IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

*Digital Object Identifier 10.1109/ACCESS.2017.Doi Number*

# Ensemble Methods for Instance-based Arabic Language Authorship Attribution

**Mohammed Al-Sarem[1,2], Faisal Saeed[1], Abdullah Alsaeedi[1], Wadii Boulila[1,3] and Tawfik Al-Hadhrami[4]**

[1]College of Computer Science and Engineering, Taibah University, Medina 344, Saudi Arabia
[2]Faculty of IT&CS., Saba'a Region University, Ma'areb, Republic of Yemen
[3]RIADI Laboratory, National School of Computer Sciences, University of Manouba, Tunisia
[4]School of Science and Technology, Nottingham Trent University, Nottingham, NG11 8NS, United Kingdom.

Corresponding author: Mohammed Al-Sarem (e-mail: mohsarem@ gmail.com).

**ABSTRACT** The Authorship Attribution (AA) is considered as a subfield of authorship analysis and it is an important problem as the range of anonymous information increased with fast growing of internet usage worldwide. In other languages such as English, Spanish and Chinese, such issue is quite well studied. However, in Arabic language, the AA problem has received less attention from the research community due to complexity and nature of Arabic sentences. The paper presented an intensive review on previous studies for Arabic language. Based on that, this study has employed the Technique for Order Preferences by Similarity to Ideal Solution (TOPSIS) method to choose the base classifier of the ensemble methods. In terms of attribution features, hundreds of stylometric features and distinct words using several tools have been extracted. Then, Adaboost and Bagging ensemble methods have been applied on Arabic enquires (Fatwa) dataset. The findings showed an improvement of the effectiveness of the authorship attribution task in the Arabic language.

**INDEX TERMS** Authorship attribution, Ensemble methods, Stylometric features, TOPSIS method

## I. INTRODUCTION

From linguistics analysis perspective, authorship attribution (AA) aims to identifying the original author of unseen text. The idea is basically formulated as follows: for each author, there are a set of features that distinguish his writing style from others. Despite author's writing style that can change from topic to topic, some persistent uncontrolled habit and writing styles are still valid over the time. The author of anonymous text can be recognized by matching the observed writing style to one of the candidate author set. From the 19th century, several approaches have been proposed to tackle the AA problem. The early approaches had a statistical background [1-4] where the length and frequency of words, characteristics, and sentences were used to characterize the writing style. These approaches, in general, were human expert-based [5] and the applications also covered literary, religious and legal texts [6]. From sixties of the last century up until 1990s, both the approaches and application were shifted to cover new challenging problems such as the source code attribution [7-9], spam detection [10,11], and plagiarism [12-15]. The approaches at that time were aimed to quantifying the writing style by extracting some features from the text. Although the statistical approaches are good to identify the author of long documents, they suffer when the length of the text, under investigation, is short. The main challenges in such cases include: are the small extracted features sufficient enough to make a fair attribution? how can we improve the precision of the authorship attribution? does the size of the training set effect on the result? what does happen if the dataset unbalanced? what is the optimum data size?

Recently, current studies in authorship attribution benefit from explosion in machine learning domain [16] where the AA task can be considered as a multi-class, single-label classification problem [17]. Basically, the machine-learning approach tackles the AA problem by assigning class labels to text samples. Surveying the literature, we found a large number of methods and approaches that were developed to tackle the AA problem such as Support Vector Machine (SVM) [18-23], Naive Bayes [4, 20, 24-25], Bayesian classifiers [25-27], k-nearest neighbor [28,29], decision trees [29,35]. Although the ensemble methods showed a good

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.2964952, IEEE Access



performance to improve machine learning results, few studies such as [30-34] employed them in AA area. The ensemble methods combine several classifiers in order to decrease variance (bagging) and bias (boosting) and then new data are classified by taking a (weighted) vote of their predictions.

Arabic language is the mother tongue for more than 250 million people reside mainly on two different continents. However, the works on AA for Arabic are still less numerous than those on English [5,23,35-45]. Thus, this paper aims to bridge the gap and investigates whether applying the ensemble methods lead to improve the accuracy of the AA task in the Arabic language, in addition to selecting the base classifier for ensemble methods and optimal combination of features. Furthermore, since appropriate tuning of the size of the training set and feature data set can render significantly lighter the machine-learning processing [17], this paper gives some recommendations for selecting the optimal settings of data set size that maximizes the accuracy of classifiers.

The rest of the article is structured as follows: Section 2 presented the related studies on authorship attribution. it also reviews the studies on the Arabic Language Authorship Attribution (ALAA) and a set of base classifiers were chosen. Section 3 presents the experimental setup, datasets used, and techniques employed. The results and their discussion are given in Section 4. Finally, we conclude the study in Section 5.

## II.RELATED STUDIES

While AA can be considered as a particular type of authorship analysis, ensemble methods is a known approach in machine learning where a set of classifiers with their results are focused in some way to obtain better decisions [47]. In this section, we briefly describe what the authorship attribution is, the features used, and the typical machine-learning based attribution process. Then, we also present some techniques for improving the classification accuracy of class-imbalanced data. In addition, a review on Arabic Authorship Attribution (ALAA) was presented.

### A. AUTHORSHIP ATTRIBUTION

As earlier said, authorship attribution can be considered as a subfield of authorship analysis. It is about identifying the author(s) of an anonymous text document depending on document's characteristics or features. In literatures, such characteristics or features are known as author's writing style or stylo-features [25]. These features are extracted in deferent ways based on how the AA algorithm covers the whole samples. In general, these ways are categorized into two major groups: profile-based and instance-based approaches [16]. While the former group extract stylo-features by concatenating all the samples, that belong to a particular author, within the training set in one big file, the latter group handles each sample in the training corpus of each author separately and in consequence extracts the writing style features from each document (see Fig. 1). In addition, the former group of approaches enables to catch the most persistent and uncontrolled habits in author's writing style, whilst the latter group enables to detect any variation in the writing style. Thus, a combination of both ways is a practical instrument to improve the accuracy of attributing process.

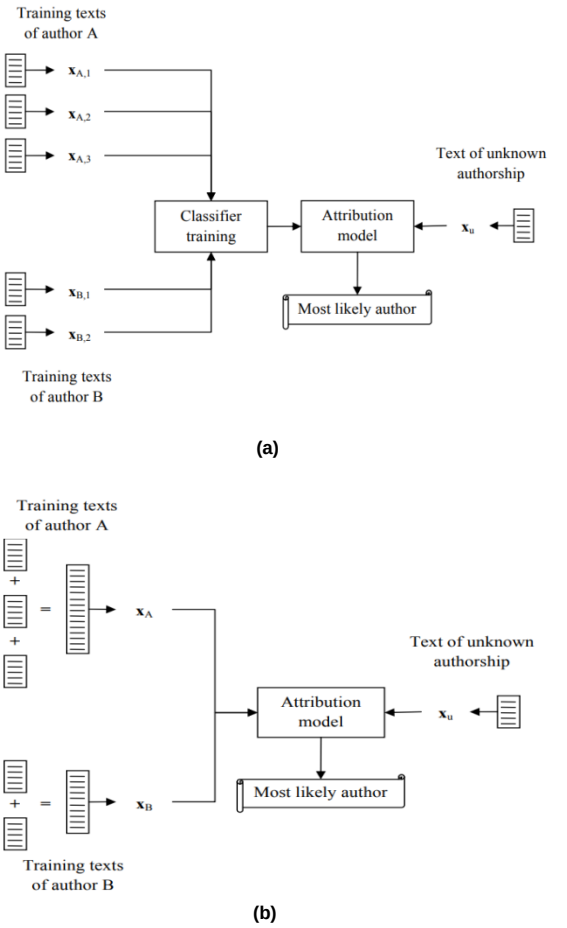


**FIGURE 1. A typical architecture for authorship attribution task [16]: (a) instance-based approaches, whereas (b) profile-based approaches.**

#### 1) AUTHORSHIP ATTRIBUTION PROCESS

Typically, the authorship attribution goes through two main stages: features acquisition, and attribution model construction. The features acquisition is a process where author's writing styles are extracted regardless the way that is used to handle the training text corpus. The earlier attempts to handle stylo-features go back to 19th century. Most of such methods were statistical attempts in its nature where the researchers have tried to quantify the writing style. However, with emergence the Internet, a vast amount of electronic texts was produced and the need for handling these texts are increased. In the shadow of these needs, domains such machine learning, natural language processing, and information retrieval have impact in guiding the authorship attribution research directions.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.2964952, IEEE Access



Back to the earlier era of authorship attribution, we can classify the used features in attributing stage into two main classes: unitary invariant class and multivariate analysis which are both classified as human expert-based approaches. The unitary invariant class uses only a single feature, such as word length, words frequencies, and sentence length to distinguish between authors. The unitary invariant methods gave unreliable results. The multivariate analysis methods, on opposite, deal with a set of features to statistically attribute texts. Methods such Bayesian statistical analysis [4], Principal component analysis (PCA) [49], Linear discriminant analysis (LDA) [50], and Distance-based methods [25;51-54] are used to attribute the texts.

The attribution model construction aims to build an adequate model that can classify the anonymous texts and match them to the right author. With the development of machine-learning techniques, the accuracy of attribution model is enhanced obviously [16].

Machine learning is a branch of artificial intelligence concerned with learning computer systems directly from examples, data, and experience. Learning methods can be categorized into two groups: supervised machine learning methods and unsupervised ones. In supervised methods, dataset is divided into sets: training set and testing set. The former set is used to learn classifiers how to predict class labels, whilst data outside the training set (called testing set) is used to evaluate how well the model does. Classification and regression analysis are the common supervised learning task. Unsupervised methods are type of learning methods that is used to find patterns in data. It does not require to split data or label them. Data visualization and clustering are classified as unsupervised learning methods.

The goal of applying machine-learning methods in AA task is concludes in building a vector of features extracted from the training text corpus, then build a classifier that can attribute anonymous texts on the testing corpus. Figure 2 shows a typical machine-learning based of an authorship attribution process.

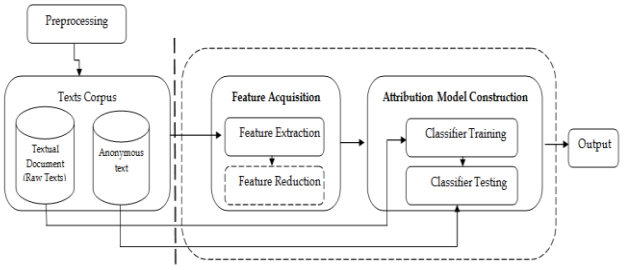


**FIGURE 2. a Typical Machine-learning based authorship attribution process. The reduction phase surrounded in dashed lines is optional step depends on the complexity of space dimensions.**

### 2) AUTHORSHIP ATTRIBUTION FEATURES

As earlier state, the authorship attribution process begins with building a vector of features elicited from the text under consideration. The aim of this step is to extract "writing style" features which are internal characteristics of text. Surveying authorship attribution studies, these features can be categorized into: lexical, character, syntactic, semantic, content-specific, structural and language-specific [35,47, 16].

- Lexical features are one of the most common features used to attribute authorship [5]. Such features can be extracted from a text by tokenizing text into list of words, sentences, numbers, and even punctuation marks. Indeed, in a case of applying the lexical features, results of AA is dependent on the ability of tokenizer to detect the boundaries of words and sentences[1].
- Character, the character features can be considered as subset of lexical features where the text content are treated as a sequence of characters. The character features are partial language-dependent which means features such uppercase and lowercase characters cannot count in e.g. Arabic.
- Syntactic, from text to another, the author may tend to use similar syntactic patterns unconsciously. These patterns can be a more reliable authorial fingerprint than the lexical features. However, they require a specific parser to analyze the text. The most common syntactic measure is a part-of-speech (POS) [16].
- Semantic, on opposite of aforementioned features, semantic features are high-level natural language processing task. Surveying literatures, only a few attempts address semantic features.
- Application-specific, these features can be either structural, content-specific, and language-specific. author's signature, font colors, and font size are obvious structural features used for attributing author [55]. Content-specific features can be extracted from the available texts only and only if all authors, in corpus, are of the same topic. The language-specific features are also common in attributing author. However, to measure them, it has to be defined manually.

## B. ENSEMBLE LEARNING

Improving accuracy of a classifier model is a critical task. One way to do that is by fusing the output of a set of classifiers which called in data mining domain as "ensemble methods". It is obvious that classifiers are vary in its accuracy and some of them perform better others in some cases. Thus, finding a way to combine them tend to be more accurate than working with each classifier separately. Ensemble methods are type of learning algorithms that combine a set of classifiers and then use a (weighted) vote of their prediction for classify new data points. Current section highlights some aspects of ensemble methods. It gives a brief introduction of the most common methods: bagging, boosting, and random forests.

[1]Languages, such as, Chinese and Arabic, require a specific tokenizes to detect words boundaries.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.2964952, IEEE Access



#### 1) ENSEMBLE METHODS

As earlier stated, an ensemble combines a set of classifiers "base classifiers". The ensemble performs e.g., majority voting method to prioritize class label of each classifier and outputs the class in majority. Due to the fact that a separated classifier may make a mistake, the ensemble will misclassify only if over half of the base classifiers are in error. Thus, the accuracy of an ensemble is more accurate than its base classifiers [56]. The most popular ensemble methods used in machine learning domain are bagging [], boosting [] and random forest [].

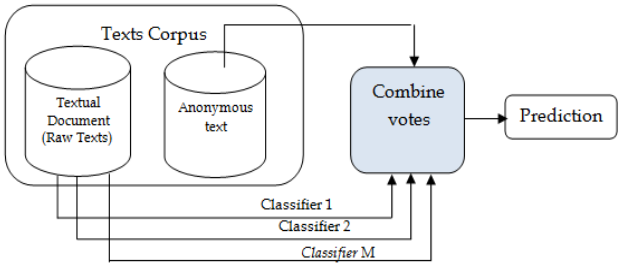


**FIGURE 3. Illustrative ensemble learning methods for AA: the ensemble method generates a set of classifiers for a training set, the class of the unseen text is labeled and voted by each classifier. The ensemble, then, combines the votes and returns a class prediction.**

#### 2) SELECTION OF BASE CLASSIFIER OF ENSEMLE METHODS

The diversity of existing machine learning classifiers that one can select as a base/weak classifier of the ensemble method makes such selection a challenging task. In [77], Zhou et al., proposed a genetic algorithm-based selective ensemble approach. The proposed approach aimed at selecting the appropriate classifiers for composing an ensemble from a set of available classifiers. However, like any optimization based approaches, falling in a local optimum point is probable. Hence, the researchers have proposed other approaches. Lazarevic and Obradovic proposed a clustering-based approach [78] which uses k-means to identify the groups that had similar classifiers and then eliminated redundant classifiers that were in each cluster. The similar approach is also found in [79] where the hierarchical agglomerative clustering algorithm is used. However, the empirical analysis shows that the clustering-based selective ensemble techniques have a bad influence on the effect [80]. In [81] ranking-based method is proposed. The results showed an improvement in the performance of the ensemble. However, the ranking-based techniques are also time-consuming and require a large amount of storage. At this end, selection the right base classifier plays the vital role in minimizing the total misclassification errors as well as the cost of training. The selection process of base classifier can be led by many factors: accuracy of classification, ability of the base classifier to deal with high dimensional data and its performance when the dataset size is increased, and sensitivity to noise data. Decision tree, in particular, C4.5 is considered a robust learner against noisy data, whereas support vector machine (SVM) is more noise-sensitive [82]. Sáez et al. in [82] showed that the SVM has better performance without noise than C4.5. However, the situation is reversed when some noisy data are added. The average performance of C4.5 is better which indicates that the $C_{4.5}$ method globally behaves better with noisy data.

From sensitivity to increase the dataset size, the SVM shows a notable robustness rather than C4.5. Nikam in [83] provided a comparative study of many classification methods including k-NN, NB, artificial neural networks. As conclusions, the k-NN classifier shows sometimes a robustness with regard to noise data, however, the performance of the classifier is significantly influenced by the number of the dimensions used as well as the dataset size and number of records. The NB shows also a great Computational efficiency and classification rate when the dataset is increased.

#### 3) ENSEMBLE WITH IMBALANCED DATA SETS

To deal with imbalanced data set problem, there are four general methods: oversampling, under-sampling, threshold moving and ensemble techniques. The first three techniques did not carry any change to the construction of the classification model. The oversampling and under-sampling techniques cause only a change in the distribution of the data in the training sets, whereas threshold moving effects the final stage of making a decision of classification new data. The ensemble methods can apply, as earliest stated, bagging, boosting and random forest to build a composite model. However, in case of imbalanced data, the oversampling technique is used to split training set into sets with the same positive and negative tuples. On the contrary, the under-sampling tends to decrease the number of negative tuples in the training sets until the number of positive and negative tuples are equals. The threshold moving technique does not involve any sampling. The classification decision is returned based on the output values. The simplest form is as follows: for the tuples that satisfies the minimum threshold, are considered positive, whilst the others are negatives.

### *C. ARABIC AUTHORSHIP ATTRIBUTION*

The authorship attribution problem in languages such as English, Spanish and Chinese are quite properly studied. However, authorship attribution problem on contexts of Arabic texts has been received much less attention [45]. In this section, we present some issues that have a direct impact on AA in context of Arabic. Some challenges that complicate researchers' works in Arabic are highlighted. Next, we present a deeper review of the recent works on Arabic authorship attribution which covers period from 2005 up to 2018.

#### 1) ARABIC CHARACTERISTICS

From the morphological point of view, Arabic is a very rich language. The nature and structure of Arabic words make Arabic very highly derivative and inflective language [46]. In addition, the compound structures of Arabic words add more complexity/ challenges especially for machine translation task where the words should syntactically be regarded as phrases

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.2964952, IEEE Access



rather than single words. The orientation of writing in Arabic, as it is known, is from right-to-left and the letters are connected each other which make Arabic writing differs distinctly from any other Latin-based languages like English, French, etc.

In Arabic, there are a quite small set of productive prefixes and suffixes, however, the number of possible produced words is very high. In many cases, it is enough to change the letter position or its diacritic2 to produce a new word. Although the inflection and diacritics increase the number of words, extracting stylometric features such as vocabulary richness measures might influence [47].

### 2) CHALLENGES IN ARABIC CONTEXT

Arabic is a very rich and challenging language. As stated above, Arabic is very derivative and inflective language [46]. Due to that, several challenges have to deal with before working on authorship attribution task: diacritics, morphological characteristics, structure and orientation of writing, elongation, word length, and word meaning [57].

- *diacritics*, are special marks placed above or below the words. Diacritics play essential role in representing short vowels and changing the word meaning and pronunciation.
- *morphological* characteristics, one of distinguished features of Arabic is a number of produced words from a common root. Such process is known as inflection where the word is derived by adding affixes (prefixes, infixes and suffixes) [5]. Arabic words, in general, are grouped into four groups: word, morpheme, root and stem [58].
- structure and orientation of writing: In Arabic, sentences are written right to left, no upper-case letters, the shape of a letter is changed based on its position in the sentence.
- *elongation*, to emphasize a feeling or meaning, special dashes are inserted between two letters. In addition to that, these dashes play a stylistic role.
- word length and meaning, word, in Arabic, can be: trilateral root, quadrilateral, root, pent-literal root and hex-literal. However, a letter might to play the role of words. The word might to have several different meaning based on the context [57].

## D. MACHINE LEARNING METHODS IN ARABIC AUTHORSHIP ATTRIBUTION

In context of authorship attribution, various methods for attributing Arabic texts have been used. Abbasi and Chen [47] were the first who addressed authorship attribution in Arabic context. Support vector machine (SVM) and C4.5 decision trees were applied on Arabic web forum messages. To cope with the elongation challenge, they proposed a filter which is used to remove elongation from the text. However, number of elongation characters is calculated and it is used later as a feature. In [35], Abbasi and Chen repeated the experiment with the same machine learning methods (SVM and C4.5) and have been applied on Arabic web forum massages however the word roots were extracted by de Roeck and Al-Fares's algorithm [59].

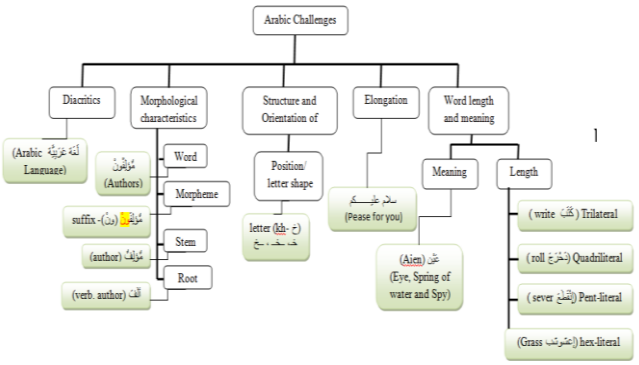


**FIGURE 4: Arabic Characteristics: the leaves present an illustrative example.**

Stamatatos [37] proposed a SVM based model for solving imbalance class problem. The dataset was collected from Alhayat newspaper reports. Ellen and Parameswaran [60] applied k-NN with cosine distance and SVM with two kernel functions to classify 2636 Arabic language forum posts from 9 different website forums. Ouamour and Sayoud [39, 40, 69] used SMO-SVM, linear regression (LR) and multilayered preceptron (MLP) methods for attributing authors of very old Arabic texts. Features such characters n-grams and word n-grams were used as input. The best precision they reached was 80%.

Alam and Kumar [61] also used SVM method to identify author of Arabic articles. Several stylometric features were extracted. They followed the method adapted by Abbasi and Chen [35] to conduct experiments. The best accuracy obtained was 98% when they applied the SVM with all feature combination.

Alwajeeh et al., [42] used Naive Bayes (NB) and SVM classifiers for automatically attributing Arabic articles. The dataset was collected and labeled manually. Through the experiment, the authors examined the effect of stop words and stemming. The findings were interesting: whilst it was expected that applying Khoja stemmer leads to enhance performance of the classifiers, the accuracies are degraded. In addition to that SVM classifier overcomes NB in most subsets. The best accuracy obtained was 99.8%. Howedi and Mohd [62] investigated the effectiveness of NB and SVM classifiers on attributing short historical Arabic texts written by10 different authors. On opposite of the findings in [42], NB exceeds SVM in term of accuracy. In addition, the character-based features give better results than the word-based features. Among the character-based features, the punctuation marks showed a significant improvement in the performance of the classifiers. The accuracies are increased from 67.5% to 74.99%. Otoom et al., [63] introduced a hybrid approach which consists of 27 stylometric features. The ensemble classifier that consists of many decision trees, MultiBoostAB,

[2]Diacritic is special mark which is placed above or below a letter to represent short vowels.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.2964952, IEEE Access



NB, SVM and BayesNet classifiers were employed on dataset with 456 Arabic newspapers instances. The best accuracy was 88 % achieved by MultiBoostAB classifier with the hold-out test and 82% with the cross-validation test.

Sayoud [64] addressed the problem of authorship discrimination. For this purpose, the Quran and Prophet's statements were used. The SMO-SVM, Linear Regression (LR) and Multi-Layer Perceptron (MLP) were employed. All classifiers proved its ability to discriminate the author of the text under consideration with 100% accuracy.

Al-Falahi et al., [65] applied Markov chain classifier on Arabic poetry with 33 different poets belong to the same era. The feature set used by Al-Falahi et al., [65] include a content-specific features such as metre of poem and rhyme. The features were partitioned in testing phase into different sets as follows:

set1: five single features (F1 set- character features, F2 set - word length, F3 set- sentence length, F4 set- first word in sentence and F5 set- rhyme).

set2: Character features + word length feature

set3: Character features + word length + sentence length

set4: Character features + word length + sentence length +first word in sentence

set5: Character features + word length + sentence length +first word in sentence+ rhyme

The best accuracy obtained was 96.7%. They also repeated the experiment with applying NB, SVM and SMO [23]. The features set consists of those features that were used in [65] and the metre of the Arabic poetry. They followed the same methodology as in [65]. The best average accuracy they got was 72,83% when the set of all features was used and SMO was applied.

Bourib and Khennouf [66] addressed the authorship attribution problem when the genre and topic are quite similar. The texts size in the training set was varies from 100 words to 3000 words per a text. The character n-gram and words were employed and SMO-SVM, MLP and LR were used. The findings show that the performance of classifiers are dependent mainly on the text size, on one hand. On the other hand, it is effected by the used features and the classification techniques themselves.

Social media posts were also under consideration. Rabab'ah et al., [67] investigated the effect of authorship attribution classifiers on tweets written in Arabic. The features set consists of: 57 morphological features MF most of which are POS based features and 340 stylometric features SF. The NB, SVM and decision trees were used. The highest accuracy was 68.67% which was achieved by applying SVM classifier on the combined feature sets. In [45], they extended the experiment to include features extracted by bag-of-words approach. Several reduction techniques were used. The findings show that SVM classifier outperforms all of the other methods in term of accuracy and the *SubEval* feature selection technique led to reduce the classifier running time.

TABLE I
PUBLICATIONS ON ARABIC AUTHORSHIP ATTRIBUTION DOMAIN.

| Publication | Domain | Text Size |
|---|---|---|
| [35] [47] [60] | Forum Messages | Short |
| [37] [61] [42] [63] | Newspaper | Long |
| [39] [40] [64] [68] | Historical Texts | Long |
| [66] | | Short-Long |
| [23] [65] | Arabic Poetry | Short |
| [45] [67] | Social Media | Short |
| [44] | Modern Islamic Fatwas | Short |

Sayoud and Hadjadj [68] extended the work in [64]. They proposed to fuse two approaches: feature-based decision fusion which combine three different features, namely character-tetra-gram, word and word bigram; and classifier-based decision fusion which fuses Manhattan centroid, SMO-SVM and MLP classifiers.

Finally, AL-Sarem and Emarra [44] addressed the attribution problem in contexts of modern Islamic fatwā'. In term of attribution classifiers, the locally weighted learning (LWL) classifier, decision tree C4.5, and Random Forest (RF) were used. The features set used by [44] consists of 10 stylometric features. Similar to the work of Al-Ayyoub [45], they investigated the effect of feature selection techniques on the performance of the classifiers. The *SubEval*, *GainRatioEval* and *PCA* were used. The findings show that applying C4.5 method with *SubEval* technique gives the best accuracy obtained is 51.70%.

TABLE II
BEST ACCURACY OBTAINED IN THE PUBLISHED WORKS

| Publication | Features | Classifier | Accuracy |
|---|---|---|---|
| [47] | Lexical +Syntactic +Structural + Content-specific features | SVM | **85.43%** |
| | | C4.5 (DT) | 81.03% |
| [35] | Lexical +Syntactic +Structural + Content-specific features | SVM | **94.83%** |
| | | C4.5 (DT) | 71.93% |
| [37] | Character n-grams | SVM | 93.6% |
| [60] | Lexical + Syntactic features | k-NN | 95% |
| | | SVM | 97% |
| [39] | Lexical features | SMO-SVM | 80% |
| [40][69] | Lexical features | MLP | 70% |
| | | SMO-SVM | **80%** |
| | | LR | 60% |
| [61] | Lexical + Syntactic + Structural + Content-specific + Semantic features | SVM | 98% |
| [42] | Lexical features | NB | 99.4% |
| | | SVM | **99.8%** |
| [62] | Lexical + Character features | SVM | 62.96% |
| | | NB | **71.85%** |
| [63] | Lexical + Syntactic + Structural + Content-specific features | NB | 84.0% |
| | | BayesNet | **86.7%** |
| | | SVM | 79.3% |
| [65] | Lexical + Structural + Content-specific features | Markov chain | 96.67% |
| [23] | Lexical + Structural + Content-specific features | SVM | 71.60% |
| | | SMO | **72.83%** |
| | | NB | 70.37% |
| [66] | Character N-grams + Words | SMO-SVM | - |
| | | MLP / LR | - |
| [45][67] | POS + Stylometric features + Emotional features | SVM | **68.67%** |
| | | DT | 59.83% |
| | | NB | 38.35% |
| [64][68] | Character n-gram + word n-gram + words | SVM | 100% |
| | | MLP | 100% |
| [44] | Lexical features | RF | 24.67% |
| | | C4.5(DT) | **51.70%.** |
| | | LWL | 40.87% |

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.2964952, IEEE Access



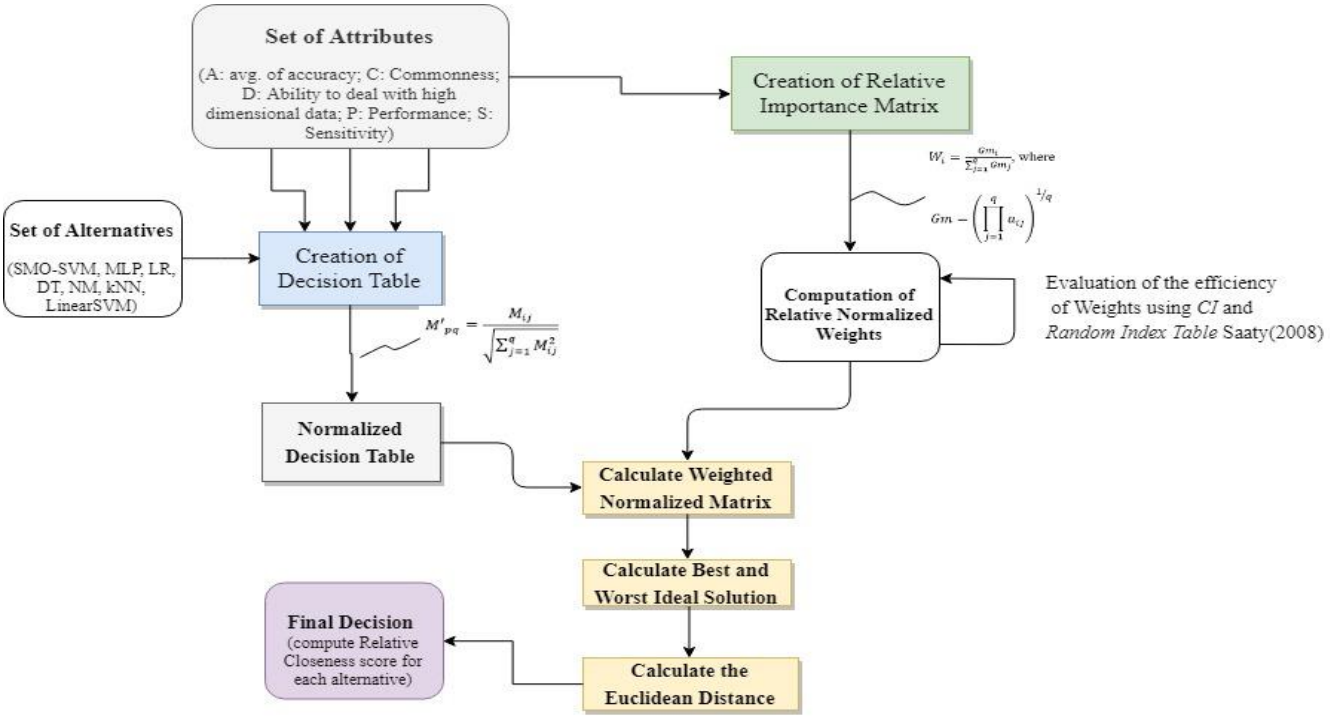


**FIGURE 5: Steps followed to rank classifiers using AHP-TOPSIS**

## III.MATERIALS AND METHODS

At the end of the previous section, we saw that different classifiers have been applied to solve the authorship attribution problem. The SVM with "linear" kernel (LinearSVM) or SMO optimizer for SVM (SMO-SVM), naïve Bayes (NB) are the most commonly used classifiers. Therefore, there is a need to investigate the performance of all mentioned earlier classifiers, which is a time-consuming and lobar intensive. Instead of that, we propose to use Analytic Hierarchy Process (AHP) weighted TOPSIS method to prioritize the classifiers. On the other hand, to avoid topic-oriented biases. Thus, this section is organized as follows: first, we describe the method used to select the base classifiers of ensemble model. Then, we test the effect of ensemble techniques on Arabic authorship attribution based on the best TOPSIS alternative. In addition, the used corpus, the main phases of authorship attribution and the experimental evaluation were also described in details.

### 1) TOPSIS-BASED AHP METHOD

In [70], Saaty introduced (TOPSIS) a technique for order preferences by calculating their similarity to so-called *ideal solution*. TOPSIS is widely used technique for scoring, ranking and choosing the best alternative. Its proficiently ability to handle both subjective and objective attributes is the reason to be one of the most used multi-attribute decision aking method. The TOPSIS method uses AHP to choose the weights for each attribute. So, to employ TOPSIS method (see Fig.5), the following steps should to follow:

**(i) Determine attributes and alternatives**

To make our TOPSIS model more reliable respect selecting authorship attribution classifiers, we propose to use the following attributes:

A- Average accuracies of classifiers stated in published papers, as shown in Table II, to fill the pair-wise comparison matrix of the criteria relating to the goal.

C- Prevalence degree or commonness of use the classifier in publications[3].

D- Ability to deal with high dimensional data.

P- Performance when increase size of training set.

S- Sensitivity to noise data (the scale is assigned based on [71])

In term of alternatives, the Linear SVM, SMO-SVM, NB, MLP, DT, LR and k-NN are taken on consideration.

**(ii) Create decision table**

Our decision table $M$ is presented as a matrix $P \times Q$ where $P$- list of alternatives and $Q$- list of attributes. In the decision table, a row represents the value of each attribute for a respective alternative.

[3] The value can be changed based on number of publications that can be published later

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.2964952, IEEE Access



$$M_{7\times5}=\begin{array}{c} \\ \text{LinearSVM} \\ \text{SMO-SVM} \\ \text{MLP} \\ \text{LR} \\ \text{DT} \\ \text{NB} \\ \text{kNN} \end{array}\begin{array}{c} \begin{array}{ccccc} \text{A} & \text{C} & \text{D} & \text{P} & \text{S} \end{array} \\ \begin{vmatrix} 84.28 & \text{v.high} & \text{v.high} & \text{v.high} & \text{high} \\ 77.61 & \text{v.high} & \text{v.high} & \text{v.high} & \text{v.high} \\ 85 & \text{medium} & \text{high} & \text{high} & \text{medium} \\ 60 & \text{medium} & \text{Low} & \text{high} & \text{v.Low} \\ 68.22 & \text{Low} & \text{Low} & \text{v.Low} & \text{Low} \\ 81.41 & \text{medium} & \text{v.high} & \text{high} & \text{v. Low} \\ 73.5 & \text{Low} & \text{medium} & \text{v.high} & \text{medium} \end{vmatrix} \end{array} \quad (3)$$

To allow dealing with categorical values as given in Eq.3, it is required to convert them into numerical values by using a consensual scale. In our case, we use the scale presented in Table III. It is also necessary to uniform scaling by normalizing $M'_{p\times q}$ as:

$$M'_{pq}=\frac{M_{ij}}{\sqrt{\sum_{j=1}^{q}M_{ij}^2}} \quad (4)$$

Hence, the decision table $M_{p\times q}$ is transformed into $M'_{p\times q}$ as shown in Eq.5.

$$M'_{7\times5}=\begin{array}{c} \\ LinearSVM \\ SMO-SVM \\ MLP \\ LR \\ DT \\ NB \\ kNN \end{array}\begin{array}{c} \begin{array}{ccccc} A & C & D & P & S \end{array} \\ \begin{vmatrix} 0.418336 & 0.542326 & 0.481125 & 0.449013 & 0.496139 \\ 0.382879 & 0.542326 & 0.481125 & 0.4490135 & 0.620174 \\ 0.42211 & 0.325396 & 0.384900 & 0.359211 & 0.372104 \\ 0.29796 & 0.325396 & 0.19245 & 0.359211 & 0.124035 \\ 0.338781 & 0.21693 & 0.19245 & 0.089803 & 0.248069 \\ 0.404282 & 0.325396 & 0.481125 & 0.359211 & 0.124035 \\ 0.365001 & 0.21693 & 0.288675 & 0.449013 & 0.372104 \end{vmatrix} \end{array} \quad (5)$$

TABLE III
CONVERTING SCALE USED IN THIS PAPER

| Attribute value | Very low | Low | Medium | High | Very High |
|---|---|---|---|---|---|
| Scale | 1 | 2 | 3 | 4 | 5 |

**(iii) Assign weights to attributes**

Following Saaty scale [70], importance of attributes is assigned by making a pair-wise comparison which might lack of subjective opinion. Thus, we invite three experts to assign the weights of attributes. The relative importance matrix $A_{q\times q}$ is produced by following the algorithm stated in [72] as:

$$A_{5\times5}=\begin{array}{c} \\ A \\ C \\ D \\ P \\ S \end{array}\begin{array}{c} \begin{array}{ccccc} A & C & D & P & S \end{array} \\ \begin{vmatrix} 1 & 1 & 5 & 3 & 9 \\ 1 & 1 & 3 & 5 & 9 \\ 1/5 & 1/3 & 1 & 5 & 3 \\ 1/3 & 1/5 & 1/5 & 1 & 3 \\ 1/9 & 1/9 & 1/3 & 1/3 & 1 \end{vmatrix} \end{array} \quad (6)$$

The relative normalized weights $W$ are found by computing the geometric mean $Gm$ for each attribute of $A_{q\times q}$ as follows: $W_i=\frac{Gm_i}{\sum_{j=1}^{q}Gm_j}$, (7)

where

$$Gm=\left(\prod_{j=1}^{q}a_{ij}\right)^{1/q} \quad (8)$$

The final normalized relative importance weighting matrix is represented in

$$W=\begin{array}{c} \\ A \\ C \\ D \\ P \\ S \end{array}\begin{array}{c} \\ \begin{vmatrix} 0.3742 \\ 0.3742 \\ 0.1403 \\ 0.0737 \\ 0.0375 \end{vmatrix} \end{array} \quad (9)$$

**(iv) Check for consistency and correctness**

The consistency index *(CI)* is computed by finding the mean of eigenvalues $\Lambda$ as: $CI=(\Lambda-q)/(q-1)$, where: (10)

q- is number of attributes, $\Lambda=\frac{1}{n}\sum_{i=1}^{n}\lambda_i$, $n$- number of alternatives, $\lambda_i=A_j\times W_i$

The eigenvalue $\lambda_i=\begin{vmatrix} 5.3682 \\ 5.0123 \\ 5.8518 \\ 5.6169 \\ 5..1152 \end{vmatrix}$ and $\Lambda=5.39292$ which means that $CI=0.0884$. Based on Saaty's model [70], the acceptable consistency ratio $CR=CI/RI$ should be less 0.1. Random Index value $RI$ is determined based on Table IV. In our case, $CR=0.0884/1.12=0.07963$ which means the model is acceptable.

TABLE IV
RANDOM CONSISTENCY (RI) USED IN SAATY [70]

| Size of matrix | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| Random consistency | 0 | 0 | 0.58 | 0.9 | 1.12 | 1.24 | 1.34 | 1.41 | 1.45 | 1.49 |

**(v) Calculate the weighted normalized matrix**

To obtain the weighted normalized matrix *C*, we have to multiply the normalized matrix $M'$ with the weights $W_i$ obtained by Eq.7

$$C=\begin{array}{c} \\ linearSVM \\ SMO-SVM \\ MLP \\ LR \\ DT \\ NB \\ kNN \end{array}\begin{array}{c} \begin{array}{ccccc} A & C & D & P & C \end{array} \\ \begin{vmatrix} 0.156552 & 0.202953 & 0.067503 & 0.033093 & 0.018626 \\ 0.143283 & 0.202953 & 0.067503 & 0.033093 & 0.023283 \\ 0.157965 & 0.121772 & 0.054003 & 0.026475 & 0.013970 \\ 0.111504 & 0.121772 & 0.027001 & 0.026475 & 0.004657 \\ 0.126781 & 0.081181 & 0.027001 & 0.006619 & 0.009313 \\ 0.151293 & 0.121772 & 0.067503 & 0.026475 & 0.004657 \\ 0.136593 & 0.081181 & 0.040502 & 0.033093 & 0.01397 \end{vmatrix} \end{array} \quad (11)$$

**(vi) Obtain the ideal solution**

The TOPSIS method judges for the beneficial or non-beneficial proposed solutions by finding the best $L^+$ and worst $L^-$ ideal solutions as follows:

$$L^+=\begin{vmatrix} l_1^+ \\ l_2^+ \\ l_3^+ \\ \vdots \\ l_n^+ \end{vmatrix}, \text{ where, } l_i^+= \quad (12)$$

$$\begin{cases} \max(C_{pq}),\ \forall q\in n \\ \min(C_{pq}),\ \forall q\in n' \end{cases} and\ \ p=1\ to\ P$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.2964952, IEEE Access



$$L^- = \begin{vmatrix} l_1^- \\ l_2^- \\ l_3^- \\ \vdots \\ l_n^+ \end{vmatrix}, \text{ where, } l_i^+ = \begin{cases} \min(C_{pq}), \ \forall q \in n \\ \max(C_{pq}), \ \forall q \in n' \end{cases} and \ p = 1 \ to \ P \quad (13)$$

Regarding the alternatives listed earlier, the average accuracy of classifier $A$, commonness indicator C, high dimensionality indicator D and the performance sensitivity P are considered as an entry of the positive ideal solution, whereas the sensitivity for noise data S is an entry of negative ideal solution. The ideal solutions obtained from matrix $C$ is represented as follows:

| | $L^+$ | $L^-$ |
|---|---|---|
| $A$ | 0.157965 | 0.111504 |
| $C$ | 0.202953 | 0.081181 |
| $D$ | 0.067503 | 0.027001 |
| $P$ | 0.033093 | 0.006619 |
| $S$ | 0.004657 | 0.023283 |

**(vii) Calculate the Euclidean distance**

The Euclidean distance is computed to measure how a solution is far from the ideal one. It is calculated as follows:

$$E_p^+ = \sqrt{\sum_{i=1}^{q} (C_{pi} - L_i^+)^2} \quad (14)$$

$$E_p^- = \sqrt{\sum_{i=1}^{q} (C_{pi} - L_i^-)^2} \quad (15)$$

So, the Euclidean distance for both $E_p^+$ and $E_p^-$ is:

| | $E^+$ | $E^-$ |
|---|---|---|
| $Linear\ SVM$ | 0.212643 | 0.166827 |
| $SMO-SVM$ | 0.210130 | 0.165138 |
| $MLP$ | 0.223924 | 0.128401 |
| $LR$ | 0.247816 | 0.141500 |
| $DT$ | 0.262721 | 0.142304 |
| $NB$ | 0.224345 | 0.131430 |
| $kNN$ | 0.237649 | 0.122780 |

**(viii) Rank the alternatives**

The final step in TOPSIS is determine how an alternative is closer to the ideal. For this, we calculate closeness scores $S$ , then rank them in descending order as follows:

$$S_p^+ = \frac{E_p^-}{(E_p^+ + E_p^-)} \Rightarrow S_p^+ = \begin{matrix} Linear\ SVM \\ SMO-SVM \\ MLP \\ LP \\ DT \\ NB \\ kNN \end{matrix} \begin{vmatrix} 0.439631 \\ 0.440052 \\ 0.364438 \\ 0.363457 \\ 0.351346 \\ 0.369419 \\ 0.340649 \end{vmatrix} \quad (16)$$

$$\Rightarrow S_p^+ = \begin{matrix} Linear\ SVM \\ SMO-SVM \\ NB \\ MLP \\ LR \\ DT \\ kNN \end{matrix} \begin{vmatrix} 0.439631 \\ 0.440052 \\ 0.369419 \\ 0.364438 \\ 0.363457 \\ 0.351346 \\ 0.340649 \end{vmatrix}$$

The alternative with highest closeness score is considered as the best preferred alternative. In our case, the SMO classifier turns out to be the best preferred classifiers among those considered in this work followed by SVM and Naive Bayes classifiers.

### 2) CORPUS

Absence a benchmark dataset of authorship attribution on Arabic makes additional difficulties for evaluating attribution classifiers' performance. Most of publications on Arabic authorship attribution domain use different dataset (see Table II). Not far of that, our dataset was gathered from Dar Al-ifta AL Misriyyah[4] website. The website contains a huge set of fatwas which are written in several language including Arabic and 9 other languages. Typically, the fatwa follows a well-defined structure. Apart of that, we deal with it as a regular textual content. We limit our corpus to only those fatwas written in Arabic. To extract the fatwas' content from the website, the OctoParse 7.0.2 web scraping tool5. The Octoparse is an easy configurable visual tool. It allows to run an extraction on the cloud as well as on the local machine. The scraped data can be exported in TXT, CSV, HTML or Excel formats. The main challenge was in scrapping the right data. Thus, first we explore the website page manually to group the similar pages and insure that the page contains required texts, then feed the scrapper the right URL. The output was an Excel sheet with some useful information: (i) fatwa's title: a given title which describes its message briefly; (ii) fatwa's date gives information about the period when the fatwa was published; (iii) mofti's name is the person or Islamic scholar who interprets and expounds the law; (iv) fatwa's question which is posed by a questioning person. It contains a lot of helpful information which aims mofti to drive his opinion and final decision; and (v) the fatwa's answer which contains the details of the scholar's. Among of the aforementioned information, mufti answer (fatwa answer) is the more important. The fatwa answer might be varying in length dependent on the nature of fatwa type and the detailed explanation given by the mofti. One thing should to mentioned here that the corpus can be unbalanced regarding the distribution of fatwas per author

[4] http://www.dar-alifta.org/Foreign/default.aspx?LangID=2&Home=1

[5] https://www.octoparse.com/download

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.2964952, IEEE Access



(Mofti). Thus, the training set has to managed before employing an attribution classifier.

### 3) DATA PRE-PROCESSING

Before doing any preprocessing, the corpus is firstly divided into two sub-corpuses. Current step allows us to investigate impact of training set size on the performance of the SMO classifier: (i) balanced sub-corpus $\mathfrak{B}$ in which the number of fatwas per each mofti is equal, and (ii) unbalanced sub-corpus $\mathfrak{U}$ where the distribution of texts per author is different. In addition, each sub-corpus is also grouped into sets of texts size. The last grouping also necessary to test the effect of increasing the training set size on the overall performance. As the dataset organized, others necessary preprocessing steps are performed:

- Normalization: to avoid any variation in Arabic word representation, we follow the steps stated in [5] [73]:
  - change the letters ( إ ), ( أ ), ( آ ) and ( ؤ ) to ( ا ).
  - change the letters (ئ ) and ( ى) to (ي)
  - change the letter (ة) to ( ه)
  - convert text encoding format to CP1256.
- Function words and non-letter removal: unlike text mining tasks, we kept these features in order to provide more authorial evidence [5].
- Stemming: to find the root of the words, we proposed to use the Khojah's stemmer[6].

To deal with the above preprocessing steps, we used the Alwajeeh's ArabicSF tool[7] for both sub-corpora before extracting attribution features.

### 4) FEATURE EXTRACTION

Since the instance-based approach [16] suggested to treat each text in the training set individually, the result of the feature extraction step is a vector of numerical values. Our features set consists of: (i) 392 features 335 out of them features extracted by the Alwajeeh's Arabic SF tool, and 56 morphological features extracted by MADAMIRA[8] tool, and (ii) 350 distinct words extracted by the WEKA[9] tool.

TABLE V
FEATURES OBTAINED BY ALWAJEEH'S ARABIC SF TOOL [45]

| Feature | Type | Description |
|---|---|---|
| ASFM1 | Character-based lexical features | Total number of characters (C) |
| ASFM2 | | Number of letters/C |
| ASFM3 | | Number of digits/C |
| ASFM4 | | Number of white-spaces/C |
| ASFM5 | | Number of tab spaces/C |
| ASFM6 | | Number of elongations |
| ASFM7 | | Number of multiple elongations |
| ASFM8- ASFM15 | | Number of diacritics |
| ASFM16- ASFM39 | | Number of special characters/C |
| ASFM40- ASFM75 | | Number of individual letters/C |
| ASFM76 | Word-based lexical features | Total number of words N |
| ASFM77 | | Average word length |
| ASFM78 | | Number of different (unique) words/N |
| ASFM79 | | Number of long words/N |
| ASFM80 | | Number of short words/N |
| ASFM81 | | Hapax legomena/N |
| ASFM82 | | Hapax dislegomena/N |
| ASFM83- ASFM97 | | Word length frequency distribution |
| ASFM98 | | Number of "digit" words/N |
| ASFM99 | | Number of words with repeated letters |
| ASFM100 | | Yule's K measure |
| ASFM101 | | Simpson's D measure |
| ASFM102 | | Sichel's S measure |
| ASFM103 | | Honore's R measure |
| ASFM104 | | Entropy measure |
| ASFM105- ASFM117 | Syntactic features | Number of different punctuation signs/C: single quotes, commas, periods, colons, semi-colons, question marks, exclamation marks, Double quotes, multiple question marks, multiple exclamation marks, and ellipsis. |
| ASFM118 | Structural features | Total number of lines (L) |
| ASFM119 | | Total number of sentences (S) |
| ASFM120 | | Total number of paragraphs (P) |
| ASFM121 | | Average number of S/ P |
| ASFM122 | | Average number of words/P |
| ASFM123 | | Average number of C/ P |
| ASFM124 | | Average number of words per sentence |
| ASFM125 | | Number of title words |
| ASFM126 | | Title length in characters |
| ASFM127 | | Title length in characters |
| ASFM128 | | Number of blank lines |
| ASFM129 | | Average length of non-blank line in characters |
| ASFM1230 | | Number of short phrases |
| ASFM131- ASFM142 | | Sentences length frequency distribution |
| ASFM143- ASFM335 | Content-specific Features | Function words |

TABLE VI
FEATURES OBTAINED BY MADAMIRA TOOL [45]

| Feature | Type | Description |
|---|---|---|
| ASFM336 | POS features | Number of nouns |
| ASFM337 | | Number of proper nouns |
| ASFM338- ASFM341 | | Number of adjectives |
| ASFM3242-ASFM345 | | Number of adverbs |
| ASFM346- ASFM350 | | Number of Pronouns |
| ASFM351- ASFM352 | | Number of verbs |
| ASFM353-ASFM362 | | Number of particles |
| ASFM363 | | Number of prepositions |
| ASFM364 | | Number of abbreviations |
| ASFM365 | | Number of punctuation |
| ASFM366-ASFM367 | | Number of conjunctions |
| ASFM368 | | Number of interjections |
| ASFM369 | | Number of digital numbers |
| ASFM370 | | Number of foreign letters |
| ASFM371 | Aspect features | Number of commands |
| ASFM372 | | Number of imperfective |
| ASFM373 | | Number of perfective |
| ASFM374 | Case features | Number of nominative |
| ASFM375 | | Number of accusative |
| ASFM376 | | Number of genitive |
| ASFM377 | Gender features | Feminine |
| ASFM378 | | Masculine |
| ASFM379 | Mood features | Indicative |
| ASFM380 | | Jussive |
| ASFM381 | | Subjunctive |
| ASFM382 | Number features | Number of singular words |
| ASFM383 | | Number of plural words |

[6] http://zeus.cs.pacificu.edu/shereen/research.htm
[7] https://github.com/AAlwajeeh/ArabicSF
[8] https://camel.abudhabi.nyu.edu/madamira/
[9] https://www.cs.waikato.ac.nz/ml/weka/downloading.html

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.2964952, IEEE Access



| | | |
|---|---|---|
| ASFM384 | | Number of dual words |
| ASFM385 | Grammatical person features | 1st person |
| ASFM386 | | 2nd person |
| ASFM387 | | 3rd person |
| ASFM388 | State features | Number of indefinite |
| ASFM389 | | Number of definitive |
| ASFM390 | | Number of construct/poss/idafa |
| ASFM391 | Voice features | Active voice |
| ASFM392 | | Passive voice |

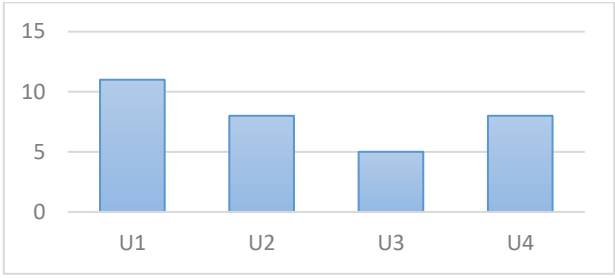


**FIGURE 6: Distribution of number of authors per imbalanced dataset**

#### 5) ENSEMBLE METHODS

As stated earlier, the SMO-SVM is assigned as a base classifier of the ensemble method. The ensemble method is trained and tested within WEKA 3.6.12 on a personal computer with an Intel Core(TM) i7-4600U CPU @2.70GHz CPU, a 8-Gbyte RAM and a 64-bit Windows 8 operating system. In addition, the Cross-validation was employed in 10-folds version and accuracy, precision, recall and F1-score are used to measure the effectiveness of the attribution model. To answer the second posed question, the features were partitioned into three different sets and the classifier is trained and tested on four different groups size as follows:

**Features partition**

set1: the Arabic Stylometric Features extracted by ArabicSF tool and MADAMIRA (*ASFMs*).

set2: the distinct words extracted by applying the bag-of-word method within WEKA environment (*DWs*)

set3: combination of both *ASFMs* and *DWs* features (*ASFMs+DWs)*

**Training Set Size: Balanced group**

The training set is partitioned into subsets with 50,100, 200 and 300 texts per author. We denote them $\beta_1$, $\beta_2$, $\beta_3$ and $\beta_4$ respectively. In addition, the amount of words within a text does not take in consideration.

**Training Set Size: Unbalanced group**

group1($U_1$): The training set has instances of 11 authors. It varies from 11 fatwas per author to 975. The number of the words within a fatwa varies between very short text (31words per text) and quit long text (400 words per text).

group2 ($U_2$): The training set has instances of eight authors. The number of texts are between 13 and 401 per author. The number of the words within a fatwa is between 400 words per a fatwa and 800 words.

group3 ($U_3$): The training set has instances of five authors. The size is quite small. The distribution of instances per authors varies from 7 fatwas per an author to 80. We limit amount of words within the text to be between 800 words per a fatwa and 1200 words.

group3 ($U_4$): The training set has instances of eight authors. The size is also quite small with quit long fatwa text. The training set contains those texts whose lengths exceed 1200 words per a texts.

## IV. RESULTS AND DISCUSSION

### A. FEATURE-BASED LEVEL

To investigate the performance of using different stylometric features (*ASFMs, DWs and ASFMs+DWs)*, Table VII-XV summarize the results obtained by the two ensemble methods on balanced and imbalanced datasets in terms of the accuracy, recall, precision and F1-score. The results shown that the combination set of features (ASFMs+DWs) obtained the best performance using Bagging and AdaBoost methods for balanced datasets, except for dataset subset β1. The dataset size of β1 is only 50 texts per author, which makes the DW features more effective than *ASFMs* that may include more zeros in the feature vector. For the imbalanced datasets, the ASFMs obtained the best results (5 out of 8 cases). Similar to the case of β1, the DW features obtained better results for the dataset subset U1.

TABLE VII
RESULT OF DIFFERENT ENSEMBLE TECHNIQUES ON BALANCED DATASET.

| Dataset | Classifier | Feature set | Acc. | Prec. | Recall | F1-score |
|---|---|---|---|---|---|---|
| $\beta_1$ | Bagging | *ASFM* | 0.4927 | 0.526 | 0.493 | 0.490 |
| | | *DW* | **0.7273** | **0.733** | **0.727** | **0.729** |
| | | *ASFM+DW* | 0.7089 | 0.718 | 0.701 | 0.709 |
| | AdaBoost | *ASFM* | 0.4618 | 0.487 | 0.462 | 0.455 |
| | | *DW* | 0.7127 | 0.715 | 0.713 | 0.713 |
| | | *ASFM+DW* | **0.7900** | **0.789** | **0.791** | **0.789** |

TABLE VIII
RESULT OF DIFFERENT ENSEMBLE TECHNIQUES ON BALANCED DATASET.

| Dataset | Classifier | Feature set | Acc. | Prec. | Recall | F1-score |
|---|---|---|---|---|---|---|
| $\beta_2$ | Bagging | *ASFM* | 0.8050 | 0.805 | 0.805 | 0.802 |
| | | *DW* | 0.8517 | 0.852 | 0.852 | 0.851 |
| | | *ASFM+DW* | **0.8789** | **0.878** | **0.878** | **0.878** |
| | AdaBoost | *ASFM* | 0.7900 | 0.787 | 0.790 | 0.788 |
| | | *DW* | 0.8517 | 0.852 | 0.852 | 0.851 |
| | | *ASFM+DW* | **0.8720** | **0.872** | **0.872** | **0.872** |

TABLE IX
RESULT OF DIFFERENT ENSEMBLE TECHNIQUES ON BALANCED DATASET.

| Dataset | Classifier | Feature set | Acc. | Prec. | Recall | F1-score |
|---|---|---|---|---|---|---|
| $\beta_3$ | Bagging | *ASFM* | 0.7060 | 0.708 | 0.706 | 0.706 |
| | | *DW* | 0.8330 | 0.833 | 0.833 | 0.833 |
| | | *ASFM+DW* | **0.8442** | **0.851** | **0.833** | **0.842** |
| | AdaBoost | *ASFM* | 0.7060 | 0.706 | 0.710 | 0.707 |
| | | *DW* | 0.8180 | 0.818 | 0.817 | 0.817 |
| | | *ASFM+DW* | **0.8910** | **0.893** | **0.891** | **0.892** |

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.2964952, IEEE Access



TABLE X
RESULT OF DIFFERENT ENSEMBLE TECHNIQUES ON BALANCED DATASET.

| Dataset | Classifier | Feature set | Acc. | Prec. | Recall | F1-score |
|---|---|---|---|---|---|---|
| $\beta_4$ | Bagging | *ASFM* | 0.9900 | 0.990 | 0.990 | 0.990 |
| | | *DW* | 0.9950 | 0.995 | 0.995 | 0.995 |
| | | *ASFM+DW* | **0.9979** | **0.961** | **0.997** | **0.979** |
| | AdaBoost | *ASFM* | 0.9900 | 0.990 | 0.990 | 0.990 |
| | | *DW* | 0.9950 | 0.995 | 0.995 | 0.995 |
| | | *ASFM+DW* | **0.9983** | **0.999** | **0.998** | **0.998** |

TABLE XI
RESULT OF DIFFERENT ENSEMBLE TECHNIQUES ON IMBALANCED DATASET.

| Dataset | Classifier | Feature set | Acc. | Prec. | Recall | F1-score |
|---|---|---|---|---|---|---|
| $U_1$ | Bagging | *ASFM* | 0.7447 | 0.745 | 0.721 | 0.722 |
| | | *DW* | 0.8148 | 0.815 | 0.814 | 0.814 |
| | | *ASFM+DW* | **0.8620** | **0.865** | **0.859** | **0.861** |
| | AdaBoost | *ASFM* | 0.7485 | 0.749 | 0.747 | 0.745 |
| | | *DW* | **0.8037** | **0.804** | **0.799** | **0.800** |
| | | *ASFM+DW* | 0.7079 | 0.713 | 0.703 | 0.708 |

TABLE XII
RESULT OF DIFFERENT ENSEMBLE TECHNIQUES ON IMBALANCED DATASET.

| Dataset | Classifier | Feature set | Acc. | Prec. | Recall | F1-score |
|---|---|---|---|---|---|---|
| $U_2$ | Bagging | *ASFM* | **0.8569** | **0.857** | **0.858** | **0.854** |
| | | *DW* | 0.8153 | 0.815 | 0.801 | 0.803 |
| | | *ASFM+DW* | 0.8319 | 0.836 | 0.829 | 0.832 |
| | AdaBoost | *ASFM* | **0.8353** | **0.835** | **0.837** | **0.834** |
| | | *DW* | 0.7554 | 0.755 | 0.719 | 0.733 |
| | | *ASFM+DW* | 0.7225 | 0.724 | 0.726 | 0.725 |

TABLE XIV
RESULT OF DIFFERENT ENSEMBLE TECHNIQUES ON IMBALANCED DATASET.

| Dataset | Classifier | Feature set | Acc. | Prec. | Recall | F1-score |
|---|---|---|---|---|---|---|
| $U_3$ | Bagging | *ASFM* | **0.8400** | **0.840** | 0.796 | 0.816 |
| | | *DW* | 0.8160 | 0.816 | 0.802 | 0.798 |
| | | *ASFM+DW* | 0.8234 | 0.824 | 0.827 | 0.825 |
| | AdaBoost | *ASFM* | **0.8241** | **0.824** | **0.816** | **0.819** |
| | | *DW* | 0.8160 | 0.816 | 0.798 | 0.800 |
| | | *ASFM+DW* | 0.8104 | 0.812 | 0.809 | 0.810 |

TABLE XV
RESULT OF DIFFERENT ENSEMBLE TECHNIQUES ON IMBALANCED DATASET.

| Dataset | Classifier | Feature set | Acc. | Prec. | Recall | F1-score |
|---|---|---|---|---|---|---|
| $U_4$ | Bagging | *ASFM* | 0.6774 | 0.677 | 0.593 | 0.630 |
| | | *DW* | 0.6613 | 0.661 | 0.584 | 0.619 |
| | | *ASFM+DW* | **0.6783** | **0.675** | **0.693** | **0.684** |
| | AdaBoost | *ASFM* | **0.6210** | **0.621** | **0.566** | **0.587** |
| | | *DW* | 0.5806 | 0.581 | 0.563 | 0.571 |
| | | *ASFM+DW* | 0.5510 | 0.556 | 0.546 | 0.550 |

For balanced datasets, the tables show that the AdaBoost classifier, in most cases, gives the highest performance. It achieves the best accuracy with 99.83%. In addition, the results show that the performance of the classifiers is effected positively with decreasing the number of the authors in the dataset. As a conclusion of that, we recommend to use the Adaboost method for solving the authorship verification problem for balanced datasets. However, for imbalanced datasets the performance of Bagging method outperformed the Adaboost method using all datasets subsets. In addition, the results shown that when the size of imbalanced dataset increased, the performance of Bagging classifier decreased.

### B. CLASSIFIER-BASED LEVEL

Table XVI reports the p-values produced by the Wilcoxon singed-rank test for comparing the significant difference between Bagging and Adaboost classifiers. The reported p-values are higher than the significant level of 0.05, the null hypothesis, that the metrics values are the same, is accepted for all metrics.

TABLE XVI
P-VALUES OBTAINED USING THE WILCOXON SIGNED-RANK TEST FOR BALANCED DATASETS

| metric | Bagging Vs Adaboost |
|---|---|
| Accuracy | 0.8334 |
| Precision | 1 |
| Recall | 0.8127 |
| F-score | 0.9056 |

Table XVII summarizes the median and mean values computed for all Balanced dataset for each ensemble classifiers. In most cases, the Bagging classifier achieved slightly higher median scores compared with Adaboost and this interprets why the p-values are higher than 0.05. These reported median and median scores do not show any superiority of one classifier over the other and this may attribute to the advantages of over-sampling that mitigate the problem of data sparseness.

TABLE XVII
MEAN AND MEDIAN OF BALANCED DATASETS

| | | Accuracy | Precision | Recall | F-score |
|---|---|---|---|---|---|
| Bagging | Median | 0.8386 | 0.842 | 0.833 | 0.8375 |
| | Mean | 0.819217 | 0.820833 | 0.8175 | 0.817 |
| AdaBoost | Median | 0.83485 | 0.835 | 0.8345 | 0.834 |
| | Mean | 0.823042 | 0.82525 | 0.823417 | 0.82225 |

On the other hand, Table XVIII shows the p-values obtained by the Wilcoxon singed-rank test after comparing the scores attained by both classifiers. The reported p-values are less than the significant level of 0.05, the null hypothesis, that the metrics values are the same, is rejected for all metrics.

TABLE XVIII
P-VALUES OBTAINED USING THE WILCOXON SIGNED-RANK TEST FOR IMBALANCED EXPERIMENTS

| | Bagging Vs Adaboost |
|---|---|
| Accuracy | 0.005099 |
| Precision | 0.005099 |
| Recall | 0.03092 |
| F-score | 0.01611 |

Table XIX shows the median and mean values computed for all Imbalanced dataset for each classifier. In all cases, the Bagging classifier achieved clearly higher median scores compared with Adaboost. These reported median and median scores show a clear dominance of Bagging classifier over the Adaboost and this proved the advantages of bagging classifier in dealing with sparse training data.

TABLE XIX
MEAN AND MEDIAN OF IMBALANCED

| | | Accuracy | Precision | Recall | F-score |
|---|---|---|---|---|---|
| Bagging | Median | 0.81565 | 0.8155 | 0.8015 | 0.8085 |
| | Mean | 0.785167 | 0.7855 | 0.76475 | 0.7715 |
| AdaBoost | Median | 0.75195 | 0.752 | 0.7365 | 0.739 |
| | Mean | 0.731367 | 0.7325 | 0.719083 | 0.7235 |

## V.CONCLUSION AND FUTURE WORK

Authorship Attribution (AA) problem in Arabic language has been addressed in quite few studies and several analysis methods were applied to tackle the issue. However , the performance of these methods needs to be improved. This work distinguishes from the existing works in employing the ensemble techniques which have not been investigated for ALAA. In addition, the TOPSIS method has been used for scoring, ranking and choosing the best alternative base classifier. In order to make the TOPSIS model more reliable for selecting authorship attribution base classifiers, several attributes were used: (i) average accuracies of classifiers stated in published paper, (ii) prevalence degree or commonness of use the classifier in publications, (iii) ability to deal with high dimensional data, (iv) performance and (v) sensitivity to noise data. Indeed, adding others attributes can lead to enhance the TOPSIS method. As a conclusion, the SMO-SVM classifier has been chosen as a base classifier of ensemble methods.

On the other hand, two types of features have been used: 397 stylometric features (ASFMs) which was extracted by Alwajeeh's ArabicSF tool and MADAMIRA tool and 350 distinct words extracted by the WEKA tool. These features were extracted from Arabic texts (Islamic fatwas) collected from Dar Al-ifta AL Misriyyah website using the OctoParse 7.0.2 web scraping tool.

Then, Bagging and AdaBoost methods have been applied. The performance of the methods was examined for balanced and unbalanced training datasets. The results showed different characteristics for the ensemble methods. The AdaBoost methods obtained the highest accuracy for the balanced dataset, whereas the Bagging methods obtained the highest accuracy with unbalanced set. The findings also showed that fusing the ASFMs features and DWs features yielded the best results.

In future work, new attributes will be researched and examined using the TOPSIS method and other ensemble methods will be investigated for ALAA.

## REFERENCES

1. Mendelhall, T.C (1887). The characteristic curves of composition, Science, IX, 237–249.
2. Zipf GK. The psycho-biology of language. Houghton, Mifflin; 1935.
3. Yule, G.U. (1938). On sentence-length as a statistical characteristic of style in prose, with application to two cases of disputed authorship. Biometrika, 30, 363-390.
4. Mosteller, Frederick and a. D. Wallace, "Inference and disputed authorship: The Federalist," 1964.
5. Altheneyan, A. S., & Menai, M. E. B. (2014). Naïve Bayes classifiers for authorship attribution of Arabic texts. Journal of King Saud University-Computer and Information Sciences, 26(4), 473-484.
6. Xavier Puig, Martí Font & Josep Ginebra (2016): A unified approach to authorship attribution and verification, The American Statistician, DOI: 10.1080/00031305.2016.1148630.
7. Krsul I, Spafford EH. Authorship analysis: Identifying the author of a program. Computers & Security, 1997, 16(3): 233±257. https://doi.org/10.1016/S0167-4048(97)00005-9.
8. Longstaff TA, Schultz EE. Beyond preliminary analysis of the WANK and OILZ worms: A case study of malicious code. Computers & Security, 1993, 12(1): 61±77. https://doi.org/10.1016/0167-4048(93)90013-U.
9. Spafford EH, Weeber SA. Software forensics: Can we track code to its authors?. Computers & Security, 1993, 12(6): 585±595. https://doi.org/10.1016/0167-4048(93)90055-A
10. De Vel, O. (2000, August). Mining e-mail authorship. In Proc. Workshop on Text Mining, ACM International Conference on Knowledge Discovery and Data Mining (KDD'2000).
11. S. Argamon, M. Saric and a. S. Stein, "Style Mining of Electronic Message for Multiple Authorship Discrimination: First Results," in ninth ACM SIGKDD international conference, New York, 2003.
12. Stearns, L. (1992). Copy wrong: Plagiarism, process, property, and the law. Cal. L. Rev., 80, 513.
13. Rudman, J. (1997). The state of authorship attribution studies: Some problems and solutions. Computers and the Humanities, 31(4), 351-365.
14. Singhe, S., & Tweedie, F. J. (1995). Neural networks and disputed authorship: New challenges.
15. Martin, B. (1994). Plagiarism: a misplaced emphasis. Journal of Information Ethics, 3(2), 36.
16. Stamatatos, E., 2009. A survey of modern authorship attribution methods. J. Am. Soc. Inf. Sci. Technol. 60 (3), 538–556. http://dx.doi.org/10.1002/asi.21001.
17. Markov, I., Baptista, J., & Pichardo-Lagunas, O. (2017). Authorship Attribution in Portuguese Using Character N-grams. Acta Polytechnica Hungarica, 14(3).
18. Posadas-Durán, J. P., Gómez-Adorno, H., Sidorov, G., Batyrshin, I., Pinto, D., & Chanona-Hernández, L. (2017). Application of the distributed document representation in the authorship attribution task for small corpora. Soft Computing, 21(3), 627-639.
19. Pan L., Gondal I., Layton R. (2017) Improving Authorship Attribution in Twitter Through Topic-Based Sampling. In: Peng W., Alahakoon D., Li X. (eds) AI 2017: Advances in Artificial Intelligence. AI 2017. Lecture Notes in Computer Science, vol 10400. Springer, Cham.
20. Dauber, E., Overdorf, R., & Greenstadt, R. (2017, June). Stylometric Authorship Attribution of Collaborative Documents. In International Conference on Cyber Security Cryptography and Machine Learning (pp. 115-135). Springer, Cham.
21. Marchenko O., Anisimov A., Nykonenko A., Rossada T., Melnikov E. (2017) Authorship Attribution System. In: Frasincar F., Ittoo A., Nguyen L., Métais E. (eds) Natural Language Processing and Information Systems. NLDB 2017. Lecture Notes in Computer Science, vol 10260. Springer, Cham.
22. Claude, F., Galaktionov, D., Konow, R., Ladra, S., & Pedreira, Ó. (2017). Competitive Author Profiling Using Compression-Based Strategies. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 25(Suppl. 2), 5-20.
23. Al-Falahi, A., Ramdani, M., & Mostafa, B. (2017). Machine Learning for Authorship Attribution in Arabic Poetry. International Journal of Future Computer and Communication, 6(2), 42.
24. Szwed, P. (2017, May). Authorship attribution for polish texts based on part of speech tagging. In International Conference: Beyond Databases, Architectures and Structures (pp. 316-328). Springer, Cham.
25. Zhao, Y., Zobel, J., & Vines, P. (2006, October). Using relative entropy for authorship attribution. In Asia Information Retrieval Symposium (pp. 92-105). Springer, Berlin, Heidelberg.
26. Pillay, S.R., Solorio, T., 2010. Authorship attribution of web forum posts. eCrime Researchers Summit (eCrime), IEEE, pp. 1–7. doi:10.1109/ecrime.2010.5706693.
27. Baron, G. (2014). Influence of data discretization on efficiency of Bayesian classifier for authorship attribution. Procedia Computer Science, 35, 1112-1121.
28. Paul, P. P., Sultana, M., Matei, S. A., & Gavrilova, M. (2018). Authorship disambiguation in a collaborative editing environment. Computers & Security.
29. Akimushkin, C., Amancio, D. R., & Oliveira Jr, O. N. (2018). On the role of words in the network structure of texts: application to authorship attribution. Physica A: Statistical Mechanics and its Applications, 495, 49-58.
30. Lahiri, S., & Mihalcea, R. (2013). Authorship attribution using word network features. arXiv preprint arXiv:1311.2978.
31. Wang, L. Z. (2017). News authorship identification with deep learning.

32. Giraud, F. M., & Artières, T. (2012). Feature Bagging for Author Attribution. In CLEF (Online Working Notes/Labs/Workshop).
33. Srinivasan, L., & Nalini, C. (2017). An improved framework for authorship identification in online messages. Cluster Computing, 1-10.
34. Ekinci, E., & Takçı, H. (2013). Comparing Ensemble Classifiers: Forensic Analysis of Electronic Mails.
35. Abbasi A, Chen H (2005b) Applying authorship analysis to extremist group web forum messages. IEEE Intell Syst 20(5):67–75.
36. Otoom AF, Abdullah EE, Jaafer S, Hamdallh A, Amer D (2014) Towards author identification of arabic text articles. In: Information and Communication Systems (ICICS), 2014 5th International Conference on, IEEE, pp 1–4.
37. Stamatatos E (2008) Author identification: Using text sampling to handle the class imbalance problem. Inf Process Manag 44(2):790–799.
38. Shaker K, Corne D (2010) Authorship attribution in Arabic using a hybrid of evolutionary search and linear discriminant analysis. In: 2010 UK Workshop on Computational Intelligence (UKCI), IEEE, pp 1–6.
39. Ouamour S, Sayoud H (2012) Authorship attribution of ancient texts written by ten Arabic travelers using a SMO-SVM classifier. In: 2012 International Conference on Communications and Information Technology (ICCIT), IEEE, pp 44–47.
40. Ouamour S, Sayoud H (2013) Authorship attribution of short historical Arabic texts based on lexical features. In: 2013 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), IEEE, pp 144–147.
41. Baraka RS, Salem S, Hussien MA, Nayef N, Shaban WA (2014) Arabic text author identification using support vector machines. J Adv Comput Sci Technol Res 4(1):1–11.
42. Alwajeeh A, Al-Ayyoub M, Hmeidi I (2014) On authorship authentication of arabic articles. In: Information and Communication Systems (ICICS), 5th International Conference on, IEEE, pp 1–6.
43. Al-Ayyoub A Mahmoud Alwajeeh, Hmeidi I (2016) An extensive study of authorship authentication of Arabic articles. Int J Web Inf Syst (IJWIS)
44. Al-Sarem, M., & Emara, A. H. (2018, June). Analysis the Arabic Authorship Attribution Using Machine Learning Methods: Application on Islamic Fatwā. In International Conference of Reliable Information and Communication Technology (pp. 221-229). Springer, Cham.
45. Al-Ayyoub, M., Jararweh, Y., Rabab'ah, A. and Aldwairi, M., 2017. Feature extraction and selection for Arabic tweets authorship authentication. Journal of Ambient Intelligence and Humanized Computing, 8(3), pp.383-393.
46. Yousif, J. H., & Sembok, T. M. T. (2008, August). Arabic part-of-speech tagger based Support Vectors Machines. In Information Technology, 2008. ITSim 2008. International Symposium on (Vol. 3, pp. 1-7). IEEE.
47. Abbasi, A., Chen, H., 2005a. Applying authorship analysis to Arabic web content. In: Kantor, P., Muresan, G., Roberts, F., Zeng, D.D., Wang, F.-Y., Chen, H., Merkle, R.C. (Eds.), Intelligence and Security Informatics, vol. 3495. Springer-Verlag, Berlin, Heidelberg, pp. 183–197.
48. Baron, G. (2017, June). Analysis of multiple classifiers performance for discretized data in authorship attribution. In International Conference on Intelligent Decision Technologies (pp. 33-42). Springer, Cham.
49. Pearson, K., 1901. On lines and planes of closest fit to systems of points in space. Philos. Mag. 2 (11), 559–572.
50. Fisher, R.A., 1936. The use of multiple measurements in taxonomic problems. Ann. Eugenics 7 (2), 179–188.
51. Burrows, J., 2002. ''Delta": a measure of stylistic difference and a guide to likely authorship. Literary Linguist. Comput. 17 (3), 267– 287. http://dx.doi.org/10.1093/llc/17.3.267.
52. Keselj, V., Peng, F., Cercone, N., Thomas, C., 2003. N-gram-based author profiles for authorship attribution. Computat. Linguist. 3, 255–264, Doi: 10.1.1.9.7388.
53. Juola, P., 2005. A controlled-corpus experiment in authorship identification by cross-entropy. Literary Linguist. Comput. 20 (Suppl.1), 59–67. http://dx.doi.org/10.1093/llc/fqi024.
54. Koppel, M., Schler, J., Argamon, S., 2010. Authorship attribution in the wild. Lang. Resour Evaluat. 45 (1), 83–94. http://dx.doi.org/10.1007/s10579-009-9111-2.
55. Zheng R, Jiexun Li, Hsinchun Chen, and Zan Huang (2006) A Framework for Authorship Identification of Online Messages: Writing-Style Features and Classification Techniques. Journal of The American Society for Information Science and Technology, 57(3):378–393, 2006.
56. Han, J., Pei, J. and Kamber, M., 2011. Data mining: concepts and techniques. Elsevier.
57. Al-Sarem M., Emara, A., Kissi. M., Abel Wahab, A. (2018) " Combination of Stylo-based Features and Frequency-based Features for Identifying the Author of Short Arabic Text". In Proceeding 12th International Conference on Intelligent Systems: Theories and Applications (SITA'18). EMI Rabat - Morocco
58. AlOtaibi, S., & Khan, M. B. Sentiment Analysis Challenges of Informal Arabic. (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 8, No. 2, 2017.
59. de Roeck, A.N., Al-Fares, W., 2000. A morphologically sensitive clustering algorithm for identifying Arabic roots. In: Proceedings of the 38th Annual Meeting on Association for Computational Linguistics, ACL'00, Stroudsburg, PA, USA, pp. 199–206. Doi: http://dx.doi.org/10.3115/1075218.1075244.
60. Ellen, J., Parameswaran, S., (2011). Machine Learning for Author Affiliation within Web Forums - Using Statistical Techniques on NLP Features for Online Group Identification. ICMLA (1), pp : 100-105
61. Alam, H., & Kumar, A. (2013, November). Multi-lingual author identification and linguistic feature extraction—A machine learning approach. In Technologies for Homeland Security (HST), 2013 IEEE International Conference on (pp. 386-389). IEEE.
62. Howedi, F., & Mohd, M. (2014). Text classification for authorship attribution using Naive Bayes classifier with limited training data. Computer Engineering and Intelligent Systems, 5(4), 48-56.
63. Otoom, A. F., Abdallah, E. E., Hammad, M., Bsoul, M., & Abdallah, A. E. (2014). An intelligent system for author attribution based on a hybrid feature set. International Journal of Advanced Intelligence Paradigms, 6(4), 328-345.
64. Sayoud, H. (2014, November). Automatic authorship classification of two ancient books: Quran and Hadith. In Computer Systems and Applications (AICCSA), 2014 IEEE/ACS 11th International Conference on (pp. 666-671). IEEE.
65. Al-Falahi A., Ramdani M., Bellafkih M, Al-Sarem M., (2015 ) "Authorship attribution in Arabic poetry". 10th International Conference on Intelligent Systems: Theories and Applications (SITA).
66. Bourib, S., & Khennouf, S. (2015). Author Identification Using Different Sizes of Documents: A Summary. Hidden Data Mining and Scientific Knowledge Discovery (HDSKD) Journal, 1, 9-12.
67. Rabab'ah, A., Al-Ayyoub, M., Jararweh, Y., & Aldwairi, M. (2016, November). Authorship attribution of Arabic tweets. In Computer Systems and Applications (AICCSA), 2016 IEEE/ACS 13th International Conference of (pp. 1-6). IEEE.
68. Sayoud, H., & Hadjadj, H. (2017, October). Fusion Based Authorship Attribution-Application of Comparison Between the Quran and Hadith. In International Conference on Arabic Language Processing (pp. 191-200). Springer, Cham.
69. Ouamour S., Sayoud H. (2018) A Comparative Survey of Authorship Attribution on Short Arabic Texts. In: Karpov A., Jokisch O., Potapova R. (eds) Speech and Computer. SPECOM 2018. Lecture Notes in Computer Science, vol 11096. Springer, Cham.
70. Saaty, T. L. (2008). Decision making with the analytic hierarchy process. International journal of services sciences, 1(1), 83-98.
71. Nettleton, D. F., Orriols-Puig, A., & Fornells, A. (2010). A study of the effect of different types of noise on the precision of supervised learning techniques. Artificial intelligence review, 33(4), 275-306.
72. Al-Sarem, M., & Al-Tamimi, B. N. (2016, October). Fuzzy unbalanced linguistic variables to enhance the course assessment process. In Intelligent Systems: Theories and Applications (SITA), 2016 11th International Conference on (pp. 1-5). IEEE.
73. Bahassine, S., Madani, A., Al-Sarem, M., & Kissi, M. (2018). Feature selection using an improved Chi-square for Arabic text classification. Journal of King Saud University-Computer and Information Sciences.
74. Hall MA (1999) Correlation-based feature selection for machine learning. PhD thesis, The University of Waikato.
75. Martin Guetlein, Eibe Frank, Mark Hall: Large Scale Attribute Selection Using Wrappers. In: Proc IEEE Symposium on Computational Intelligence and Data Mining, 332-339, 2009.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.2964952, IEEE Access



76. Dunteman GH (1989) Principal components analysis, vol 69. Sage, Thousand Oaks.
77. Zhou, Z. H., Wu, J., & Tang, W. (2002). Ensembling neural networks: many could be better than all. Artificial intelligence, 137(1-2), 239-263.
78. Lazarevic, A., & Obradovic, Z. (2001, July). Effective pruning of neural network classifier ensembles. In IJCNN'01. International Joint Conference on Neural Networks. Proceedings (Cat. No. 01CH37222) (Vol. 2, pp. 796-801). IEEE.
79. Giacinto, G., & Roli, F. (2001). An approach to the automatic design of multiple classifier systems. Pattern recognition letters, 22(1), 25-33.
80. Lin, C., Chen, W., Qiu, C., Wu, Y., Krishnan, S., & Zou, Q. (2014). LibD3C: ensemble classifiers with a clustering and dynamic selection strategy. Neurocomputing, 123, 424-435.
81. Bryll, R., Gutierrez-Osuna, R., & Quek, F. (2003). Attribute bagging: improving accuracy of classifier ensembles by using random feature subsets. Pattern recognition, 36(6), 1291-1302.
82. Sáez, J. A., Luengo, J., & Herrera, F. (2016). Evaluating the classifier behavior with noisy data considering performance and robustness: The equalized loss of accuracy measure. Neurocomputing, 176, 26-35.
83. Nikam, S. S. (2015). A comparative study of classification techniques in data mining algorithms. Oriental journal of computer science & technology, 8(1), 13-19.

**Mohammed Al-Sarem** received the M.S. degree in Information Technology from the Faculty of Informatics and Computer Engineering, Volgograd State Technical University, Volgograd, Russia, and the Ph.D. degree from the Faculty of Informatics, University of Hassan II Casablanca-Mohamadia, Mohamadia, Morocco, in 2007 and 2014, respectively. He is currently an Assistant Professor with the Information System Department, Taibah University, Al Madinah Al Munawarah, Kingdom of Saudi Arabic. He published several papers and participated in managing several international conferences. His current research interests include group decision making, multi-criteria decision making, data mining, E-learning, natural language processing and social analysis.

**Faisal Saeed** is an Assistant Professor at Information Systems Department, Taibah Universitym, KSA since 2017. Previously, he worked as Senior Lecturer at the Department of Information Systems, Faculty of Computing, Universiti Teknologi Malaysia (UTM), Malaysia. He received his BSc in Computers (Information Technology) from Cairo University, Egypt, MSc in Information Technology Management and PhD in Computer Science from UTM, Malaysia. His research interests are data mining, information retrieval and machine learning.

**Abdullah Alsaeedi** received the B.Sc. degree in computer science from the College of computer science and engineering, Taibah University, Madinah, Saudi Arabia, in 2008, M.Sc. degree in Advanced software engineering, The university of Sheffield, department of computer science, Sheffield, UK, in 2011, and the Ph.D. degree in computer science from the University of Sheffield, UK, in 2016. He is currently an Assistance Professor at the Computer Science Department, Taibah University, Madinah, Saudi Arabia. His research interests include software engineering, software model inference, grammar inference, machine learning, Social Network Mining, data mining and document processing.

**Wadii Boulila** received the Engineering degree (Hons.) in computer science from the Aviation School of Borj El Amri, in 2005, the M.Sc. degree from the National School of Computer Science (ENSI), University of Manouba, Tunisia, in 2007, and the Ph.D. degree conjointly from ENSI and Telecom Bretagne, University of Rennes 1, France, in 2012. From 2012 to 2015, he was an Assistant Professor in computer science with the Higher Institute of Multimedia Arts of Manouba, Manouba University, Tunisia. He is currently an Assistant Professor of computer science with the IS Department, College of Computer Science and Engineering, Taibah University, Saudi Arabia. He is also a Permanent Researcher with the RIADI Laboratory, University of Manouba, and an Associate Researcher with the ITI Department, University of Rennes 1, France. His primary research interests include big data analytics, deep learning, data mining, artificial intelligence, uncertainty modeling, and remote sensing images. He has served as the Chair, Reviewer, and a TPC Member for many leading international conferences and journals. Dr. Boulila is a Senior IEEE Member.

**Tawfik Al-Hadhrami** is currently working as a Senior Lecturer at the Nottingham Trent University, UK. He received his MSc degree in IT/Applied System Engineering from Heriot-Watt University, Edinburgh, United Kingdom. He received his PhD degree in Wireless Mesh Communication from University of the West of Scotland, Glasgow, UK, 2015. He was involved in research at University of the West of Scotland, Networking Group. He is an *Associate Editor* for *IEEE Access and IEEE Sensors journals.* His research interest includes Internet of Things (IoT) and Applications, Network Infrastructures & Emerging Technologies, Artificial Intelligence, Computational Intelligence and 5G Wireless Communications. He is a member of Network Infrastructure and Cyber Security group (NICS) at NTU. He is involved in different projects with industries.