trial 3

| **Method** | Time | Features | Accuracy |
|---|---|---|---|
| **KPCA** | 18sec | 1236 | 99.11% |
| **PCA** | 16 sec | 1236 | 98.27% |
| **SVM** | 25 sec | 1236 | 96.44% |
| **Bayes** | 20 sec | 1236 | 96.89% |

Table 5: Results of KPCA, PCA, SVM and Bayes Detector for Trial 4

| **Method** | Time | Features | Accuracy |
|---|---|---|---|
| **KPCA** | 38sec | 1603 | 99.23% |
| **PCA** | 36 sec | 1603 | 98.69% |
| **SVM** | 42 sec | 1603 | 96.67% |
| **Bayes** | 40sec | 1603 | 97.5% |

Table 6: Results of KPCA, PCA, SVM and Bayes Detector for Trial 5

| **Method** | Time | Features | Accuracy |
|---|---|---|---|
| **KPCA** | 123sec | 3981 | 98.89% |
| **PCA** | 120 sec | 3981 | 98.03% |
| **SVM** | 155 sec | 3981 | 96.21% |
| **Bayes** | 250 sec | 3981 | 96.95% |

.

## VII. CONCLUSION

Email filtering task depends on document classification approach. When classifying documents, choosing the best performing classifier is an elementary step. Thus extracting the best characterizing features, and correctly classifying incoming messages are key issues. The performance of the system is measured in terms of its accuracy and its consumed time.

Comparing with the PCA, support vector machine and Bayes detector, kernel PCA had the best performance regarding the accuracy .The number of ham and spam tests classified correctly was the highest for all trials and the time required was very comparable to the PCA. The accuracy of the Bayes detector was high but the main drawback of this method was the time needed to achieve the classification especially for large number of features.

## REFERENCES


[1] [1] Gomez, J. C., & Moens, M. F. (2011). *PCA document reconstruction for email classification.* Belgium: Kuleuven University.

[2] [2] Jolliffe, I. T. (2002). *Principal component analysis*. (2nd ed.). New York: Springer. *3*, 29-61.

[3] [3] Banchs, R. (2012). *Text mining with MATLAB.* New York: Springer.

[4] [4] Anderson, T.W. (2003). *An introduction to multivariate statistical analysis*. (3rd ed.). New Jersey: Wiley.

[5] [5] Haykin, S. (1999). *Neural networks: a comprehensive foundation*. (2nd ed.). Prentice Hall.

[6] [6] Drucker, H., Wu, D., & Vapnik, V.N. (1999*). Support vector machine for spam categorization.* IEEE Transactions on Neural Networks 10 (5), 1048-1054.

[7] [7] Kim, H., Howland, P., & Park, H. (2005). *Dimension reduction in text classification with support vector machines.* Journal of Machine Learning Research 6, 37-53.

[8] [8] Madsen, R.E., Sigurdsson S., Hansen, L.K., & Larsen J. (2004). *Pruning the vocabulary for better context recognition.* IJCNN 2004, 2(1439-1444).

[9] [9] Roelleke T, Wang J. (2008) *TF-IDF uncovered: a study of theories and probabilities.* In Proceedings of the 31st annual international ACM SIGIR conference, 435-442.

[10] [10] Turney PD, Pantel P (2010) *From frequency to meaning:vector space models of semantics.* J Artif Intell Res 37(1):141-188

[11] [11] Pninski L (2003) *Estimation of entropy and mutual information.* Neural Computation 15:1191-1253.

[12] [12] Li CH, Park SC (2007) *Neural Network for text classification based on SVD*. In proceddingsof the 7th international conference on computer and information technology,47-52.

[13] [13] Rish I (2001) *an empirical study of the naïve Bayes Classifier*. In Proceedings of the IJCAI 2001 workshop on empirical methods in artificial intelligence.

[14] [14] Sebastiani F (2002) *machine learning in automated text categorization*. ACM computer Surv 34(1):1-47.

[15] [15] Baeza-Yates R, Ribeiro-Neto B (2011) *Modern information retrieval: the concepts and technology behind search*, 2nd edition, Assisson-Wesley Professional, Boston.

[16] [16] Abu-Nimeh S, Nappa D, Wang X, Nair S (2007). *A comparison of machine learning techniques for phising detection.* In proceedings of the Anti-Phising working groups 2nd annual eCrime Researchers Summit: ECrime 2007. ACM ,NY,60-69.

[17] [17] Barman P, Iqbal N, Lee S (2006) . *Non-negative matrix factorization based text mining: feature extraction and classification*. In Proceedings of the 13th International conference ICONIP 2006. Springer-Verlag Berlin, 703-712.

[18] [18] Bratko A, Cormack G, Filipic B, Lynam T, Zupan B (2006) *Spam filtering using statistical data compression methods*. Journal of machine learning research 3, 993-1022.

[19] [19] Gomez J C, Moens M (2010). *Using biased discriminant analysis for email filtering.* In proceedings of the 14th international conference KES 2010, Springer-Verlag, Berlin, 566-575.

[20] [20] Sculley D, Wachman GM (2007)*Relaxed online SVM for spam filtering.* In proceedings of the 30th Annual International ACM SIGIR Conference ACM, NY, 9-17.



**Issam Dagher** is an associate professor at the University of Balamand. He finished his PhD from the University of Central Folrida in 1997. His area of interests are neural networks and fuzzy logic. He published many papers in this area.

**Rima Antoun** is an MS student at the University of Balamand. Her area of interests are Machine learning and neural networks.