Multi-Classifier System for Authorship Verification task using Word Embeddings

Nacer Eddine Benzebouchi*, Nabiha Azizi
Computer Science Department, Labged Laboratory
Badji Mokhtar Annaba University, PO BOX 12, 23000
Annaba, Algeria
nasrobenz@hotmail.fr, azizi@labged.net

Monther Aldwairi
College of Technological Innovation
Zayed University, P.O. Box 144534
Abu Dhabi, United Arab Emirates
Monther.Aldwairi@zu.ac.ae

Nadir Farah
Computer Science Department, Labged Laboratory
Badji Mokhtar Annaba University, PO BOX 12, 23000
Annaba, Algeria
farah@labged.net

***Abstract*— Authorship Verification is considered as a topic of growing interest in research, which has shown excellent development in recent years. We want to know if an unknown document belongs to the documents set known to an author or not. Classical text classifiers often focus on many human designed features, such as dictionaries, knowledge bases and special tree kernels. Other studies use the N-gram function that often leads to the curse of dimensionality. Contrary to traditional approaches, this article proposes a new scheme of Machine Learning model based on fusion of three different architectures namely, Convolutional Neural Networks, Recurrent- Convolutional Neural Networks and Support Vector Machine classifiers without human-designed features. Word2vec based Word Embeddings is proposed to learn the best word representations for automatic authorship verification. Word Embeddings provides semantic vectors and extracts the most relevant information about raw text with a relatively small dimension. As well as the classifiers generally make different errors on the same learning samples which results in a combination of several points of view to maintain relevant information contained in different classifiers. The final decision of our system is obtained by combining the results of the three models using the voting method.**

Keywords—Deep Learning, Convolutional Neural Networks (CNN), Recurrent- Convolutional Neural Networks R-CNN, Word Embeddings, Authorship Verification, Natural Language Processing (NLP)

I. INTRODUCTION

Text classification is a basic task and a key element in many NLP applications such as information filtering, web search, sentiment analysis and Authorship attribution [1]. For this reason, many studies have been conducted by many researchers in this field.

Authorship Attribution is a current topic dealing with the issue of attributing a given unknown text to an author, taking into account a set of candidate authors for whom the text samples of uncontested authors are available. The text samples corresponding to all authors are generally called the reference set, which will be analyzed in order to obtain the writing style of the candidate authors.

The authorship attribution task is considered as a closed set problem if the reference set contains the actual author otherwise it is an open set problem. The authorship verification is a task that aims to know if a given unknown document was written by a certain author (A) or not. Figure 1 shows an example authorship verification process.

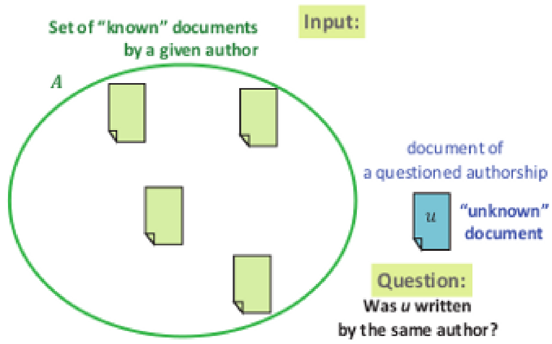

Fig. 1. Authorship Verification Problem

The reference set in the authorship verification includes samples of the alleged author's text. It can be scanned for well-defined features that show the writing style of the author. If they are similarity on the writing style of the given text, then paternity is true, otherwise it is false. A series of authorship verification problems concerning the given candidate authors may be considered as an authorship attribution problem that will be resolved if the verification problem is resolved [2]. Inversely, authorship verification is a specific task of the authorship attribution with an open set of candidate authors. In this respect, authorship verification can be considered as one-class classification problem [3]. This signifies that a text is classified as belonging to the given class (attributed to the assumed author) or on the contrary does not belong to it (classified as a negative example).

Features representation is considered as a major problem in text classification. Generally, this step focuses on the bag-of-words, unigrams, bigrams and n-grams models for features extraction.

For this reason, numerous studies have been carried out in this field and many machine learning techniques have been suggested for authorship attribution task in recent years.

978-1-5386-4543-7/18/$31.00 ©2018 IEEE

Deep Learning is a branch of Machine Learning that can be applied to many problems such as Image Classification, Voice Recognition, and Natural Language Processing (NLP).

Recently, word embeddings and deep neural networks have experienced rapid development, and have created a new field of research in several NLP tasks.

Word Embeddings is a class of approaches to represent words and documents using a dense vector representation and resolves the problem of data scarcity, which can retain the significant syntax and semantics presented in the text [4].

CNN can learn by themselves useful features from raw data and accurately and determines discriminate sentences in a text with a max-pooling layer. Thus, CNN can better preserve the semantics of texts compared to other machine learning techniques. It has demonstrated that represents a successful approach for text classification [5, 6].

CNN works very well in pattern recognition, especially, 2D pattern because it ignores object positions and extracts local features. However, in NLP, the position of a word in the sentence is very important. The Recurrent Neural Network (RNN) simulates the human in reading (which is from left to right) but CNN classifier smoothes and ignores the word order [7]. Actually, RNNs, especially those using Long Short-Term Memory (LSTM) hidden units have demonstrated excellent performance on text representation, especially for long word sequence. In this paper, we would analyze the behavior of LSTM use to learn sentence representations.

This study proposes a multi-classifier approach for author verification using three classifiers namely Convolutional Neural Networks (CNN), Recurrent-CNN and Support Vector Machines (SVM) based on Word Embeddings strategy to generate the words representation. A voting method is adopted to combine the outputs of classifiers knowing that each classifier gives a unique class label output in order to obtain the final decision. To validate our approach, the PAN 2015 datasets for Author verification is used. According to our knowledge, this is the first work using Word Embeddings with Deep Convolutional Neural Networks for Authorship Verification.

This article is organized as follows. An overview of a recent works based Author verification is presented in Section 2. Section 3 describes the general concept of Word Embeddings model and Section 4 explains the main steps of the proposed approach. In section 5, we illustrate the obtained results of our work. Finally, section 6 presents a conclusion of this study.

II. Related Studies

Several studies have been conducted to develop Authorship Verification tools using different machine learning techniques.

In 2004, Koppel et al. [8] presented an authorship verification method named "unmasking". Its purpose is to quantify the dissimilarity between the sample document produced by the suspect and that of other. The experimental evaluation of the approach yields 95.70% of correct verification only for documents of at least 500 words long.

In 2010, Iqbal et al. [9] proposed an approach for email authorship verification by extracting 292 different features and analyzing these features using different classification and regression algorithms. The shown result using the Enron e-mail corpus yielded EER ranging from 17.1% to 22.4%.

In 2011, Chen and Hao's [10] used 150 stylistic features from e-mail messages for authorship verification. The obtained accuracy rates were 84% and 89% for 10 and 15 short e-mails, respectively using 40 authors of the Enron dataset based on the number of analyzed emails.

In 2013, Brocardo et al. [11] used n-gram analysis for authorship verification of short texts merged with a supervised learning technique. They used stylometric techniques through the linguistic analysis styles and writing authors features. The Equal Error Rate (EER) of 14.35% for message blocks of 500 characters using the Enron email dataset involving 87 authors obtained.

In 2015, Maitra et al.[12] proposed an automatic system for authorship verification based on PAN at CLEF 2015 datasets. They have used the Random Forest (RF) classifier to choose the important features among the 17 types of features such as punctuation, sentence length, vocabulary, N-gram and Parts-of-Speech (POS). The same RF classifier was used for the classification stage.

These above mentioned approaches are not applicable in real time as the CNN and Word Embeddings.

III. Learning Word Representations: *Word2vec*

Word2vec is a commonly used tool in recent years because of its best results and its rapid training. Word2vec is associated models set used to generate Word embeddings.

Word Embeddings is a recent and very popular method in the field of natural language processing (NLP) that allows learning vector representations of words from raw texts with a small dimension and capturing syntactic and semantic relationships contrary to other traditional methods such as onehot representation, unigrams, bigrams and n-grams.

Word2vec models are two layer neural networks, shallow, that are formed to rebuild the linguistic contexts of words. It has as input a large body of text and it generates a vector space, with for each single word of the corpus a corresponding vector in space.

Word2vec can use one of two models, namely continuous bag of words (CBOW) or continuous skip-gram in order to generate a words distributed representation.

The bag of words (CBOW) model uses the context to predict a target word and the skip-gram model does the opposite of the latter, uses a word to predict a target context.

The Skip-gram model is an essential choice lately of several NLP tasks; it forms the embeddings of the words $w_1, w_2, \dots w_T$ by maximizing the average probability of log.

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\le j\le c, j\neq 0}\log p(w_{t+j}|w_t) \quad (1)$$

$$p(w_b|w_a) = \frac{\exp(e(w_b)^T\, e(w_a))}{\sum_{k=1}^{|V|}\exp(e\ (w_k)^T\, e(w_a))} \quad (2)$$

Where, |V| represents the vocabulary of text without a label.

$e'(w_i)$ is other embedding for w_i.

Some speed-up approaches like hierarchical softmax are used, which justifies the use of the embedding e and e' is not calculated in practice.

IV. PROPOSED APPROACH

The creation of our network by the proposed approach, illustrated in Fig. 2, was obtained after several experiments and after a deep study of the literature for other NLP tasks.

Majority voting is considered to be one of the effective methods of fusion [13]. In the process of majority voting method, a decision to choose the label of an input sample is produced by each classifier. The class that has the highest number of votes is determined as the representative class of all the classifiers in the set.

Let $C = [c_1, c_2, \dots c_L]$ is a set of L classifiers, x is the input

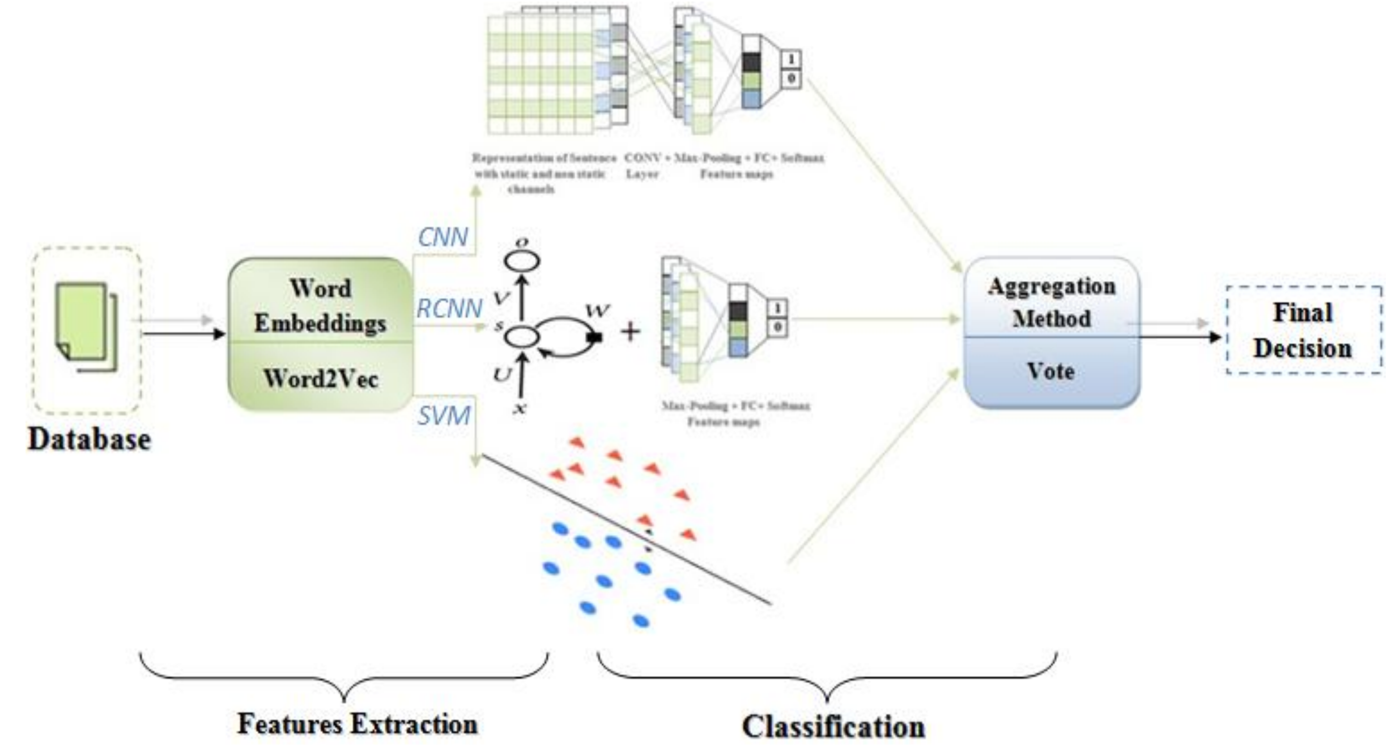

Fig. 2. Proposed Network Architecture for Authorship verification

sample and $c_{i,j}$ is the output of the i^{tn} classifier for the j^{th} class. The final decision using the majority voting can be defined as follows:

$$MV(x) = max_{j\in\Omega} \sum_{i=1}^{L} c_{i,j} \tag{3}$$

However, the specific accuracy of each classifier is not considered in the final decision, which is considered to be the main flaw in the majority voting methods.

In general, the chosen classifiers do not have a similar skill. Consequently, the weighted voting method is used to combine the decision of the chosen classifiers.

In this aggregated method, the result of each classifier is weighted by a coefficient that affects the combination process. Note that w_i is the weight of the i^{th} classifier, the weighted majority voting is defined as:

$$WMV(x) = max_{j\in\Omega} \sum_{i=1}^{L} w_i c_{i,j} \quad With \ \sum_{i=1}^{L} w_i = 1 \tag{4}$$

Many schemes have been suggested to estimate the weights of classifiers. Usually, the weights are estimated using the specific accuracy of each classifier. Let a_i and a_j be the

A. Integration of several learning models

Recently, the combination of classifiers has been proposed as a research path to make recognition more reliable by using the complementarity that can exist between classifiers; the reason why we adopt the multi-classifier approach which consists of Convolutional Neural Networks (CNNs), Recurrent- Convolutional Neural Networks (R-CNNs) and Support Vector Machines (SVM) classifiers based on word2vec Word Embeddings.

Convolutional neural network and recurrent neural network are two main architectures of deep neural networks. SVM is considered to be among the best classifiers in bi- class problem as in our authorship verification problem.

In our first experiment we use the CNN as a feature extractor and classifier at the same time, which is able to extract position invariant functions. Simultaneously, we use RNN especially Long Short-Term Memory (LSTM), in order to apply a recurrent structure to capture contextual information as far as possible when learning word representations, which may introduce considerably less noise compared to traditional window-based neural networks and we also use a maximum pooling layer that determines automatically which words play a key role in the classification of texts. The third used classifier is SVM classifier for the classification stage by introducing the word Embeddings (Features).

To generate the final decision of our system, a vote-based merge technique is applied:

accuracies of the i^{th} and j^{th} classifiers on the validation set. The weight w_i is calculated by:

$$w_i = \frac{a_i}{\sum_j a_j} \quad (5)$$

In this study, the best-worst weighted Vote (BWWV) scheme is used as a measure to quantify the weights. The main idea behind this scheme is to identify the best and worst members of the set using their estimated error on the validation set. The a_i values are determined using the following expression:

$$a_i = \mathbf{1} - \frac{e_k - e_b}{e_w - e_b} \quad (6)$$

Where, e_w indicates the maximum error among the classifiers and e_b is the minimum error. The a_i value varies in [0,1], where 0 indicates the worst classifier and 1 corresponds to the best classifier.

B. CNN Architecture and Conception

Four (04) layers are used in our CNN Architecture: input layer, Convolutional layer C, Subsampling layer S, Dense layer D and Output layer O.

The Convolutional Neural Networks are currently the most powerful models for classifying images [15, 16]. They have two distinct parts. At the input, an image is provided in the form of a matrix of pixels. The first part of a CNN is the conventional part itself. It functions as an extractor of image features. An image is passed through a succession of filters, or convoluted nuclei, creating new images called convolution maps. Some intermediate filters reduce the resolution of the image by a local maximum operation. In the end, the convolution maps are combined in a feature vector, called a CNN code.

This code CNN got out of it from the convolutive party is then connected in the entry of a second part, constituted by completely connected layers (multilayer perceptron). The role of this part is to combine the features of the code CNN to classify the image. The output is a last layer with one neuron per category.

Instead of the pixels of an image, we use a matrix representing the sentences and words of our corpus; each line of the matrix corresponds to a word (word embeddings). Now we can apply filters of the convolution layer and preferably use filters of width equivalent to the vectors dimension.

We note that d is the dimension of a word vector; s is the sentence size and s × d is the dimension of the sentence matrix.

Subsampling layer has several advantages including ensuring that the output matrix will be of a fixed size equal to the number of filters used. Thus, it makes it possible to use sentences of different sizes and different filter sizes to always obtain the same output dimensions. In addition to this, this method also ensures the selection of features process because each filter will try to detect a specific feature (only the maximum value of each feature map is considered) in order to generate the relevant features for the classification stage.

C. Training

The configuration of our neural network is carried out after several tests of performance. We start with the convolution layer as introduction the word Embeddings with an activation function "*Tanh*" followed by the creation of the convolution blocks with activation. After each convolution layer, we carry out a normalization of the batches and we pass to a normalization of batch per block after the increase of the Number of feature maps. We use a stochastic gradient descent with a momentum of 0.9, L2 (0.0005) as a regularization method for weight and bias, and a low learning rate of 0.0001 to train our neural network.

We use one convolution layer and one sub-sampling layer which are structured one after the other with a commonly used *ReLU* function. For the convolution layer, a stride (1, 1) is performed, and for sub-sampling layer, a size of 2x2 for kernel size is used and stride with the Max-pooling function is applied. For the dense layer, we use the *ReLU* function, also the Square Mean Error (MSE) function has been used to optimize the loss function. Finally, *Softmax* function is used for classification.

V. EXPERIMENTAL RESULTS

In order to evaluate the use of the word2vec vector representation for document classification and to benefit from the merger complementarity of several classifiers on the one hand, and to analyze the behavior of the ensemblist approach generated according to the adopted Word2vec model (Skip-Gram or CBOW) on the other hand, three models have been used which are CNN, RCNN and SVM classifiers. In the sections that follow, the details of the experiments and their results are represented.

A. The Features Extraction by using Word2Vec approach

We construct various models based on different architectures and the dimensionality of words vector using word2Vec. Table1 illustrates the training parameters used for several models based on the Skip-Gram and CBOW models.

The word2vec configuration accepts a number of hyper-parameters. Some explanations of these parameters are presented below:

- **Layerize:** defines the features number in the word vector. This is equal to the number of dimensions in the feature space.
- **WindowSize:** when processing a word in the corpus, number of contextual words to be taken into account for sampling purposes.
- **MinWordFrequency:** is the minimum number of occurrences of a word that the corpus must contain for this word to appear in a generated ontolection. Here, if it appears less than 5 times.
- **Iterations:** this is the number of times you allow the network to update its coefficients for a batch of data.
- **LearningRate:** is the step size for each update of the coefficients, when the words are repositioned in the feature space.
- **Sampling:** is the word sampling frequency.

The Skip-Gram method gives good performance and produces more accurate results.

TABLE I. Model Configurations

Model	layerSize	windowSize	minWordFrequency	iterations	learningRate	sampling
Skip-Gram	200	10	10	3	1.0E-4	1.0E-5
CBOW	200	10	10	3	1.0E-4	1.0E-5

B. *Data description*

In our experiments, we use PAN 2015[1] datasets. This database has been developed, with 100 authors, including 100 known documents and 100 unknown documents written in English; for each author A_i, a known document and another unknown are provided and the task consists of determining whether a known and interrogated document belongs to the same author or not. The truth data of the training corpus are founded in truth.txt file including one line per problem and the correct binary answer (Y means the known and the questioned documents are by the same author; N means the opposite).

The implementation of the proposed work was done with Deeplearning4j. Deeplearning4j[2] is the first commercial-grade, open-source, distributed deep-learning library written for Java and Scala.

C. *Evaluation criteria*

In order to evaluate the performance of our system, we use the following evaluation criteria: accuracy, sensitivity, and specificity.

D. *Results*

The following tables summarize the individual performance of each used classifier.

As shown in Table2 (CNN), 48 and 47 documents considered as known documents are exactly classified as known documents respectively according to the Skip-Gram and CBOW techniques, 47 and 44 unknown documents are correctly classified as foreign documents respectively using Skip-Gram and CBOW techniques.

TABLE II. Confusion Matrix of CNN results using both word2vec models

Word2Vec Technique		Unknown 'Known'	Unknown
Skip-Gram	**Unknown 'Known'**	48	2
	Unknown	3	47
CBOW	**Unknown 'Known'**	47	3
	Unknown	6	44

In summary, 95 documents were accurately labeled by the proposed method, resulting in 95% accuracy using Skip-Gram technique with sensitivity 96%, specificity 94%. For the CBOW technique, 91 documents were accurately labeled by the proposed method, resulting in 91% accuracy with sensitivity 94% and specificity 88%.

Table3 (Recurrent-CNN) illustrates that, 48 and 45 unknown (known) documents are formally classified as documents known respectively using the Skip-Gram and

CBOW techniques, 49 and 48 unknown (unknown) documents are properly classified as documents unknown respectively according to Skip-Gram and CBOW techniques.

TABLE III. Confusion Matrix of R-CNN results using both word2vec techniques

Word2Vec Technique		Unknown 'Known'	Unknown
Skip-Gram	**Unknown 'Known'**	48	2
	Unknown	1	49
CBOW	**Unknown 'Known'**	45	5
	Unknown	2	48

In summary, 97 documents were accurately labeled by the proposed method, resulting in 97% accuracy using Skip-Gram technique with sensitivity 96%, specificity 98%.

For the CBOW technique, 93 documents were accurately labeled by the proposed method, resulting in 93% accuracy with sensitivity 90% and specificity 96%.

Table4 (SVM) illustrates that, 47 and 44 unknown (known) documents are correctly classified as documents known respectively using the Skip-Gram and CBOW techniques, 48 unknown (unknown) documents are exactly classified as documents unknown respectively according to both Skip-Gram and CBOW techniques.

TABLE IV. Confusion Matrix of SVM results using word2vec models

Word2Vec Technique		Unknown 'Known'	Unknown
Skip-Gram	**Unknown 'Known'**	47	3
	Unknown	2	48
CBOW	**Unknown 'Known'**	44	6
	Unknown	2	48

In summary, 95 documents were accurately labeled by the proposed method, resulting in 95% accuracy using Skip-Gram technique with sensitivity 94%, specificity 96%.

[1] http://pan.webis.de/data.html
[2] https://deeplearning4j.org/

By using the CBOW technique, 92 documents were accurately labeled by the proposed method, resulting in 92% accuracy with sensitivity 88% and specificity 96%.

By comparing the results obtained, we conclude that the Skip-Gram technique is better than the CBOW technique, surpasses it in all cases and gives good performances. The following table summarizes the obtained results by our networks.

TABLE V. Obtained results of the proposed method

Model	Accuracy	Sensitivity	Specificity
CNN+Skip-gram	95%	96%	94%
CNN+CBOW	91%	94%	88%
RCNN+Skip-gram	97%	96%	98%
RCNN+CBOW	93%	90%	96%
SVM+ Skip-gram	95%	94%	96%
SVM+CBOW	92%	88%	96%

The overall sensitivity of our method is improved by combining the outputs of the three models, namely the decision based on CNN, RCNN and SVM using Skip-Gram technique in order to obtain the final decision of authorship verification by using the Best-Worst Weighted Vote (BWWV) method with 97.4% classification rate.

VI. Conclusion

Authorship Verification is an important task and a very recent domain of research in natural language processing applications, which will determine if a given unknown document belongs to a certain author or not.

For this reason, several studies have been carried out in this field and many techniques have been proposed.

In this paper, we proposed an authorship verification system that depends on fusion of several classifiers namely Convolutional Neural Network (CNN), Recurrent-CNN and Support Vector Machines (SVM) using *word2vec* word embeddings method. Our model is able to learn meaningful texts without any artisanal features.

The Best-Worst Weighted Vote (BWWV) method is used to merge the outputs of the three models and benefit from the complementarity of these three classifiers in order to obtain the overall sensitivity of our approach.

Future work also focuses on the integration of other languages such as Arabic.

In conclusion, the proposed method shows very encouraging results with great precision and is comparable to the results of the state of the art using the same PAN database.

Acknowledgment

This work was supported in part by Zayed University Research Office, Research Cluster Award # R17079

References

[1] A. Rabab'ah, M. Al-Ayyoub, Y. Jararweh, and M. Aldwairi, "Authorship Attribution of Arabic Tweets," In the Proc. of the13th ACS/IEEE International Conference on Computer Systems and Applications (AICCSA 2016), Agadir, Morocco, Nov 29 – Dec 02, 2016.

[2] M. Koppel and Y. Winter, "Determining if two documents are by the same author," JASIST, vol. 65 (1), 2014, pp. 178-187

[3] B. Stein, N. Lipka, and S. Meyer zu Eißen, " Meta analysis within authorship verification," 19th International Workshop on Database and Expert Systems Applications (DEXA 2008), 1–5 September 2008, Turin, Italy, IEEE Computer Society, 2008, pp. 34-39

[4] T. Mikolov, W.t. Yih, and G. Zweig, "Linguistic regularities in continuous space word representations," In NAACL-HLT, 2013, pp. 746–751

[5] X. Zhang, J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification," In: Advances in Neural Information Processing Systems, 2015, pp. 649–657

[6] P. Shrestha, S. Sierra, F. Gonzalez, M. Montes, P. Rosso, and T. Solorio, "Convolutional neural networks for authorship attribution of short texts," In: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, Association for Computational Linguistics, Valencia, Spain, vol. 2, 2014, pp. 669–674

[7] M. Aldwairi and K. Al-Khamaiseh, "Exhaust: Optimizing Wu-Manber Pattern Matching for Intrusion Detection using Bloom Filters," The 2nd World Symposium on Web Applications and Networking (WSWAN'2015), Sousse, Tunisia, 21-23 March 2015, DOI: 10.1109/WSWAN.2015.7209081

[8] M. Koppel and J. Schler, "Authorship verification as a one-class classification problem," In Proceedings of the 21st international conference on Machine learning, ICML '04, pp. 62–, New York, NY, USA, 2004, https://dl.acm.org/citation.cfm?doid=1015330.1015448

[9] F. Iqbal, H. Binsalleeh, B.C. Fung, and M. Debbabi, "Mining writeprints from anonymous e-mails for forensic investigation," Digital Investigation, vol. 7(1-2), 2010, pp. 56 –64

[10] X. Chen, P. Hao, R. Chandramouli, and K.P. Subbalakshmi, "Authorship similarity detection from email messages," In Proceedings of the 7th international conference on Machine learning and data mining in pattern recognition, MLDM'11, Berlin, Heidelberg, Springer-Verlag, 2011, pp. 375–386

[11] M.L. Brocardo, I. Traore, S. Saad, and I. Woungang, "Authorship Verification for Short Messages using Stylometry," the 2013 International Conference on Computer, Information and Telecommunication Systems (CITS), IEEE Conference Publications, University of Piraeus 80, Karaoli & Dimitriou St. Athens, Greece , 2013, pp. 1-6

[12] P. Maitra, S. Ghosh, and D. Das, "Authorship Verification – An Approach based on Random Forest," CLEF 2015 - Conference and Labs of the Evaluation forum, CEUR Workshop Proceedings,Toulouse, France, September 2015, ISSN 1613-0073

[13] B. Krawczyk, "Untrained weighted classifier combination with embedded ensemble pruning," Neurocomputing, vol.196,2016, pp.14–22

[14] N. Azizi, N. Farah, M. Sellami, and A. Ennaji, "Using Diversity in Classifier Set Selection for Arabic Handwritten Recognition," Multiple Classifier Systems, 2010, pp. 235-244

[15] N.E. Benzebouchi, N. Azizi, and K. Ayadi, "A Computer-Aided Diagnosis System for Breast Cancer Using Deep Convolutional Neural Networks," Proceedings of the 4th International Conference on CIDM 2017, Computational Intelligence in Data Mining, Advances in Intelligent Systems and Computing, Springer, No. 52, DOI. 10.1007/978-981-10-8055-5_52

[16] N.E. Benzebouchi, N. Azizi, and S.E. Bouziane, "Glaucoma Diagnosis Using Cooperative Convolutional Neural Networks," International Journal of Advances in Electronics and Computer Science, vol.5(1), jan-2018, pp. 31-36, ISSN: 2393-2835