IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received November 19, 2019, accepted December 20, 2019, date of publication January 8, 2020, date of current version January 28, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2964952

Ensemble Methods for Instance-Based Arabic Language Authorship Attribution

MOHAMMED AL-SAREM[1], FAISAL SAEED[1], ABDULLAH ALSAEEDI[1], WADII BOULILA[1,2], (Senior Member, IEEE), AND TAWFIK AL-HADHRAMI[3]

[1]College of Computer Science and Engineering, Taibah University, Medina 42353, Saudi Arabia
[2]RIADI Laboratory, National School of Computer Sciences, University of Manouba, Manouba 2010, Tunisia
[3]School of Science and Technology, Nottingham Trent University, Nottingham NG11 8NS, U.K.

Corresponding author: Mohammed Al-Sarem (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 the Arabic language, the AA problem has received less attention from the research community due to the complexity and nature of Arabic sentences. The paper presented an intensive review of 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 to 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 identify the original author of an 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 the author's writing style that can change from topic to topic, some persistent uncontrolled habits and writing styles are still valid over time. The author of anonymous text can be recognized by matching the observed writing style to one of the candidate authors 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 the sixties of the last century up until the1990s, both the approaches and applications 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 at 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 to make a fair attribution? How can we improve the precision of the authorship attribution? Does the size of the training set affect the result? What does happen if the dataset unbalanced? What is the optimum data size?

Recently, current studies in authorship attribution benefit from the explosion in the 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 (NB) [4], [20], [24], [25], Bayesian classifiers [26], [27], k-nearest neighbor (k-NN) [28], [29], decision trees [30], and

The associate editor coordinating the review of this manuscript and approving it for publication was Fatih Emre Boran.

Recurrent Neural Network (RNN) [31]. Although the ensemble methods showed a good performance to improve machine learning results, few studies such as [32]–[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.

The Arabic language is the mother tongue for more than 250 million people who reside mainly on two different continents. However, the works on AA for Arabic are still less numerous than those on English [5], [23], [35]–[46]. 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], [47], 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 [48]. 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 the document's characteristics or features. In literature, such characteristics or features are known as the 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 extracts 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

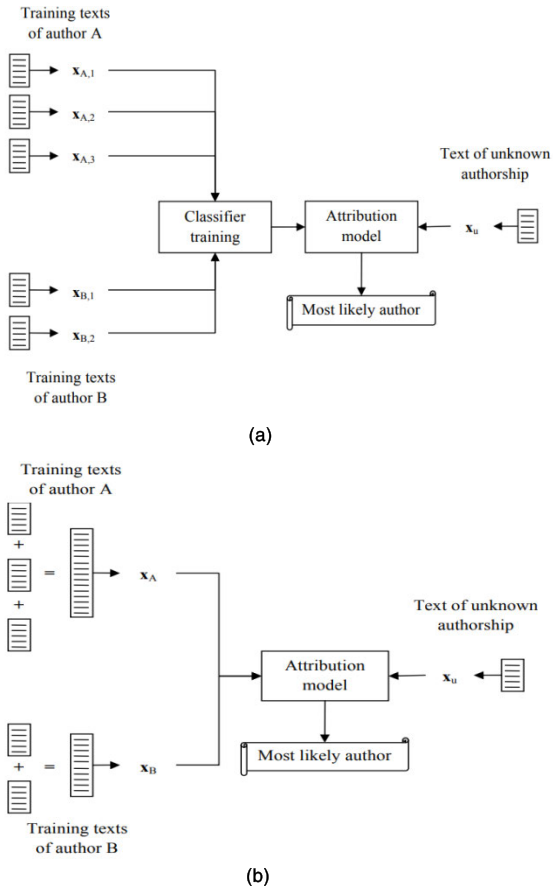

FIGURE 1. A typical architecture for authorship attribution task [16]: (a) instance-based approaches, whereas (b) profile-based approaches.

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 the attributing process.

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 of the way that is used to handle the training text corpus. The earlier attempts to handle stylo-features go back to the 19th century. Most of such methods were statistical attempts in its nature where the researchers have tried to quantify the writing style. However, with the emergence of the Internet, a vast amount of electronic texts was produced and the need for handling these texts is increased. In the shadow of these needs, domains such as machine learning, natural language processing, and information retrieval have an impact in guiding the authorship-attribution research directions.

Back to the earlier era of authorship attribution, we can classify the used features in the attributing stage into two

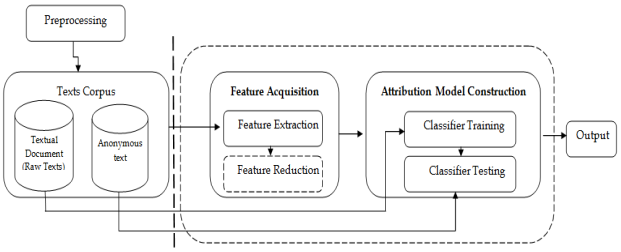

FIGURE 2. A typical machine-learning-based authorship attribution process. The reduction phase surrounded in dashed lines is an optional step depends on the complexity of space dimensions.

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, word frequencies, and sentence length to distinguish between authors. The unitary invariant methods gave unreliable results. The multivariate analysis methods, on the opposite, deal with a set of features to statistically attribute texts. Methods such as 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 the 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, a dataset is divided into sets: training set and testing set. The former set is used to learn classifiers on how to predict class labels, whilst data outside the training set (called a testing set) is used to evaluate how well the model does. Classification and regression analysis are the common supervised learning task. Unsupervised methods are a 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 concluded 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.

2) AUTHORSHIP ATTRIBUTION FEATURES

As an 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 the text. Surveying authorship attribution studies, these features can be categorized into lexical, character, syntactic, semantic, content-specific, structural and language-specific [16], [35], [47].

- 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 a 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 a subset of lexical features where the text content is 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 part-of-speech (POS) [16].
- Semantic, on the opposite of the aforementioned features, semantic features are high-level natural language processing tasks. Surveying literature, only a few attempts address semantic features.
- Application-specific, these features can be either structural, content-specific, and language-specific. The author's signature, font colors, and font size are obvious structural features used for attributing the author [55]. Content-specific features can be extracted from the available texts only and only if all authors, in the corpus, are of the same topic. The language-specific features are also common in attributing the author. However, to measure them, it has to be defined manually.

B. ENSEMBLE LEARNING

Improving the 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 is called in data mining domain as ''ensemble methods''. It is obvious that the accuracies of classifiers are varying and some classifiers perform better than others in some cases.

Thus, finding a way to combine them tends to be more accurate than working with each classifier separately. Ensemble methods are a type of learning algorithms that combine a set of classifiers and then use a (weighted) vote of their prediction for classifying new data points. The 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.

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. The most popular ensemble methods used in the machine-learning domain are bagging, boosting and random forest [56].

2) SELECTION OF BASE CLASSIFIER OF ENSEMBLE 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. Zhou et al. *et al.* [57] 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 [58], which uses k-means to identify the groups that had similar classifiers and then eliminated redundant classifiers that were in each cluster. A similar approach is also found in [59] where the hierarchical agglomerative clustering algorithm is used. However, the empirical analysis shows that the clustering-based selective ensemble techniques have a bad instability [60]. In [61] 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, the selection of the right base classifier plays a 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. Sáez [62] 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 C4.5 method globally behaves better with noisy data.

From sensitivity to increase the dataset size, the SVM shows notable robustness rather than C4.5. Nikam [63] 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 the 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 the case of imbalanced data, the oversampling technique is used to split the 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 satisfiy 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. On the context of Arabic texts, authorship attribution problem has received less attention [45]. In this section, we present some issues that have a direct impact on AA in the context of Arabic language. 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 the period from 2005 up to 2018.

1) ARABIC CHARACTERISTICS

From a 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 tasks where the words should syntactically be regarded as phrases rather than single words. The orientation of writing in Arabic, as it is known, is from right-to-left and the letters are connected to each other which makes Arabic writing differs distinctly from any other Latin-based languages like English, French, etc.

In Arabic, there is 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 diacritic[2] to produce a new word.

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

Although the inflection and diacritics increase the number of words, extracting stylometric features such as vocabulary richness measures might influence [48].

2) CHALLENGES IN ARABIC CONTEXT

Arabic is a very rich and challenging language. As stated above, Arabic is a very derivative and inflective language [46]. Due to that, several challenges that have to be dealt with before working on the authorship attribution task include: diacritics, morphological characteristics, structure and orientation of writing, elongation, word length, and word meaning'[64].

- *Diacritics* are special marks placed above or below the words. Diacritics play an essential role in representing short vowels and changing the word meaning and pronunciation.
- *morphologica*l characteristics, one of the distinguished features of Arabic is a number of produced words from a common root. Such a process is known as inflection where the word is derived by adding affixes (prefixes, infixes, and suffixes) [5]. Arabic words, in general, are categorized into four groups: word, morpheme, root, and stem [65].
- 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 hexliteral. However, a letter might play the role of words. The word might have several different meanings based on the context [64].

D. MACHINE LEARNING METHODS IN ARABIC AUTHORSHIP ATTRIBUTION

In the context of authorship attribution, various methods for attributing Arabic texts have been used. Abbasi and Chen [48] were the first who addressed authorship attribution in the 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, the number of elongation characters is calculated and it is used later as a feature. Abbasi and Chen [35] 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 [66].

Stamatatos [37] proposed an SVM based model for solving the class imbalance problem. The dataset was collected from Alhayat newspaper reports. Lin [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] used SMO-SVM, linear regression (LR) and multilayered perceptron (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 [68] also used the SVM method to identify the 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 a combination of all features.

Alwajeeh *et al.* [42] used 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 improve the 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 [69] investigated the effectiveness of NB and SVM classifiers on attributing short historical Arabic texts written by 10 different authors. On the opposite of the findings in [42], NB exceeds SVM in terms 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.* [70] introduced a hybrid approach which consists of 27 stylometric features. The ensemble classifier that consists of many decision trees, MultiBoostAB, NB, SVM, and BayesNet classifiers were employed on a dataset with 456 Arabic newspaper instances. The best accuracy was 88 % achieved by the MultiBoostAB classifier with the hold-out test and 82% with the cross-validation test.

Sayoud [71] addressed the problem of authorship discrimination. For this purpose, the Quran and the 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.* [72] applied the Markov chain classifier on Arabic poetry with 33 different poets belong to the same era. In terms of features, the authors used content-specific features such as metre of poem and rhyme. The features were partitioned in the 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 [72] and the metre of the Arabic poetry and followed the same methodology. The best average accuracy they got was 72, 83% when the set of all features was used and SMO was applied.

Bourib and Khennouf [73] addressed the authorship attribution problem when the genre and topic are quite similar. The text size in the training set was varied from 100 words to 3000 words per 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 is dependent mainly on the text size, on one hand. On the other hand, it is affected by the used features and the classification techniques themselves.

Social media posts were also under consideration (see Table 1). Rabab'ah *et al.* [74] 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 the SVM classifier on the combined feature sets. In [45], they extended the experiment to include features extracted by the bag-of-words approach. Several reduction techniques were used. The findings show that the SVM classifier outperforms all of the other methods in terms of accuracy and the *SubEval* feature selection technique led reducing the classifier running time.

TABLE 1. Publications on Arabic authorship attribution domain.

Publication	Domain	Text Size
[35] [48] [67]	Forum Messages	Short
[37] [42] [68] [70]	Newspaper	Long
[39] [40] [71] [75]	Historical Texts	Long
[73]		Short-Long
[23] [72]	Arabic Poetry	Short
[45] [74]	Social Media	Short
[44]	Modern Islamic Fatwas	Short

Sayoud and Hadjadj [75] extended the work in [71]. They proposed to fuse two approaches: feature-based decision fusion, which combines 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 terms 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 *et al.* [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%.

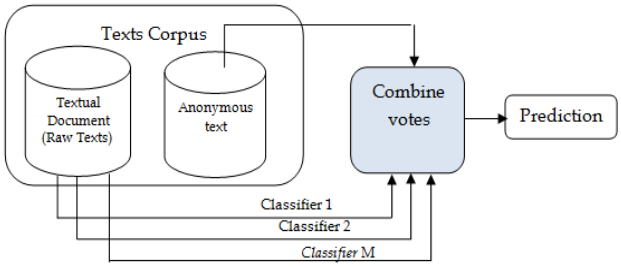

FIGURE 3. Illustrative ensemble learning methods for AA: the ensemble method generates a set of classifiers for a training set, the class of the unseen text is labeled and voted by each classifier. The ensemble, then, combines the votes and returns a class prediction.

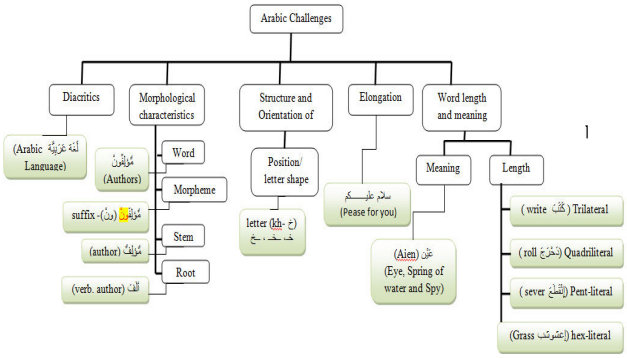

FIGURE 4. Arabic characteristics: the leaves present an illustrative example.

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), 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 cumbersome task. Instead of that, we propose to use the 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 the 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 detail.

1) TOPSIS-BASED AHP METHOD

Saaty [76] introduced (TOPSIS) a technique for order preferences by calculating their similarity to the so-called *ideal solution*. TOPSIS is widely used technique for scoring, ranking and choosing the best alternative. Its proficient ability to handle both subjective and objective attributes is the

TABLE 2. Best accuracy obtained in the published works.

Publication	Features	Classifier	Accuracy
[48]	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%
[74]	Lexical + Syntactic features	k-NN	95%
		SVM	97%
[39]	Lexical features	SMO-SVM	80%
[40][76]	Lexical features	MLP	70%
		SMO-SVM	**80%**
		LR	60%
[68]	Lexical + Syntactic + Structural + Content-specific + Semantic features	SVM	98%
[42]	Lexical features	NB	99.4%
		SVM	**99.8%**
[69]	Lexical + Character features	SVM	62.96%
		NB	**71.85%**
[70]	Lexical + Syntactic + Structural + Content-specific features	NB	84.0%
		BayesNet	**86.7%**
		SVM	79.3%
[72]	Lexical + Structural + Content-specific features	Markov chain	96.67%
[23]	Lexical + Structural + Content-specific features	SVM	71.60%
		SMO	**72.83%**
		NB	70.37%
[73]	Character N-grams + Words	SMO-SVM	-
		MLP / LR	-
[45][74]	POS + Stylometric features + Emotional features	SVM	**68.67%**
		DT	59.83%
		NB	38.35%
[71][75]	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%

reason to be one of the most used multi-attribute decision-making method. The TOPSIS method uses AHP to choose the weights for each attribute. So, to employ the TOPSIS method (see Fig.5), these steps should be followed:

(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 2, to fill the pair-wise comparison matrix of the criteria relating to the goal.

TABLE 3. Converting scale used in this paper.

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

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

D- Ability to deal with high dimensional data.

P- Performance when increasing size of training set.

S- Sensitivity to noise data (the scale is assigned based on [77])

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

(ii) Create the 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.

To allow dealing with categorical values as given in Eq.1, as shown at the bottom of this page, it is required to convert them into numerical values by using a consensual scale. In our case, we use the scale presented in Table 3. It is also necessary to uniform scaling by normalizing $M'_{p\times q}$ as:

$$M'_{pq} = \frac{M_{ij}}{\sqrt{\sum_{j=1}^{q} M_{ij}^2}} \tag{2}$$

Hence, the decision table $M_{p\times q}$ is transformed into $M'_{p\times q}$ as shown in Eq.3, as shown at the bottom of the next page.

(iii) Assign weights to attributes

Following Saaty scale [76], the 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 [78] as:

$$A_{5\times5} = \begin{array}{c|ccccc} & A & C & D & P & S \\ A & 1 & 1 & 5 & 3 & 9 \\ C & 1 & 1 & 3 & 5 & 9 \\ D & 1/5 & 1/3 & 1 & 5 & 3 \\ P & 1/3 & 1/5 & 1/5 & 1 & 3 \\ S & 1/9 & 1/9 & 1/3 & 1/3 & 1 \end{array} \tag{4}$$

The relative normalized weights W are found by computing the geometric mean Gm for each attribute of $A_{q\times q}$ as

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

$$M_{7\times5} = \begin{array}{c|ccccc} & A & C & D & P & S \\ \text{LinearSVM} & 84.28 & \text{v.high} & \text{v.high} & \text{v.high} & \text{high} \\ SMO-SVM & 77.61 & \text{v.high} & \text{v.high} & \text{v.high} & \text{v.high} \\ \text{MLP} & 85 & \text{medium} & \text{high} & \text{high} & \text{medium} \\ \text{LR} & 60 & \text{medium} & \text{Low} & \text{high} & \text{v.Low} \\ \text{DT} & 68.22 & \text{Low} & \text{Low} & \text{v.Low} & \text{Low} \\ \text{NB} & 81.41 & \text{medium} & \text{v.high} & \text{high} & \text{v.Low} \\ \text{kNN} & 73.5 & \text{Low} & \text{medium} & \text{v.high} & \text{medium} \end{array} \tag{1}$$

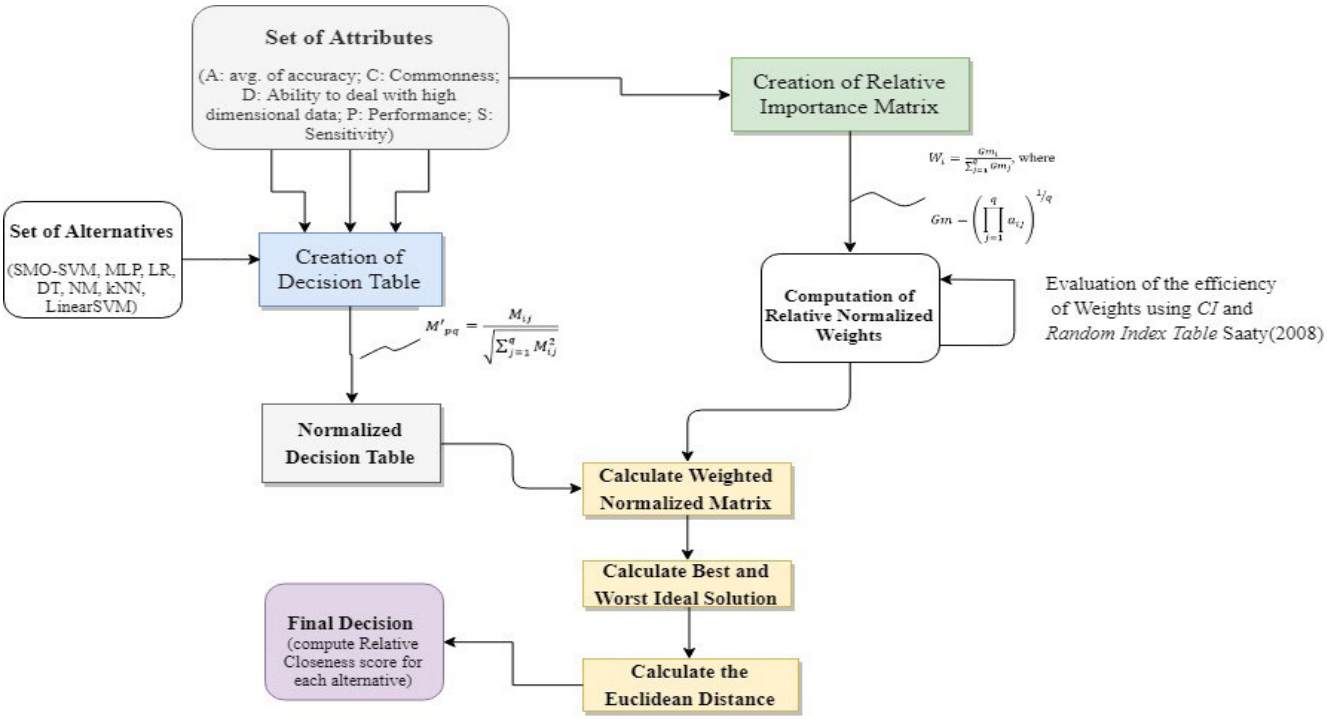

FIGURE 5. Steps followed to rank classifiers using AHP-TOPSIS.

follows:

$$W_i = \frac{Gm_i}{\sum_{j=1}^{q} Gm_j}, \tag{5}$$

where

$$Gm = \left(\prod_{j=1}^{q} a_{ij}\right)^{1/q} \tag{6}$$

The final normalized relative importance weighting matrix is represented in

$$W = \begin{array}{c|c} A & 0.3742 \\ C & 0.3742 \\ D & 0.1403 \\ P & 0.0737 \\ S & 0.0375 \end{array} \tag{7}$$

(iv) Check for consistency and correctness

The consistency index *(CI)* is computed by finding the mean of eigenvalues Λ as:

$$CI = {}^{(\Lambda - q)}/_{(q-1)}, \tag{8}$$

where:

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

$$\lambda_i = \begin{array}{|c|} 5.3682 \\ 5.0123 \\ 5.8518 \\ 5.6169 \\ 5..1152 \end{array}$$

and $\Lambda = 5.39292$ which means that $CI = 0.0884$. Based on Saaty's model, the acceptable consistency ratio $CR = {}^{CI}/_{RI}$ should be less 0.1. Random Index value *RI* is determined

$$M'_{7\times5} = \begin{array}{c|ccccc} & A & C & D & P & S \\ LinearSVM & 0.418336 & 0.542326 & 0.481125 & 0.449013 & 0.496139 \\ SMO-SVM & 0.382879 & 0.542326 & 0.481125 & 0.4490135 & 0.620174 \\ MLP & 0.42211 & 0.325396 & 0.384900 & 0.359211 & 0.372104 \\ LR & 0.29796 & 0.325396 & 0.19245 & 0.359211 & 0.124035 \\ DT & 0.338781 & 0.21693 & 0.19245 & 0.089803 & 0.248069 \\ NB & 0.404282 & 0.325396 & 0.481125 & 0.359211 & 0.124035 \\ kNN & 0.365001 & 0.21693 & 0.288675 & 0.449013 & 0.372104 \end{array} \tag{3}$$

TABLE 4. Random consistency (RI) used in Saaty [76].

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

based on Table 4. In our case, $CR = {}^{0.0884}/_{1.12} = 0.07963$ which means the model is acceptable.

(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.5, (9) as shown at the bottom of this page.

(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:

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

$$l_i^+ = \begin{cases} \max(C_{pq}), & \forall q \in n \\ \min(C_{pq}), & \forall q \in n' \end{cases} \quad and\ p = 1\ to\ P \quad (10)$$

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

$$l_i^+ = \begin{cases} \min(C_{pq}), & \forall q \in n \\ \max(C_{pq}), & \forall q \in n' \end{cases} \quad and\ p = 1\ to\ P \quad (11)$$

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:

$$E_p^+ = \sqrt{\sum_{i=1}^{q} (C_{pi} - L_i^+)^2} \quad (12)$$

$$E_p^- = \sqrt{\sum_{i=1}^{q} (C_{pi} - L_i^-)^2} \quad (13)$$

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

$$S_p^+ = \frac{E_p^-}{(E_p^+ + E_p^-)}$$

$$\Longrightarrow S_p^+ = \begin{vmatrix} LinearSVM & 0.439631 \\ SMO - SVM & 0.440052 \\ MLP & 0.364438 \\ LP & 0.363457 \\ DT & 0.351346 \\ NB & 0.369419 \\ kNN & 0.340649 \end{vmatrix}$$

$$C = \begin{array}{c|ccccc} & A & C & D & P & C \\ linearSVM & 0.156552 & 0.202953 & 0.067503 & 0.033093 & 0.018626 \\ SMO - SVM & 0.143283 & 0.202953 & 0.067503 & 0.033093 & 0.023283 \\ MLP & 0.157965 & 0.121772 & 0.054003 & 0.026475 & 0.013970 \\ LR & 0.111504 & 0.121772 & 0.027001 & 0.026475 & 0.004657 \\ DT & 0.126781 & 0.081181 & 00.027001 & 0.006619 & 0.009313 \\ NB & 0.151293 & 0.121772 & 0.067503 & 0.026475 & 0.004657 \\ kNN & 0.136593 & 0.081181 & 0.040502 & 0.033093 & 0.01397 \end{array} \quad (9)$$

$$\Longrightarrow S_p^+ = \begin{array}{|c|c|}\hline LinearSVM & 0.439631 \\ SMO-SVM & 0.440052 \\ NB & 0.369419 \\ MLP & 0.364438 \\ LR & 0.363457 \\ DT & 0.351346 \\ kNN & 0.340649 \\ \hline \end{array} \quad (14)$$

The alternative with the 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 NB classifiers.

2) CORPUS

The absence of benchmark datasets of authorship attribution on Arabic add more difficulties for evaluating attribution classifiers' performance. Most of the publications on the Arabic authorship attribution domain use different datasets obtained from different sources (see Table 1). 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 languages including Arabic and 9 other languages. Typically, the fatwa follows a well-defined structure. Apart from that, we deal with it as regular textual content. We limit our corpus to only those fatwas written in Arabic. To extract the content of fatwa' from the website, we used the OctoParse 7.0.2 web scraping tool.[5] The Octoparse is an easy configurable visual tool. It allows running 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 ensure 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 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 (Mofti). Thus, the training set is preprocessed before employing an attribution classifier.

3) DATA PRE-PROCESSING

Before doing any preprocessing, the corpus is firstly divided into two sub-corpuses. The current step allows us to investigate the 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 a 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, other necessary preprocessing steps are performed:

- Normalization: to avoid any variation in Arabic word representation, we follow the steps stated in [5], [79]
  - 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 too[7] for both sub-corpora before extracting attribution features.

A. 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) 335 out of 392 features were extracted by the Alwajeeh's Arabic SF tool (Table 5), and 56 morphological features extracted by MADAMIR[8] tool (Table 6), and (ii) 350 distinct words extracted by the WEKA[9] tool.

1) 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, an 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*)

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

TABLE 5. 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 6. 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
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

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 β_1, β_2, β_3 and β_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 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 words within a fatwa is between 400 words per 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 author varies from 7 fatwas per author to 80. We limit the amount of words within the text to be between 800 words per 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 text.

IV. RESULTS AND DISCUSSION

A. FEATURE-BASED LEVEL

To investigate the performance of using different stylometric features (*ASFMs, DWs, and ASFMs + DWs)*, Tables 7-14 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 show that the combination set of features (ASFMs + DWs) obtained the best performance using Bagging and AdaBoost methods for balanced datasets, except for dataset subset $\beta 1$. The dataset size of $\beta 1$ 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

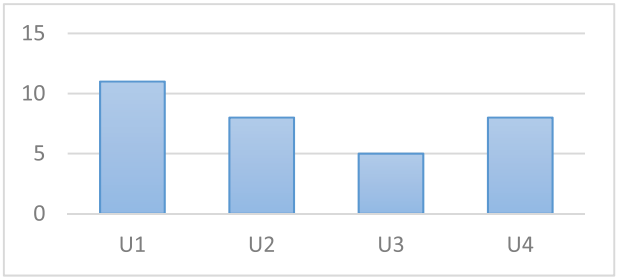

FIGURE 6. Distribution of the number of authors per imbalanced dataset.

TABLE 7. Result of different ensemble techniques on balanced dataset.

Dataset	Classifier	Feature set	Acc.	Prec.	Recall	F1-score
β_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 8. Result of different ensemble techniques on balanced dataset.

Dataset	Classifier	Feature set	Acc.	Prec.	Recall	F1-score
β_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 9. Result of different ensemble techniques on balanced dataset.

Dataset	Classifier	Feature set	Acc.	Prec.	Recall	F1-score
β_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**

TABLE 10. Result of different ensemble techniques on balanced dataset.

Dataset	Classifier	Feature set	Acc.	Prec.	Recall	F1-score
β_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**

(5 out of 8 cases). Similar to the case of $\beta 1$, the DW features obtained better results for the dataset subset U1.

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 by decreasing the number of authors in the dataset. As a conclusion, we recommend using the AdaBoost method for solving the authorship verification problem for balanced datasets. However, for imbalanced datasets, the performance of the Bagging method outperformed the AdaBoost method using all datasets subsets. In addition,

TABLE 11. 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 12. 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 13. 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 14. 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

TABLE 15. 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

the results showed that when the size of the imbalanced dataset increased, the performance of the Bagging classifier decreased.

B. CLASSIFIER-BASED LEVEL

Table 15 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 16 summarizes the median and mean values computed for all Balanced dataset for each ensemble classifiers.

TABLE 16. 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

TABLE 17. 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 18. 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

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.

On the other hand, Table 17 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 18 shows the mean and median values computed for all imbalanced datasets for each classifier. In all cases, the Bagging classifier achieved clearly higher median scores compared with AdaBoost. These reported mean and median scores show a clear dominance of the Bagging classifier over the AdaBoost and this proved the advantages of bagging classifier in dealing with sparse training data.

V. CONCLUSION AND FUTURE WORK

Authorship Attribution (AA) problem in the Arabic language has been addressed in quite a 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 other 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 were 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 the 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] A. Morgan, "The characteristic curves of composition," *Science*, vol. 9, no. 313, p. 92, Feb. 1889.

[2] G. K. Zipf, *The Psycho-Biology of Language*. Boston, MA, USA: Houghton Mifflin Harcourt, 1935.

[3] G. U. Yule, "On sentence-length as a statistical characteristic of style in prose: With application to two cases of disputed authorship," *Biometrica*, vol. 30, pp. 363–390, 1939.

[4] M. G. Kendall, F. Mosteller, and D. L. Wallace, "Inference and disputed authorship: The federalist," *Biometrics*, vol. 22, no. 1, p. 200, Mar. 1966.

[5] A. S. Altheneyan and M. E. B. Menai, "Naïve Bayes classifiers for authorship attribution of Arabic texts," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 26, no. 4, pp. 473–484, Dec. 2014.

[6] X. Puig, M. Font, and J. Ginebra, "A unified approach to authorship attribution and verification," *Amer. Statistician*, vol. 70, no. 3, pp. 232–242, 2016, doi: 10.1080/00031305.2016.1148630.

[7] I. Krsul and E. H. Spafford, "Authorship analysis: Identifying the author of a program," *Comput. Secur.*, vol. 16, no. 3, pp. 233–257, 1997, doi: 10.1016/S0167-4048(97)00005-9.

[8] T. A. Longstaff and E. E. Schultz, "Beyond preliminary analysis of the WANK and OILZ worms: A case study of malicious code," *Comput. Secur.*, vol. 12, no. 1, pp. 61–77, 1993, doi: 10.1016/0167-4048(93)90013-U.

[9] E. H. Spafford and S. A. Weeber, "Software forensics: Can we track code to its authors?" *Comput. Secur.*, vol. 12, no. 6, pp. 585–595, 1993, doi: 10.1016/0167-4048(93)90055-A.

[10] O. De Vel, "Mining e-mail authorship," in *Proc. Workshop Text Mining, ACM Int. Conf. Knowl. Discovery Data Mining (KDD)*, Aug. 2000.

[11] S. Argamon, M. Šarić, and A. S. Stein, "Style mining of electronic messages for multiple authorship discrimination: First results," in *Proc. 9th ACM SIGKDD Int. Conf.*, New York, NY, USA, 2003.

[12] L. Stearns, "Copy wrong: Plagiarism, process, property, and the law," *Calif. L. Rev.*, vol. 80, p. 513, 1992.

[13] J. Rudman, "The state of authorship attribution studies: Some problems and solutions," *Comput. Humanities*, vol. 31, no. 4, pp. 351–365, 1997.

[14] S. Singhe and F. J. Tweedie, "Neural networks and disputed authorship: New challenges," in *Proc. 4th Int. Conf. Artif. Neural Netw.*, 1995, pp. 24–28.

[15] B. Martin, "Plagiarism: A misplaced emphasis," *J. Inf. Ethics*, vol. 3, no. 2, p. 36, 1994.

[16] E. Stamatatos, "A survey of modern authorship attribution methods," *J. Amer. Soc. Inf. Sci. Technol.*, vol. 60, no. 3, pp. 538–556, 2009, doi: 10.1002/asi.21001.

[17] I. Markov, J. Baptista, and O. Pichardo-Lagunas, "Authorship attribution in portuguese using character n-grams," *Acta Polytechnica Hungarica*, vol. 14, no. 3, pp. 59–78, 2017.
[18] J.-P. Posadas-Durán, H. Gómez-Adorno, G. Sidorov, I. Batyrshin, D. Pinto, and L. Chanona-Hernández, "Application of the distributed document representation in the authorship attribution task for small corpora," *Soft Comput.*, vol. 21, no. 3, pp. 627–639, 2017.
[19] L. Pan, I. Gondal, and R. Layton, "Improving authorship attribution in Twitter through topic-based sampling," in *Proc. Australas. Joint Conf. Artif. Intell.* Cham, Switzerland: Springer, Aug. 2017, pp. 250–261.
[20] E. Dauber, R. Overdorf, and R. Greenstadt, "Stylometric authorship attribution of collaborative documents," in *Proc. Int. Conf. Cyber Secur. Cryptogr. Mach. Learn.*, Jun. 2017, pp. 115–135.
[21] O. Marchenko, A. Anisimov, A. Nykonenko, T. Rossada, and E. Melnikov, "Authorship attribution system," in *Proc. Int. Conf. Appl. Natural Lang. Inf. Syst.* Cham, Switzerland: Springer, Jun. 2017, pp. 227–231.
[22] F. Claude, D. Galaktionov, R. Konow, S. Ladra, and Ó. Pedreira, "Competitive author profiling using compression-based strategies," *Int. J. Unc. Fuzz. Knowl. Based Syst.*, vol. 25, pp. 5–20, Dec. 2017.
[23] A.-F. Ahmed, R. Mohamed, and B. Mostafa, "Machine learning for authorship attribution in Arabic poetry," *Int. J. Future Comput. Commun.*, vol. 6, no. 2, pp. 42–46, Jun. 2017.
[24] P. Szwed, "Authorship attribution for polish texts based on part of speech tagging," in *Proc. Int. Conf., Beyond Databases, Archit. Struct.* Cham, Switzerland: Springer, May 2017, pp. 316–328.
[25] Y. Zhao, J. Zobel, and P. Vines, "Using relative entropy for authorship attribution," in *Proc. Asia Inf. Retr. Symp.* Berlin, Germany: Springer, Oct. 2006, pp. 92–105.
[26] S. R. Pillay and T. Solorio, "Authorship attribution of Web forum posts," in *Proc. eCrime Researchers Summit*, 2010, pp. 1–7, doi: 10.1109/ecrime.2010.5706693.
[27] G. Baron, "Influence of data discretization on efficiency of Bayesian classifier for authorship attribution," *Procedia Comput. Sci.*, vol. 35, pp. 1112–1121, 2014.
[28] P. P. Paul, M. Sultana, S. A. Matei, and M. Gavrilova, "Authorship disambiguation in a collaborative editing environment," *Comput. Secur.*, vol. 77, pp. 675–693, Aug. 2018.
[29] C. Akimushkin, D. R. Amancio, and O. N. Oliveira, "On the role of words in the network structure of texts: Application to authorship attribution," *Phys. A, Stat. Mech. Appl.*, vol. 495, pp. 49–58, Apr. 2018.
[30] S. Lahiri and R. Mihalcea, "Authorship attribution using word network features," 2013, *arXiv:1311.2978*. [Online]. Available: https://arxiv.org/abs/1311.2978
[31] L. Z. Wang. *News Authorship Identification With Deep Learning*. Accessed: Jan. 4, 2017. [Online]. Available: https://cs224d.stanford.edu/reports/ZhouWang.pdf
[32] F. M. Giraud and T. Artières, "Feature bagging for author attribution," in *Proc. CLEF (Online Working Notes/Labs/Workshop)*, 2012.
[33] L. Srinivasan and C. Nalini, "An improved framework for authorship identification in online messages," *Cluster Comput.*, vol. 22, no. S5, pp. 12101–12110, Sep. 2019.
[34] E. Ekinci and H. Takçı, "Comparing ensemble classifiers: Forensic analysis of electronic mails," Tech. Rep., 2013.
[35] A. Abbasi and H. Chen, "Applying authorship analysis to extremist-group Web forum messages," *IEEE Intell. Syst.*, vol. 20, no. 5, pp. 67–75, Sep./Oct. 2005.
[36] A. F. Otoom, E. E. Abdullah, S. Jaafer, A. Hamdallh, and D. Amer, "Towards author identification of Arabic text articles," in *Proc. 5th Int. Conf. Inf. Commun. Syst. (ICICS)*, Apr. 2014, pp. 1–4.
[37] E. Stamatatos, "Author identification: Using text sampling to handle the class imbalance problem," *Inf. Process. Manage.*, vol. 44, no. 2, pp. 790–799, 2008.
[38] K. Shaker and D. Corne, "Authorship attribution in Arabic using a hybrid of evolutionary search and linear discriminant analysis," in *Proc. UK Workshop Comput. Intell. (UKCI)*, 2010, pp. 1–6.
[39] S. Ouamour and H. Sayoud, "Authorship attribution of ancient texts written by ten Arabic travelers using a SMO-SVM classifier," in *Proc. Int. Conf. Commun. Inf. Technol. (ICCI)*, 2012, pp. 44–47.
[40] S. Ouamour and H. Sayoud, "Authorship attribution of short historical Arabic texts based on lexical features," in *Proc. Int. Conf. Cyber-Enabled Distrib. Comput. Knowl. Discovery*, 2013, pp. 144–147.
[41] R. S. Baraka, S. Salem, M. A. Hussien, N. Nayef, W. A. Shaban, "Arabic text author identification using support vector machines," *J. Adv. Comput. Sci. Technol. Res.*, vol. 4, no. 1, pp. 1–11, 2014.
[42] A. Alwajeeh, M. Al-Ayyoub, and I. Hmeidi, "On authorship authentication of Arabic articles," in *Proc. 5th Int. Conf. Inf. Commun. Syst. (ICICS)*, 2014, pp. 1–6.
[43] M. Al-Ayyoub, A. Alwajeeh, and I. Hmeidi, "An extensive study of authorship authentication of Arabic articles," *Int. J. Web Inf. Syst.*, vol. 13, no. 1, pp. 85–104, 2017.
[44] M. Al-Sarem and A.-H. Emara, "Analysis the Arabic authorship attribution using machine learning methods: Application on islamic Fatwā," in *Proc. Int. Conf. Reliable Inf. Commun. Technol.*, Jun. 2018.
[45] M. Al-Ayyoub, Y. Jararweh, A. Rabab'ah, and M. Aldwairi, "Feature extraction and selection for Arabic tweets authorship authentication," *J. Ambient Intell. Humanized Comput.*, vol. 8, no. 3, pp. 383–393, 2017.
[46] J. H. Yousif and T. M. T. Sembok, "Arabic part-of-speech tagger based support vectors machines," in *Proc. Int. Symp. Inf. Technol.*, vol. 3, Aug. 2008, pp. 1–7.
[47] G. Baron, "Analysis of multiple classifiers performance for discretized data in authorship attribution," in *Proc. Int. Conf. Intell. Decis. Technol.* Cham, Switzerland: Springer, Jun. 2017, pp. 33–42.
[48] A. Abbasi and H. Chen, "Applying authorship analysis to Arabic Web content," in *Proc. Int. Conf. Intell. Secur. Inform.* Berlin, Germany: Springer, May 2005, pp. 183–197.
[49] K. Pearson, "LIII. On lines and planes of closest fit to systems of points in space," *London, Edinburgh, Dublin Philos. Mag. J. Sci.*, vol. 2, no. 11, pp. 559–572, 1091.
[50] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Ann. Eugenics*, vol. 7, no. 2, pp. 179–188, 1936.
[51] J. Burrows, "'Delta': A measure of stylistic difference and a guide to likely authorship" *Literary Linguistic Comput.*, vol. 17, no. 3, pp. 267–287, 2002, doi: 10.1093/llc/17.3.267.
[52] V. Kešelj, F. Peng, N. Cercone, and C. Thomas, "N-Gram-based author profiles for authorship attribution," *Comput. Linguistics*, vol. 3, pp. 255–264, 2003, doi: 10.1.1.9.7388.
[53] P. Juola and R. H. Baayen, "A controlled-corpus experiment in authorship identification by cross-entropy," *Literary Linguistic Comput.*, vol. 20, pp. 59–67, Jun. 2005, doi: 10.1093/llc/fqi024.
[54] M. Koppel, J. Schler, and S. Argamon, "Authorship attribution in the wild," *Lang. Resour. Eval.*, vol. 45, no. 1, pp. 83–94, 2010, doi: 10.1007/s10579-009-9111-2.
[55] R. Zheng, J. Li, H. Chen, and Z. Huang, "A framework for authorship identification of online messages: Writing-style features and classification techniques," *J. Amer. Soc. Inf. Sci. Technol.*, vol. 57, no. 3, pp. 378–393, 2006.
[56] J. Han, J. Pei, and M. Kamber, *Data Mining: Concepts and Techniques*. Amsterdam, The Netherlands: Elsevier, 2011.
[57] Z.-H. Zhou, J. Wu, and W. Tang, "Ensembling neural networks: Many could be better than all," *Artif. Intell.*, vol. 137, nos. 1–2, pp. 239–263, May 2002.
[58] A. Lazarevic and Z. Obradovic, "Effective pruning of neural network classifier ensembles," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, vol. 2, Jul. 2001.
[59] G. Giacinto and F. Roli, "An approach to the automatic design of multiple classifier systems," *Pattern Recognit. Lett.*, vol. 22, no. 1, pp. 25–33, Jan. 2001.
[60] C. Lin, W. Chen, C. Qiu, Y. Wu, S. Krishnan, and Q. Zou, "LibD3C: Ensemble classifiers with a clustering and dynamic selection strategy," *Neurocomputing*, vol. 123, pp. 424–435, Jan. 2014.
[61] R. Bryll, R. Gutierrez-Osuna, and F. Quek, "Attribute bagging: Improving accuracy of classifier ensembles by using random feature subsets," *Pattern Recognit.*, vol. 36, no. 6, pp. 1291–1302, Jun. 2003.
[62] J. A. Sáez, J. Luengo, and F. Herrera, "Evaluating the classifier behavior with noisy data considering performance and robustness: The equalized loss of accuracy measure," *Neurocomputing*, vol. 176, pp. 26–35, Feb. 2016.
[63] S. S. Nikam, "A comparative study of classification techniques in data mining algorithms," *Oriental J. Comput. Sci. Technol.*, vol. 8, no. 1, pp. 13–19, 2015.
[64] M. Al-Sarem, A. Emara, M. Kissi, and A. A. Wahab, "Combination of stylo-based features and frequency-based features for identifying the author of short Arabic text," in *Proc. 12th Int. Conf. Intell. Syst.*, 2018.
[65] S. AlOtaibi and M. B. Khan, "Sentiment analysis challenges of informal Arabic," *Int. J. Adv. Comput. Sci. Appl.*, vol. 8, no. 2, 2017.
[66] A. N. de Roeck and W. Al-Fares, "A morphologically sensitive clustering algorithm for identifying Arabic roots," in *Proc. 38th Annu. Meeting Assoc. Comput. Linguistics (ACL)*, 2000.

[67] J. Ellen and S. Parameswaran, "Machine learning for author affiliation within Web forums–using statistical techniques on NLP features for online group identification," in *Proc. ICMLA*, vol. 1, 2011, pp. 100–105.
[68] H. Alam and A. Kumar, "Multi-lingual author identification and linguistic feature extraction—A machine learning approach," in *Proc. IEEE Int. Conf. Technol. Homeland Secur. (HST)*, Nov. 2013, pp. 386–389.
[69] F. Howedi and M. Mohd, "Text classification for authorship attribution using Naive Bayes classifier with limited training data," *Comput. Eng. Intell. Syst.*, vol. 5, no. 4, pp. 48–56, 2014.
[70] A. F. Otoom, E. E. Abdallah, M. Hammad, M. Bsoul, and A. E. Abdallah, "An intelligent system for author attribution based on a hybrid feature set," *Int. J. Adv. Intell. Para*, vol. 6, no. 4, p. 328, 2014.
[71] H. Sayoud, "Automatic authorship classification of two ancient books: Quran and Hadith," in *Proc. IEEE/ACS 11th Int. Conf. Comput. Syst. Appl. (AICCSA)*, Nov. 2014, pp. 666–671.
[72] sA.-F. Ahmed, R. Mohamed, B. Mostafa, and A.-S. Mohammed, "Authorship attribution in Arabic poetry," in *Proc. 10th Int. Conf. Intell. Syst., Theories Appl. (SITA)*, 2015.
[73] S. Bourib and S. Khennouf, "Author identification using different sizes of documents: A summary," *Hidden Data Mining Sci. Knowl. Discovery J.*, vol. 1, pp. 9–12, 2015.
[74] A. Rabab'ah, M. Al-Ayyoub, Y. Jararweh, and M. Aldwairi, "Authorship attribution of Arabic tweets," in *Proc. IEEE/ACS 13th Int. Conf. Comput. Syst. Appl. (AICCSA)*, Nov./Dec. 2016, pp. 1–6.
[75] H. Sayoud and H. Hadjadj, "Fusion based authorship attribution-application of comparison between the Quran and Hadith," in *Proc. Int. Conf. Arabic Lang. Process.*, Oct. 2017, pp. 191–200.
[76] T. L. Saaty, "Decision making with the analytic hierarchy process," *Int. J. Services Sci.*, vol. 1, no. 1, pp. 83–98, 2008.
[77] D. F. Nettleton, A. Orriols-Puig, and A. Fornells, "A study of the effect of different types of noise on the precision of supervised learning techniques," *Artif. Intell. Rev.*, vol. 33, no. 4, pp. 275–306, Apr. 2010.
[78] M. Al-Sarem and B. N. Al-Tamimi, "Fuzzy unbalanced linguistic variables to enhance the course assessment process," in *Proc. 11th Int. Conf. Intell. Syst., Theories Appl. (SITA)*, Oct. 2016.
[79] S. Bahassine, A. Madani, M. Al-Sarem, and M. Kissi, "Feature selection using an improved Chi-square for Arabic text classification," *J. King Saud Univ.-Comput. Inf. Sci.*, 2018, doi: 10.1016/j.jksuci.2018.05.010.

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 Associate Professor with the Information System Department, Taibah University, Medina, Saudi Arabia. He has published several articles and participated in managing several international conferences. His current research interests include group decision making, multicriteria decision making, data mining, E-learning, natural language processing, and social analysis.

FAISAL SAEED received the B.Sc. degree in computers information technology from Cairo University, Egypt, the M.Sc. degree in information technology management, and the Ph.D. degree in computer science from Universiti Teknologi (UTM). He was a Senior Lecturer with the Department of Information Systems, Faculty of Computing, UTM, Malaysia. He has been an Assistant Professor with the Information Systems Department, Taibah University, Saudi Arabia, since 2017. His research interests include 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, Medina, Saudi Arabia, in 2008, the M.Sc. degree in advanced software engineering from the Department of Computer Science, The University of Sheffield, Sheffield, U.K., in 2011, and the Ph.D. degree in computer science from The University of Sheffield, in 2016. He is currently an Assistance Professor with the Computer Science Department, Taibah University. His research interests include software engineering, software model inference, grammar inference, machine learning, social network mining, data mining, and document processing.

WADII BOULILA (Senior Member, IEEE) received the B.Eng. degree 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. He is currently an Assistant Professor of computer science with the IS Department, College of Computer Science and Engineering, Taibah University, Medina, Saudi Arabia. He is 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 served as the chair, a Reviewer, and a TPC member of many leading international conferences and journals.

TAWFIK AL-HADHRAMI received the M.Sc. degree in IT/applied system engineering from Heriot-Watt University, Edinburgh, U.K, and the Ph.D. degree in wireless mesh communication from the University of the West of Scotland, Glasgow, U.K., in 2015. He was involved in research with the Networking Group, University of the West of Scotland. He is currently a Senior Lecturer with Nottingham Trent University, U.K. He is an Associate Editor of IEEE Access and the IEEE Sensors journals. He is involved in different projects with industries. His research interests include the Internet of Things (IoT) and its applications, network infrastructures and emerging technologies, artificial intelligence, computational intelligence, and 5G wireless communications. He is a member of the Network Infrastructure and Cyber Security Group (NICS), NTU.