Hybrid Email Spam Detection Model Using Artificial Intelligence

Samira. Douzi, Feda A. AlShahwan, Mouad. Lemoudden, and Bouabid. El Ouahidi

Abstract—**The growing volume of spam Emails has generated the need for a more precise anti-spam filter to detect unsolicited Emails. One of the most common representations used in spam filters is the Bag-of-Words (BOW). Although BOW is very effective in the classification of the emails, it has a number of weaknesses. In this paper, we present a hybrid approach to spam filtering based on the Neural Network model Paragraph Vector-Distributed Memory (PV-DM). We use PV-DM to build up a compact representation of the context of an email and also of its pertinent features. This methodology represents a more comprehensive filter for classifying Emails. Furthermore, we have conducted an empirical experiment using Enron spam and Ling spam datasets, the results of which indicate that our proposed filter outperforms the PV-DM and the BOW email classification methods.**

Index Terms—**Spam, deep learning, word2vec, bag of word.**

I. Introduction

Email use continues to grow along with other methods of interpersonal communication. In 2017, the total number of business and consumer Emails sent and received per day reached 269 billion. Volume is expected to continue to grow at an average annual rate of 4.4% over the next four years, reaching 319.6 billion by the end of 2021 [1].

Email spam is an increasing problem that not only affects normal internet users but also causes a major problem for companies and organizations [2]. According to annual reports, the average volume of spam Emails sent per day increased from 2.4 billion in 2002 to 300 billion in 2010 [3] [4], and according to the Symantec Intelligence Report, the global percentage of Email traffic defined as spam is 71.9% [5]. Many solutions are being proposed to counteract this 'plague'. Bag of Words (BOW) and machine learning techniques are the most frequently used for automatically filtering Email messages [6].

In the BOW model, emails are represented by vectors in which each dimension corresponds to a word or group of words. It is a representation that is based on frequency to determine the values associated with each dimension of the vector. For example, given a set of terms *V*, $V=\{t_1, t_2,\ldots\ldots, t_n\}$, BOW represents an Email text '*E*' as a n-dimensional feature vector $X=\{x1,\ldots\ldots..,xn\}$, where the value of xi is given as a function of the occurrence of ti in E, depending on the representation of the features adopted. The features are generally given as single words occurring in messages used for training.

While this approach is fast and simple, and has a low-computational cost, it disregards grammar and even word order. Different Emails can have the same representation, since the same words are used. It also suffers from the Curse of Dimensionality. To represent a short sentence, the BOW approach needs a very high dimensional feature vector, which is hugely sparse. In such situations, most classifiers lose their power of discrimination [7].

In this work we show how to build a new representation of each email, based on Paragraph Vector-Distributed memory (PV-DM) [8], and the scheme TF-IDF. The proposed approach gives a compact representation which contains information about the context of an Email, as well as its relevant features. We have conducted an empirical experiment using the public Enron and Ling spam datasets. The reported results indicate that our approach outperforms BOW and even PV-DM representation.

This paper consists of six sections. Section II presents the related work. Next, we introduce the main concepts of the proposed spam filter (Section III). While the proposed approach is highlighted in Section IV. Experimental results are described and discussed in Section V and finally, conclusions and future works are presented in Section VI.

II. Related Work

In the field of email filtering, several different methods have been proposed; Androutsopoulos *et al.* include the use of bag-of-words representations with Bayesian classifiers [9]. Woitaszek *et al.* (2003) used SVM approach to construct an automated classification system to detect unsolicited commercial emails. In their study, several sets of sample messages were collected to build dictionaries of words found in email communications, which were processed by the SVM to create a classification model for spam or non-spam messages. They found that neural networks and SVM are good for spam filters [10]. Johan Hovold assumes that it is possible to achieve very good classification performance using a word-position-based variant of naive Bayes [11]. Kanaris *et al.* use n-grams to produce more robust features for email filtering [12]. Sahami *et al.* (1998) proposed an email filter based on an enhanced Naive Bayes classifier. Recall and precision were improved when phrases and header specific information were added as features [13]. Elisabeth Crawford *et al.* [14] show also that using phrase-based

Manuscript received October 9, 2019; revised December 31, 2019.

Samira. Douzi is with IPSS, Faculty of Science, University Mohammed Rabat, Morocco (e-mail: samiradouzi8@gmail.com).

Feda A. AlShahwan was with University of Surrey UK. She is now with the College of Technological Studies, Kuwait (e-mail: fa.alshahwan@paaet.edu.kw).

Mouad. Lemoudden was with the IPSS, Faculty of Science, University Mohammed Rabat, Morocco. He is now with INRIA Rennes- Bretagne Atlantique, France (e-mail: mouad.lemoudden@gmail.com).

Bouabid El Ouahidi was with University of Caen-France. He is now with the IPSS, Faculty of Science, University Rabat, Morocco (e-mail bouabid.ouahidi@gmail.com).

doi: 10.18178/ijmlc.2020.10.2.937

representation can be used to increase the performance of email classifiers. While Matthew Chang and Chung Keung Poon [15] studied the use of phrases as the basic features in the email classification problem; they found that use of size two phrases generally gives the best classification results. Although many of the email filtering methods perform with high true positive and low false positive rates, there is constant research into novel ways of solving the problem, since spammers are continually evolving their techniques to bypass known filter methodologies.

III. Basic Concepts

In this section we present the main concepts of both the Word embedding and PV-DM approaches that compose the core of the proposed anti-spam.

A. Word Embedding

Word embedding is a model based on the hypothesis: “words that occur in similar contexts tend to have similar meanings” [16]. The core idea of the word embedding is to explore the local context (phrase, sub phrase..) of a missing word, by using the concatenation or the average of previous word vectors, in order to predict .It is computed using neural networks by the following formula:

$$p(w_t / w_{t-k},......,w_{t+k},P) = \frac{e^{y_{wt}}}{\sum_i e^{y_i}} \tag{1}$$

Each of is un-normalized log-probability for each output word, computed as:

$$y = b + Uh(w_{t-k},....,w_{t+k},W) \tag{2}$$

where *U* and *b* are the softmax parameters and *h* is constructed by a concatenation or average of word vectors extracted from the matrix *W*.

Over a large corpus, at every step *t*, the target word vector and the Matrix W are updated to bring similar words close in the vector space [17]. An interesting characteristic of these vectors is that words which appear in common contexts in the corpus are related approximately to each other in the vector space. This ability to capture the semantics of words and the relationships between them is the reason why more and more researchers in the field of natural language processing include in their systems the knowledge extracted by this type of tool [18], [19].

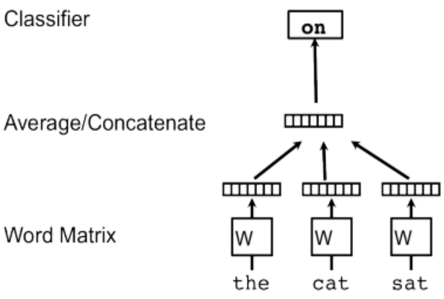

Fig. 1. Process of learning the words vectors [17].

Word2Vec is one of the most popular techniques to learn word embedding using neural network. It was developed by Tomas Mikolov in 2013 at Google [20].

Word2vec involves two models to learn the representations for words: Continuous bag-of-words (CBOW) and Skip gram model.

B. Skip Gram Model

The training objective of the Skip-gram model is to find word representations that are useful for predicting the surrounding words in a sentence or a document. More formally, given a sequence of training words w_1, w_2, and $w_3 \ldots w_{t.}$ the objective of the Skip-gram model is to maximize the average log probability:

$$\frac{1}{t}\sum_{1}^{t}\sum_{-c<j<c,j\neq 0} logp(wt+j \backslash wt) \tag{3}$$

where *c* is the size of the training context (which can be a function of the center word w_t). Larger *c* results in more training examples and thus can lead to a higher accuracy, at the expense of the training time.

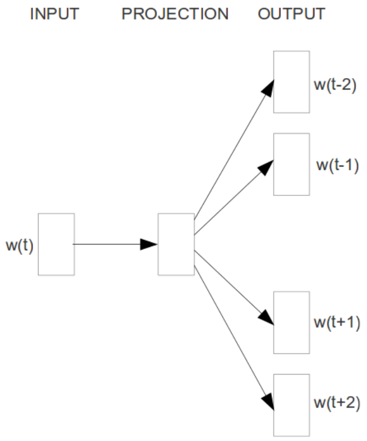

Fig. 2. Process of skip gram model.

C. Continuous Bag-of-Words (CBOW) Model

The CBOW model architecture tries to predict the current target word (the center word) based on the source context words (surrounding words). The order of context words does not influence prediction. The goal of CBOW is to maximize the log probability:

$$\frac{1}{t}\sum_{j=k}^{t-k} \log p(wt|wt-k,\ldots,\ldots wt+k) \tag{4}$$

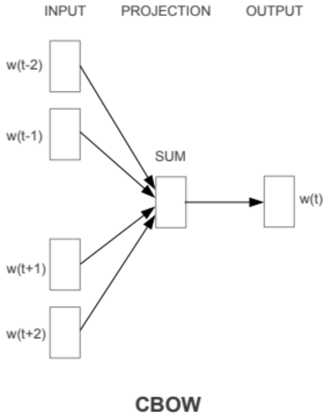

Fig. 3. Process of CBOW model.

By capturing the relationships between words and their context, word2vec by its two models (Skip gram and CBOW) is able to represent words that are semantically close with vectors that are also close to each other.

Doc2Vec is an extension of Word2Vec that attempts to determine an adequate continuous vector for a paragraph or even a larger document in order to preserve the semantic relationship among various documents. Doc2Vec involves two models to learn the representations for documents: PV-DM and DBOW models.

D. Paragraph Vector PV-DM

PV-DM is a Deep learning Algorithm, inspired from Word2Vec model. In the CBOW model of Word2Vec, the model learns to predict a center word based on the context. While the PV-DM model uses the paragraph vector in conjunction with the word vectors to contribute to the prediction task of the next word given.

For example given a set of *n*-1 words, w_1 , w_2 ... w_{j-1} , w_{j+1} , .. w_n in a paragraph where a word is missed, PV-DM predicts the missing word vector by taking into account the other *n*-1 word vectors $v(w_1)$, $v(w_2)$, .., $v(w_{j-1})$, $v(w_{j+1})$ $\cdots v(w_n)$ as well as the paragraph vector. The paragraph vector represents the global context of the word that we are trying to predict. Thus, the paragraph vector and word vectors are averaged or concatenated by a classifier that predicts the missing word [8], so the formula becomes:

$$p(w_t / w_{t-k},......,w_{t+k},P) = \frac{e^{y_{wt}}}{\sum_i e^{y_i}} \quad [8] \qquad (5)$$

where:

$$y = b + Uh(w_{t-k},....,w_{t+k},P,W,D) \qquad (6)$$

U and *b* are the softmax parameters and *h* is constructed from *W* and *D*. In the PV-DM framework, every paragraph in the corpus is mapped to a unique vector, represented by a column in matrix *D* and *W* every word is also mapped to a unique vector, represented by a column in matrix (Fig. 4).

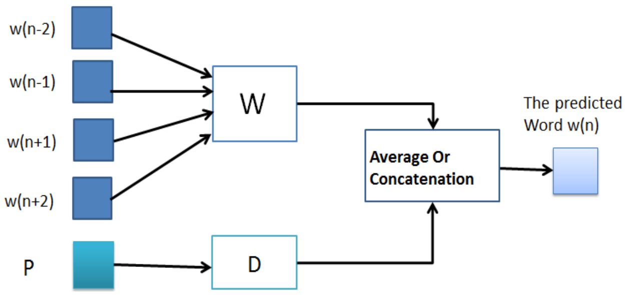

Fig. 4. The framework of PV-DM.

E. DBOW Model

Distributed Bag of Words (DBOW) model is slightly different from the PVDM model. The DBOW model ignores the context words in the input, but force the model to predict words randomly sampled from the paragraph in the output.

This means that at each iteration of stochastic gradient descent, DBOW model samples a text window, then samples a random word from the text window and form a classification task given the Paragraph Vector. This technique is shown in Fig. 5 [11].

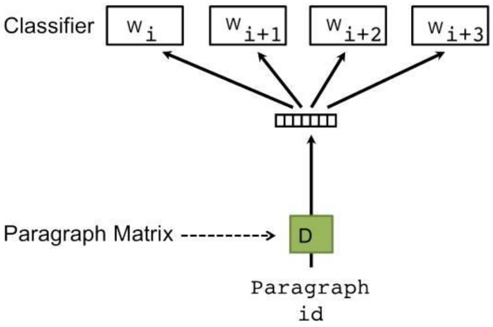

Fig. 5. The framework of DBOW.

F. Feature Selection

High dimensional data is a significant problem in both supervised and unsupervised learning [12] which is becoming prominent with the recent explosion of the size of the avail- able datasets. The main motivation for reducing the dimensionality of the data and keeping the number of features as low as possible is to decrease the training time and limitation of required storage space and reduction of processing cost [13]. Dimensionality reduction methods can be divided into two main groups: Those based on feature extraction and those based on feature selection.

Feature extraction methods transform existing features into a new feature space of lower dimensionality. During this process, new features are created based on linear or nonlinear combinations of features from the original set. Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA) and Autoencoder are examples of such algorithms [14].

Feature selection methods reduce the dimensionality by selecting a subset of features which minimizes a certain cost function. Unlike feature extraction, feature selection does not alter the data, and it is used at the data pre-processing stage before training a classifier. This process is also known as variable selection, feature reduction or variable subset selection .In this paper we are using Term frequency –inverse document frequency algorithm which is a very popular research method in the field of natural language processing (NLP).

G. Term Frequency –Inverse Document Frequency (TF-IDF)

TF-IDF term weight algorithm is widely applied into language models to build NLP Systems. For instance, in SMART system, vector space model (VSM) of text document is put forward by Salton [13]. In the vector space model, a document is represented by a vector of terms. And a term-by-document matrix is used to represent a collection of documents, where each entry represents the weight of a term in a document and is calculated usually via TF-IDF.

TF-IDF works by determining the relative frequency of words in a specific document compared to the inverse proportion of that word over the entire document corpus. Intuitively, this calculation determines how relevant a given word is in a particular document. Words that are common in a single or a small group of documents tend to have higher TF-IDF numbers than common words such as articles and prepositions. The formal procedure for implementing TF-IDF has some minor differences over all its applications, but the overall approach works as follows. Given an emails collection *E*, a word *w*, we calculate the relative frequency of the word *w* in a specific Email through an inverse proportion of the word over the entire Emails corpus. In determining the value, the method uses two elements: tf-term frequency of term i in document j and idf -inverse document frequency of term i.

The algorithm tf_idf can be calculated as:

$$tf\_idf(w_{i,e_j}) = tf(w_{i,e_j}) \times \log(\frac{N}{Ne_{w_i}}) \quad (7)$$

where $tf\ (w_{i,ej})$ is the weight of term *i* in Email *j*, *N* is the number of Emails in the collection, Ne_{w_i} is the number of Emails containing the word *i*. This formula implemented in the framework, and has shown good results.

IV. Proposed Approach

The aim is to overcome the drawbacks of BOW, especially the fact that it ignores the order and the relationship between words. The present study proposes two complementary representations of each email, both based on PV-DM. The first representation describes the global context of an Email, while the second representation describes the local context of pertinent features of each Email. By doing this, we try to provide a better representation that captures the semantic aspect of words by combining the embedded information extracted from both the local and global context of each email.

A. How to Represent an Email by Our Methodology?

To define the baseline of our approach, two vector representations of each Email are calculated by using the deep learning Model PV-DM and the TF-IDF method. PV-DM model was trained to generate a vector for each word and for each Email in the training corpus. The generated vectors were grouped in two matrices:

- Matrix *D* where each column represents a vector representation of an Email.
- Matrix *W* where each column is a vector representation of a word.

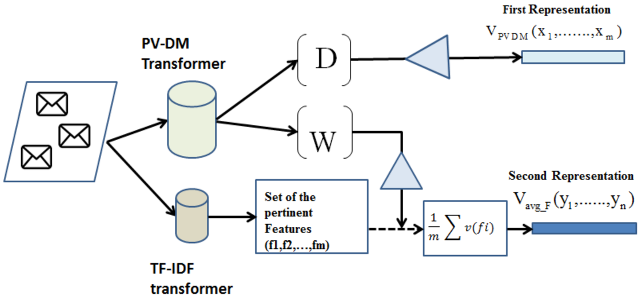

Fig. 6. Structure of the representation phase.

The first vector representation of a given email is obtained by extracting the corresponding vector (column) from the matrix *D*. We call the extracted vector V_{PVDM}.

To obtain the second vector representation of a given Email, the TF-IDF method was applied on the training corpus. The *tf-idf* value increases with the number of occurrences of the word in an Email, but decreases with the number of occurrences of the word in the whole corpus of Emails.

Therefore, TF-IDF method catches only the relevant features with high value. In the next step, however, we will extract the vector representation of the selected features from matrix *W*, to calculate their average. The resulting vector is used as the second email representation vector that we call V_{avg_F}.

V. Experimental Design

A. Corpus

For the task of classification, we applied our proposed approach to two datasets. The first dataset is the Enron spam dataset that is used in several research papers on email classification [7], [20]-[22]. Consisting of 33,702 emails in total, we merge all Enron user messages into a single corpus. In particular, we use the preprocessed form and select six Enron employees: Kaminski-v, farmer-d, beck-s, lokay-m, kitchen-l and William-w3. We also use the spam collection of GP, BG, and spam – assassin_ honeypot. These Emails are randomly divided into a training data set (26960 Emails) and a test data set (6742 Emails).

The second dataset that we used, known as "Ling spam corpus", contains 2892 Emails in total. This dataset was split as follows: 2314 Emails as training data and 578 Emails as testing data.

TABLE I: The Repartition of Data Set for Experiment

Data Set	Training Data		Test Data	
	Ham	Spam	Ham	Spam
Enron Data set	13237	13723	3308	3434
Ling Spam corpus	1930	384	482	96

B. Performance Metrics

We use five popular evaluation metrics to measure the performance of the filtering method proposed in this paper: Recall, Specificity, Accuracy, Precision and F-score .We employ the indexes of confusion matrix (TP, FP, FN, and TN) to calculate these Performance Metrics.

TABLE II: The Indexes of Confusion Matrix

The True Email label	classified as Spam	classified as Ham
Spam	True Positif(TP)	False Negatif(FN)
Ham	False Positif(FP)	True Negatif(TN)

- Recall: can be defined as the probability of correctly classifying spam Emails. Higher Recall indicates that the filter tends not to make FN, but it may make FP. The formula is defined as follows:

$$\text{Recall} = \text{TP}/\ (\text{TP}+\text{FN})$$

- Precision: measures the precision of the filtering method to classify spam emails correctly

$$\text{Precision} = \text{TP}/\ (\text{TP}+\text{FP})$$

- Accuracy: is the capability of the filtering method to correctly classify legitimate Emails and spam Email.

$$\text{Accuracy} = (TP+TN)/\ (TP+TN+FP+FN)$$

- F-score: A popular measure that combines precision and recall by calculating their harmonic mean. This metric represents the fact that classifying as spam only what is really spam is more important than filtering out all the spam.

$$\text{F-score} = 2 \times (\text{Precision} \times \text{Recall})/\ (\text{Precision}+\text{Recall})$$

- ROC curves and Area Under the Curve (AUC): ROC curve is a bi-dimensional graph where the Y axis represents the true positive rate (sensitivity) and the X axis represents false positive rate (1-specificity).One of the main advantages of using ROC curves is the fact that ROC is not sensitive to changes in the class distribution. If the ratio between positive and negative samples in the test database is different from the relationship found in the training database, the ROC curves remain the same [23].
- AUC is another tool used to represent the efficiency of an algorithm by providing a scalar value, which is basically the area under the ROC curve. The higher the AUC the better the algorithm

C. Representation Model

To obtain the vector representation VPVDM, the Doc2vec module from the Genism toolkit [24] implemented in Python has been used and trained with the follows parameters: size=100, window=5 and with 25 training epochs starting with a learning rate of 0.025.While for the second vector representation Vavg_F of an Email, we used the scikit-learn python library implementation of the TF-IDF algorithm [25] with the default parameters and number of features equal to 1000 for the ling spam data set and 1500 features for the Enron spam date set. Then, we calculate the average of the vectors of the selected features contained in each Email.

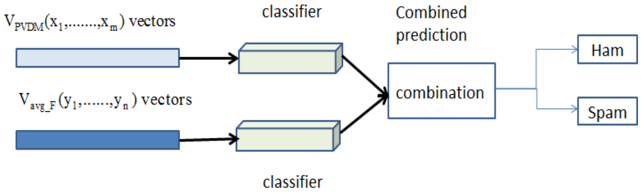
$V_{PVDM}(x_1,.......,x_m)$ vectors
classifier
Combined prediction
Ham
combination
$V_{avg\_F}(y_1,......,y_n)$ vectors
Spam
classifier

Fig. 7. Combined model for predicting.

VI. Experimental Results

In order to evaluate the impact of our approach, we compared the two best-known methods of representation; the deep learning PV-DM model and the BOW model. The performances of our proposed approach are compared to the performances of the two aforementioned methods. The three models are trained with different classifiers on the Ling spam and Enron spam data sets.

We used Receiver Operating Characteristic (ROC) and Area under the Curve (AUC), which are useful for evaluate the performance of our approach.

A. Experimental Results on Ling Spam Data Set

Our experiments were first performed on Ling Spam Data set. The performances of the proposed scheme for detecting attacks were measured by computing the ROC curves. The experimental results indicate that AUC (Figure 8) of the proposed approach was superior to all other techniques and it was able to achieve the highest detection rate.

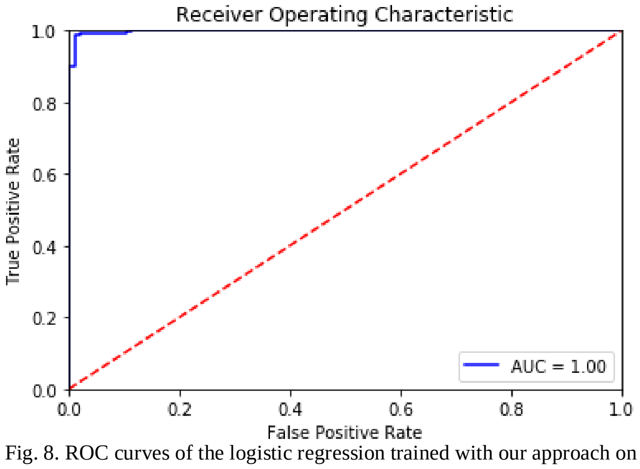

Fig. 8. ROC curves of the logistic regression trained with our approach on ling spam dataset.

TABLE III: Performance Metrics of Logistic Regression Trained with Our Approach , PV_DM Representation and the BOW Model on Ling Spam Dataset

classifier	Model	Accuracy	precision	Recall	F1-score
Logistic regression	PV-DM	0.9619	0.9618	0.9938	0.9805
	Our Approach	0.9827	0.9797	1	0.9897
	BOW	0.8342	0.8342	1.0	0.9096

TABLE IV: Performance Metrics of SVM Trained with Our Approach, PV_DM Representation and the BOW Model on Ling Spam Dataset

classifier	Model	Accuracy	precision	Recall	F1-score
SVM	PV-DM	0.9619	0.9582	0.9979	0.9776
	Our Approach	0.9827	0.9797	1.0	0.9897
	BOW	0.8463	0.8648	0.9669	0.9130

TABLE V: Performance Metrics of KNN Trained with Our Approach, PV_DM Representation and the BOW Model on Ling Spam Dataset

classifier	Model	Accuracy	precision	Recall	F1-score
KNN	PV-DM	0.9756	0.9756	0.9937	0.9740
	Our Approach	0.9827	0.9797	1.0	0.9897
	BOW	0.8238	0.8685	0.9296	0.8980

B. Experimental Results on Enron Data Set

Observing the values of the AUC (Area Under the Curve) from Fig. 9, it is clear that our model was again superior to all other techniques on Enron Data set as well as on Ling spam which proves that the proposed approach offers a significant improvement in terms of accuracy.

In addition, the results show a significant disparity between the performance of PV-DM and Bow on the Ling spam data set and the Enron set. After close inspection, we think that this disparity exists due to the differences in the style of language and message cohesion that exist between the emails of the two data sets. We found that the Emails in the Ling spam data set are generally grammatically correct and follow the conventions of the English language, whereas the emails of the Enron dataset generally do not follow formal language conventions and contain a lot of grammatical errors and inaccurate choice of language items. It is a testament to the quality of our proposed approach that the results were so strong despite the lack of systematic coherence in the language contained in the Enron data set.

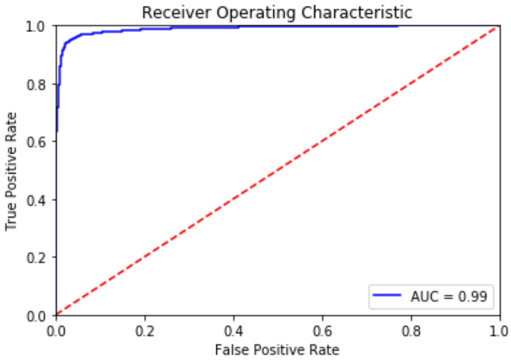

Fig. 9. ROC curves of the logistic regression trained with our approach on Enron Dataset.

TABLE VI: PERFORMANCE METRICS OF LOGISTIC REGRESSION TRAINED WITH OUR APPROACH , PV_DM REPRESENTATION AND THE BOW MODEL ON ENRON DATASET

Classifier	Model	Accuracy	precision	Recall	F1-score
Logistic regression	PV-DM	0.9027	0.9063	0.8942	0.9002
	Our Approach	0.9588	0.9644	0.9510	0.9576
	BOW	0.7195	0.7599	0.6265	0.6868

TABLE VII: PERFORMANCE METRICS OF SVM TRAINED WITH OUR APPROACH, PV_DM REPRESENTATION AND THE BOW MODEL ON ENRON DATASET.

classifier	Model	Accuracy	precision	Recall	F1-score
SVM	PV-DM	0.91115	0.9293	0.8863	0.9073
	Our Approach	0.9616	0.9655	0.9559	0.9607
	BOW	0.5094	1.0	0.0006	0.0012

TABLE VIII: PERFORMANCE METRICS OF KNN TRAINED WITH OUR APPROACH, PV_DM REPRESENTATION AND THE BOW MODEL ON ENRON DATASET.

Classifier	Model	Accuracy	precision	Recall	F1-score
KNN	PV-DM	0.8667	0.8716	0.8540	0.8627
	Our Approach	0.9307	0.9435	0.9135	0.9283
	BOW	0.5736	0.5626	0.5905	0.5762

VII. CONCLUSION

In this paper we propose a novel approach for spam filtering that focuses on the complementary nature of the information provided by the global context of an Email and the local context of its pertinent features. Our method considers the neural network model PV-DM and the scheme TF-IDF, to assign to each message dual representation vectors. The final classification is made by combining the classifications prediction from each vector.

Experimental results clearly confirm that the classifiers trained with our method get the best results and surpass the PV-DM and Bow models. Moreover, they prove that the proposed method is more resistant to differences in the language system and message cohesion.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

AUTHOR CONTRIBUTIONS

Samira. Douzi and Mouad. Lemoudden concuted the research.

Samira. Douzi, Feda A. Alshahwan and bouabid. El Ouahidi analyzed the data.

Samira. DOUZI, Feda A. Alshahwan and Bouabid. El Ouahidi wrote the paper.

All authors had approved the final version.

REFERENCES

[1] *The Radicati Group*, INC. Email Market, 2017-2021.

[2] D. Talbot, "Where SPAM is born," *MIT Technol. Rev*., vol. 111, no. 3, p. 28, 2008.

[3] T. A. Almeida and A. Yamakamil, "Facing the spammers: A very effective approach to avoid junk Emails," *Expert Systems with Applications*, vol. 39, Issue 7, 1 June 2012, Pages 6557–6561.

[4] Symantec Security Center. [Online]. Available: http://www.symantec.com/security_response/landing/spam/

[5] Symantec Intelligence Report. (2013). [Online]. Available: https//Symantec.com/security_response/landing/spam

[6] T. S. Guzella and W. M. Caminhas, "A review of machine learning approaches to spam filtering," *Expert Systems with Applications*, vol. 36, pp. 10206–10222, 2009.

[7] T. Guzella and W. Caminhas, "A review of machine learning approaches to spam filtering," *Expert Syst. Appl*., vol. 36, no. 7, pp. 10206–10222, 2009.

[8] Q. Le and T. Mikolov, "Distributed representation of sentences and documents," presented at the 31 the International Conference on MachineLearning, Beijing, China, 2014.

[9] I. Androutsopoulos, J. Koutsias, K. Chandrinos, and C. Spyropoulos, "An experimental comparison of Naive Bayesian and keyword-based anti-spam filtering with encrypted personal e-mail messages," in *Proc. 23rd ACM SIGIR Conference*, Athens, Greece 2000, pp. 160–167.

[10] M. Woitaszek, M. Shaaban, and Czernikowski. "Identifying junk electronic mail in Microsoft outlook with a support vector machine," in *Proc. Symposium on Applications and the Internet*, 2003, pp. 66–169.

[11] Q. Le and T. Mikolov, "Distributed representations of sentences and documents," arXiv:1405.4053.

[12] J. Hovold, "Naive Bayes spam filtering using word-position-based attributes," presented at 2nd Conference on Email and Anti-Spam, Stanford, CA, 2005.

[13] I. Kanaris, K. Houvardas, and E. I. Stamatatos, "Words vs. character n-grams for anti-spam filtering," *International Journal on Artificial Intelligence Tools*, vol. 16, no. 6, pp. 1047–1067, 2007.

[14] G. Salton, *Introduction to Modern Information Retrieval*, Auckland, McGraw-Hill, 1983.

[15] A. G. K. Janecek, W. N. Gansterer, M. A. Demel, and G. F. Ecker, "On the relationship between feature selection and classification accuracy," in *Proc. the Workshop on New Challenges for Feature Selection in Data Mining and Knowledge Discovery*, 2008, p. 16.

[16] J. S. Lee and D.-W. Kim, "Fast multi-label feature selection based on information-theoretic feature ranking," *Pattern Recognition*, vol. 48, issue 9, pp. 2761-2771, 2015.

[17] B. Trstenjak, S. Mikac, and D. Donko, "KNN with TF-IDF based Framework for Text Categorization," *Procedia Engineering*, vol. 69, pp. 1356-1364, 2014.

[18] M. Sahami, S. Dumaisis, D. Heckerman, and E. Horvitz, "A Bayesian approach to filtering junk e-mail," AAAI Technical Report WS-98-05, 1998.

[19] C. Elisabeth, K. Irena, and P. Jon, "Phrases and feature selection in e-mail classification," in *Proc. the 9th Australasian Document Computing Symposium*, Melbourne, Australia, December 13, 2004.

[20] M. Chang and C. K. Poon, "Using phrases as features in email classification," *Journal of Systems and Software*, vol. 82, issue 6, pp. 1036-1045, June 2009.

[21] T. Mikolov, I. Sutskever, and K. Chen, "Distributed representations of words and phrases and their compositionality," arXiv:1310.4546.

[22] G. E. Hinton, "Learning distributed representations of concepts," in *Proc. the Eighth Annual Conference of the Cognitive Science Society*," Amberst Massachusetts, 1986.

[23] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," arXiv, 1301.3781v3, 2013.

[24] C. Li, L. Ji, and J. Yan. "Acronym disambiguation using word embedding," in *Proc. the Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

[25] S. Bengio and G. Heigold, "Word embeddings for speech recognition," in *Proc. 15th Annual Conference of the International Speech Communication Association*, Singapore, September 14-18, 2014.

[26] Enron Spam Datasets. [Online]. Available: http://csmining.org/index.php/enron-spam-datasets.html

[27] V. Metsis, I. Androutsopoulos, and G. Paliouras, "Spam filtering with Naive Bayes – Which Naive Bayes?" presented at Third Conference on Email and Anti-Spam, July 27-28, 2006.
[28] A. Barushka *et al.*, "Spam filtering using regularized neural networks with rectified linear uni.," presented at XVth International Conference of the Italian Association for Artificial Intelligence, Genova, Italy, 2016.
[29] T. Fawcett. "An introduction to ROC analysis," *Pattern Recognition Letters*, vol. 27, issue 8, pp. 861-874, June 2006.
[30] models.doc2vec – Doc2vec paragraph embeddings. [Online]. Available: https://radimrehurek.com/gensim/models/doc2vec.html.
[31] Scikit Learn. [Online]. Available: http://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.htm

Samira Douzi received the master degree in development quality in 2013, from the Department of Computer Science at the Faculty of Sciences Rabat Agdal. Since 2016 she is a predoctoral researcher in the Department Computer Science at the Faculty of Sciences Rabat Agdal where she is pursuing a Ph.D. degree. Her main researches interests include big data, deep learning and cyber security.

Feda A. AlShahwan is an assistant professor at the Electronic Engineering Department/Computer Section of the College of Technological Studies in the Public Authority for Applied Education & Training.

She obtained her B.Sc., M.Sc. in computer engineer from Kuwait University 1992, 2004 respectively. She has her Ph.D. in "Adaptive Service Provision and Execution in Mobile Environments" from Center for Communications Systems Research in University of Surrey. Her research interests include studies of adaptive mobile web services, social networks, internet of things, artificial intelligent, MANET and mobile cloud computing.

Mouad Lemoudden is a postdoc researcher in INRIA Rennes - Bretagne Atlantique. He holds a Ph.D. in computer science from Mohammed-V University in Rabat, Morocco, where he received his master's degree in applied computer science. His main research interests include computer security, intrusion detection, data science, cloud computing, big data, machine learning, natural language processing, parallel and distributed systems.

Bouabid El Ouahidi is a university professor and ex head of the Computer Science Department. He received Ph.D. degree in computer security from the. University of Caen-France. His research interests include Open distributed systems, quality of services of distributed applications, big data, cyber security and machine learning.