Text Classification Using Enhanced Binary Wind Driven Optimization Algorithm

Jaffar Atwan[1], Mohammad Wedyan[2], Ahmad Hamadeen[3], Qusay Bsoul[4], Ayat Alrosan[5], Ryan Alturki[6]
Prince Abdullah Bin Ghazi Faculty, AlBalqa Applied University, Jordan[1,3]
Faculty of Information Technology and Computer Sciences, Yarmouk University, Irbid, Jordan[2]
Information Security and Cybersecurity Department, Philadelphia University, Amman, Jordan[4]
School of Information Technology, Skyline University College, Sharjah, United Arab Emirates[5]
Department of Software Engineering, College of Computing, Umm Al-Qura University, Makkah, Saudi Arabia[6]

***Abstract*—Document classification using supervised machine learning is now widely used on the internet and in digital libraries. Several studies have focused on English-language document classification. However, Arabic text includes high variation in its morphology, which leads to high extracted features and increases the dimensionality of the classification task. Towards reducing the curse of dimension in Arabic text classification, a wrapper feature selection method is proposed in this study. In more detail, a hybrid metaheuristic model based on the Wind Driven and Simulated Annealing is designed to solve FS task in Arabic text, known as WDFS. The Wind Driven method is initially introduced to optimize the Fs task in the exploration phase. Then, WD is hybridized with simulated annealing as a local search in the exploitation phase to enhance the solutions located by the WD. Three classifiers are utilized to evaluate the selected features using the proposed WDFS: K-nearest Neighbor, Naïve Bayesian, and Decision Tree. The proposed WDFS method was assessed on selected four groups of files from a benchmark TREC Arabic text newswire dataset. Comparative results showed that the WDFS method outperforms other existing Arabic text classification methods in term of the accuracy. The obtained results reveal the high potentiality of WDFS in reliably searching the feature space to obtain the optimal combination of features.**

Keywords*—Text classification; Arabic documents; wind driven optimization algorithm; simulating annealing; feature selection

I. Introduction

There is a large and growing amount of machine-readable information out there, which makes it more difficult to use and understand. It provides resources that make it possible to automatically recognize and organize huge amounts of text [1]. Technologies concerning document classification that are automatically applied to huge amounts of diverse document information have grown due to the early progress on the Internet and the escalation of electronic information. Among document classification tasks, Arabic document classification is the most recent challenge in the field of document classification because of the Arabic language's complexity [2]. Additionally, there are a lot of documents conducted almost every day which has made it difficult to accurately detect Arabic documents. Thus, many studies have focused on identifying the difficulties facing compilations of standard Arabic documents classifier [3, 4, 5], encouraging more studies for to improve the performance of the Arabic document classifiers. Most Arabic document classifiers are notoriously difficult to handle huge amounts of documents and to properly identify Arabic documents. As a result, this issue is considered the primary Arabic classification problem. There are two main aspects to the obstacles facing Arabic classifiers. The first is the huge diversity of features resulting from different Arabic word variants, and the second aspect is another operation of the document classification system may be affected by the failure of classifiers to handle stemming. Thus, the algorithm will be used to create a stemming rule that relies on the grammatical rules of speech to address the difficulty of processing the Arabic text while bypassing the morphological and grammatical rules [6].

Arabic document classification task is the process of classifying documents based on their content into predefined category classifications [7]. In this task, large, enormous features are extracted from the text, and they may amount to hundreds or thousands. Due to the high dimensions of the feature space, classifier performance may be affected and more time may be required to replace the term with alternative concepts. By sifting through the text and identifying key semantics, the number of characteristics can be reduced; however, this reduction is not necessarily optimal as some of the omitted terms may be significant characteristics [8]. Choosing the right features is the most critical application in machine learning. Substantially, the most important features identify the learning objective where the attention of the learning algorithm highlights the most useful data for analysis and of the future prediction [1, 9, 10, 11]. Concepts from testing theory were used to create a correlation-based feature selection method and evaluate a set of machine learning algorithms that have been trained on various combinations of real-world and compositional issues. Feature selection (hereinafter referred to as FS) is simple and quick to implement because it removes redundant and irrelevant data while also iteratively improving the performance of learning algorithms. Comparison of the results of this technology with a contemporary feature identified from the literature [12] requires much less calculation.

Although several FS methods have been introduced to assist in Arabic text classification, most of these studies focus on filter FS methods, such as X, which introduced the Chi-square filter for Arabic text classification, and Y, which introduced the cosine similarity for this task. These filter methods were utilized to rank the features but not to select a suitable subset. Using wrapper FS methods with an optimization algorithm assists in choosing an appropriate subset, which increases the performance of text classification.

In designing an FS method, the main goal is to find the most appropriate subset of features; hence, FS problems are considered NP-hard problems [7]. Using exact search algorithms with the NP-hard problem will generate all possible feature subsets. For instance, assuming we have N text words (i.e., features), then using FS methods such as wrapper methods will generate 2^N subsets of features using a specific learning algorithm (i.e., KNN), which is computationally costly. Whereas FS aims to keep the number of features selected to a minimum while maximizing classification performance, it can be considered an optimization problem. To choose wisely, it is usually necessary to find a middle ground between these two goals. It is best to think of the feature selection process for text classification as a multi-objective task rather than a single-objective task. Numerous meta-heuristic algorithms based on multi-objective approaches have recently been employed to solve text FS issues in an encapsulated framework [3].

The following contributions were made to the classification of documents in this study:

- The stemming process is introduced as a preprocessing step for Arabic text classification, which groups words of the same root into a single word.
- The Bag of Words method is used as a preprocessing technique to extract features from Arabic documents.
- The BWDO-SA wrapper-based FS method is investigated to select the most relevant features from the extracted features. SA is introduced into the BWDO to improve the original BWDO's capacity for exploitation.
- Different learning algorithms, including Decision Tree, Naïve Bayesian, and K-nearest Neighbor, were investigated with BWDO-SA for classifying Arabic documents.

The rest of this study are organized as following; Section II discusses an overview of the related work for text classification. The research methodology considered in this study and the proposed framework of Arabic text classification is provided in Section III. Section IV introduces the experiment dataset and parameter settings and the performance measures. Section V describes the experiment results and discussion. Finally, Section VI contains the conclusion and the future work.

II. Related Works

Several previous studies on classification in Arabic language are reviewed to uncover new classes of classifiers, as well as to address issues related to them, with the aim of bridging the gap between feature selection and the impact of using classifier algorithms, such as optimization feature selection and popular classifiers. An actual evaluation is conducted on studies related to Arabic classifiers to determine the potential for high-performance Arabic classifiers with a small number of selected features. Additionally, the problems and obstacles related to feature selection are identified and discussed, along with their impact on optimization feature selection, by reviewing text classifiers and classifier algorithms. Text categorization (TC) is a type of supervised learning activity that entails classifying documents according to labels already assigned to a set of training documents. Knowledge engineering, in which a manually generated set of rules is used to classify documents into predefined categories, was the most widely used strategy until the development of machine learning approaches in TC.

In [13], the authors explained that the incorporation of machine learning (ML) in TC offers several advantages, such as reducing costs and time by relying on expert workforce only, without compromising precision. K-Near Neighbor Learning (KNN), Decision Tree (DT), Neural Networks, least Squares Fitting (LLSF) and Naive Bayesian (NB) have all been developed and applied to TC [14, 15, 16]. All of these learning strategies perform equally well when there are more than 300 documents in each category, as measured by their comparison. However, when there are less than ten positive training papers in each category, DT, KNN, and LLSF outperform Neural Networks and NB.

The volume of textual data currently available has increased, leading to initiatives such as Arabic language filtering [17] that aim to filter and highlight incoming papers for unwanted or unwelcome content. This has led to the increasing importance of transcript taxonomy (TC). Sentiment analysis [18] which aims to ascertain the overall sentiment expressed in a document, is another use. By providing classification models with examples labeled for the problem at hand, such as appropriate classifications (labels), supervised learning methods can be used to solve problems associated with text classification. Therefore, labels of unlabeled documents can be predicted using these model [3, 19, 17, 20, 21, 16, 10]. The task still requires more work of pre-defining classes and assigning class labels to training set documents, despite the fact that several supervised machine learning algorithms have been applied to TC.

Researchers in [22] compared different Arabic classifiers using BOW as extraction without FS. The comparison showed that NB performs better than DT and tree-based J48. Another study conducted by [23] compared NB, tree based J48, and IB1 using BOW as an extraction. The study reported NB to be better in classification tasks compared to FS methods. In [24], the authors compared DT, AdaBoost, and RF using BOW as extraction and found that DT is the best classifier. In addition, they contrasted IG with Chi-Square and deduced that both methods are comparable when using DT as a classifier and Chi-Square as an FS. However, RF was the best classifier, compared to DT with FS. A study done by [25] utilized six FS methods and two classifiers (DT and NB) on six datasets. The study merged ODFFS and TFFSs to generate a hybrid method (HBM) to propose the FSEPO for parameter optimization. The report showed that NB is the best classifier compared to DT and HBM in enhancing optimal FS. In [26], the authors conducted a comparison among various classifiers for Arabic text. The text preprocessing steps included removing digits, dashes, punctuation marks, and other non-Arabic characters, as well as filtering out stop words without using FS. The results of the comparison revealed that the decision tree (D.T) classifier performed better than the Naïve Bayesian (NB) method, K-Nearest Neighbors (KNN), and NaïveBayes Multinomial (NBM) classifiers. In [27], the authors compared Decision tree (C5.0) SVM using stop words as extraction with Chi-Squared statistic FS. The study reported a Decision tree (C5.0) to be better in classification tasks compared to SVM.

The selection of feature subsets is a common problem for

Arabic classifiers due to the large dimensionality of textual Arabic types. Therefore, the feature selection (FS) process is important to reduce these dimensions and to choose a large number of high-quality attributes without significantly influencing the performance. The investigation of the problem can be in two areas: the first is to focus on the selection of significant features and eliminate redundant ones.

The problem of selecting the best subset of features from the original feature space without affecting the accuracy of the classifiers is known as FS. The second perspective involves enhancing the prediction performance of the classifiers by considering their specific problems. Existing literature has shown that several approaches have been investigated for FS with different techniques of Arabic classifiers. Most importantly, FS methods should consider the properties of the learning algorithm and the problem domain. The effects of different FS methods on different learning frameworks and fields may vary, and therefore, ongoing studies are being conducted to develop new methods for handling FS and addressing dimensionality problems, particularly for Arabic classifiers. Hence, the aim of using FS approaches may include performance enhancement, data simplification, visualization, as well as feature dimensionality reduction [28, 29].

Various approaches have been applied for dimensionality reduction, including IG, Chi-Square Statistic, MI, TS, and DFT. Chi-square scale the level of dependence between a particular term and a particular category, i.e., A measure of how strongly a term is believed to indicate whether a document belongs to a particular category [30]. Several researchers have reformulated the statistic and deployed it for document categorization [30, 31]. In [32], the authors designed a classifier for Arabic document filters by employing Chi-square and IG on ML techniques, namely, BA tree-based J48 and DT. From their report, the Chi-square and IG showed similar performance as the BA and tree-based J48 classifiers, while DT without FS performed better than Chi-square and IG with FS. IG determines the number of bits of information acquired for each prediction class when the presence or absence of a term is determined in a document.

In [33], the authors suggested an improved Feature selection method for Arabic text classification. To enhance the effectiveness of classification, a number of experiments were performed, comparing improved chi-square with three known types of FS, namely chi-square, information gain, and mutual information, using the SVM classifier on a dataset of 5070 Arabic documents. The researchers evaluated these methods using classification measures, The results showed that the Arabic text classification model performs significantly better when the improved chi-square is combined with the SVM classifier. In [34], the authors suggested an innovative and new technique of feature selection that relies on practical swarm optimization to enhance the work of text classification. The proposed method was compared with several other methods, including genetic algorithm, chi-square, and the information gain, using Matlab and the Reuters_21578 dataset. The proposed method efficiency and effectiveness were evaluated using the nearest neighbor classifier. In conclusion, the findings indicated that the proposed method surpassed other methods. Marie-Sainte and Alalyani [35], introduced a new firefly algorithm that relies on FS for Arabic Text Classification (ATC). In this study, various experiments and comparisons were conducted using six feature selection techniques in combination with one state-of-the-art techniques that used the SVM classifier. These techniques were applied to the OSAC dataset, and three different techniques were used to measure execution accuracy. The outcomes showed that the suggested technique (ATC_FA) produced superior performance.

In [36], the authors presented a hybrid method combining the Flower Pollination Algorithm (FPA) and the Adaboost algorithm. Each algorithm served a different purpose, with FPA focusing on feature selection and Adaboost aiming to categorize text contents. The experiments have been made on Reuters-21578, WEBKB, and CADE 12 datasets, and the outcomes demonstrated that the suggested approach outperforms NB-K-Means, KNN-K-Means, and other learning models in terms of FS and performance. In [37], the authors proposed a new hybrid algorithm that combines bee colony optimization (BCO) and ant colony optimization (ACO) to improve FS in 2017. This approach was applied to 13 datasets from the University of California, using Decision Tree (DT) classifiers. The experiments demonstrated that the hybrid model achieved higher categorization accuracy and better feature selection compared to other methods, including ACO Hybrid ACO, ACO-NN Hybrid, Based FS, IQR Bee, CatFish Binary PSO, PSO-SVM, and ABC-FS Swarm based Hybrids such as ACO-PSO Hybrid, and ABC-DE Hybrid, as evidenced by the Prediction Accuracy (PA) measure. In [29], the authors used the improved feature selection technique of whale optimization with SA. The outcomes were contrasted with those of other population-based techniques were outperformed them in terms of data classification performance.

For feature extraction and selection [38], the authors exploited the Aravec embedding model. Aravec is a large dataset trained using an Arabic twitter data domain. The extracted features were exploited to classify twittes using a multilayer bidirectional long short term memory (BiLSTM). The model outperforms the state of the art models.

The authors of Targio and Tubishat [39] proposed an approach that used a hybrid fillter-wrapper method based on Principal Component Analysis (PCA) to filter the features and select relevant ones. These features will be exploited for categorizing Arabic documents. To select more relevant features GreyWolf Optimizer (GWO) as wrapper approach is performed. To evaluate the classification task the Logistic Regression (LR) is used. Three Arabic datasets were exploited to perform the experimentations. These laters show that the approach-based PCA-GWO outperforms the baseline classifiers.

In the work [40], the authors present a feature section approach that is composed of two stages. the first one exploited the filter model. The second one used and the multi-objective wrapper model, an upgraded version of the Whale Optimization Algorithm (WOA) with Particle Swarm Optimization (PSO). Four benchmark text corpora were used to evaluate the proposed approach which shows encouraging results. The study of [41] presents an approach that uses NLP and machine learning methods for text classification. To tune the hyperprameter of machine learning algorithms two techniques were exploited: Grid Search and Random Search. Experimentations were performed using CNN Arabic dataset. The results showed that random search is more efficient than grid search in terms of accuracy, precision, recall, and F1-score and execution times for some of the algorithms.

In this research we have analyzed a number of previous studies that help us better understand this study. Various concepts, including TC, is one of the most important applications that are in high demand today, and the most used classifiers like k-Nearest Neighbors (KNN), Naive Bayes (NB), and Decision Trees (DT), have been discussed. Given the huge quantity of online information that is constantly increasing, the findings of these studies are valuable in terms of saving effort and reducing costs for users and developers. Additionally, feature selection depends on optimization techniques which result in reducing redundant and unnecessary features in texts.

III. Methodology

The proposed methodology embraces a wrapper-FS method to select the most suitable subset of features from the original feature set to improve the performance of Arabic document classification. Fig. 1 below displays the principal methodology used to conduct this study, which consists of several phases: data pre-processing, feature extraction, feature selection, and finally, the classification phase. The proposed BWDO-SA is introduced within the feature selection phase.

A. Data Pre-Processing

The TC system begins with the pre-processing stage. This step is important for document representation because it pre-processes the collection's text data to determine the indexed features or terms. Several procedures are performed during this phase to clean and clarify the text data [42, 43]: First, a tokenization and normalization procedure is used to remove unnecessary terms such as punctuation marks, non-Arabic letters, diacritics, and digits [44]. Second stop-word removal, which focuses on eliminating unimportant words for machine-learning tasks. The third phase is the stemming phase, which is used in text classification to find the stem word for several related terms. The following steps provide additional details regarding these processes for Arabic document classification.

1) Tokenizing and normalizing of data: When text is divided into separate units by using either space or special symbols. As such, every word in a text is represented in a unit. Such a method is called tokenization. For instance, (خير جليس في الزمان كتاب) it explains the method tokenizing works in a text (خير_جليس_في_الزمان_كتاب). Furthermore, normalization is beneficial for carrying out before the stemming task, particularly in the Arabic text is being performed due to the text normalization that decreases several types of characters in Arabic languages to enable one uniform character to act to those characters. The following steps followed in the pre-processing.

- Remove diacritics, punctuation marks, hyphens, numbers, digits, and non-Arabic letters.
- Substitute إ ,آ as well as أ by ا
- Substitute the last ة by ه.
- Substitute the last ى by ي

2) Stop word removal: The removal of these words will not have any major impact on the document's meanings, but rather will be clearer to smooth the process of the meaning interpretation of the document [6, 44, 45, 43, 46]. Fig. 2 shows the document sample words after the removal of the stop-word.

3) Stemming: A stemming algorithm reduces all the inflectional derivational variants of words into a common form called the stem [45]. For example, the words, 'work', 'works', 'working', 'worked', and 'worker' are taken from the root (stem) 'work'. The root of a word is obtained by removing all or some of the affixes attached to the word. The Light10 stemmer was used in the stemming process following the same steps in [47]. This process applies to words containing additional characters whether suffixes or prefixes to get the root of these words [48]. There are four types of affixes shown in Table I.

B. Feature Extraction

Text pre-processing is a necessary step in TC using ML technologies. This entails converting the text into forms that learning algorithms can use. The document representation vector space model, which is widely used, was first described in [49]. Each dimension in the "d" vector that this model uses to describe each text corresponds to a different term in the document set space term. The text document is often converted into a term frequency vector to implement the document processing process [34, 85]. Each document was represented by a vector, where each term acts as an attribute and the value of the attribute is the document's TF*IDF weight [50] a statistical method for measuring the importance of a term for a document in a group. The most common terminology weighing method (TF*IDF) takes this property into account. According to this method, the term weight in the document d is inversely related to the number of documents in the group where the term appears (IDF) and proportional to the number of times the word appears in the document (TF). When using the TF-IDF weighting strategy, terms are given weight based on their frequency in a document using a factor that ignores their importance if they are present in most documents, especially when the term is thought to have poor discriminatory power [see Eq. (1)].

$$w_i = tf_i . \log(\frac{N}{n_i}) \tag{1}$$

where, N denotes the total amount of documents in the collection, w_i denotes the weight of the term in the document, tf denotes its frequency, and n_i denotes the number of documents in which the term appears. Since each token in each document represents a dimension in the feature space, this transformation raises the problem of high dimensionality of the feature space. As a result, large computational costs significantly reduce classification effectiveness [51].

C. Feature Selection

Choosing a subset of features for an Arabic text classification system is crucial. Generally, in ML, three methods for feature selection: embedded methods, wrapper, and filter. By using these techniques, a cost-effective and accurate text

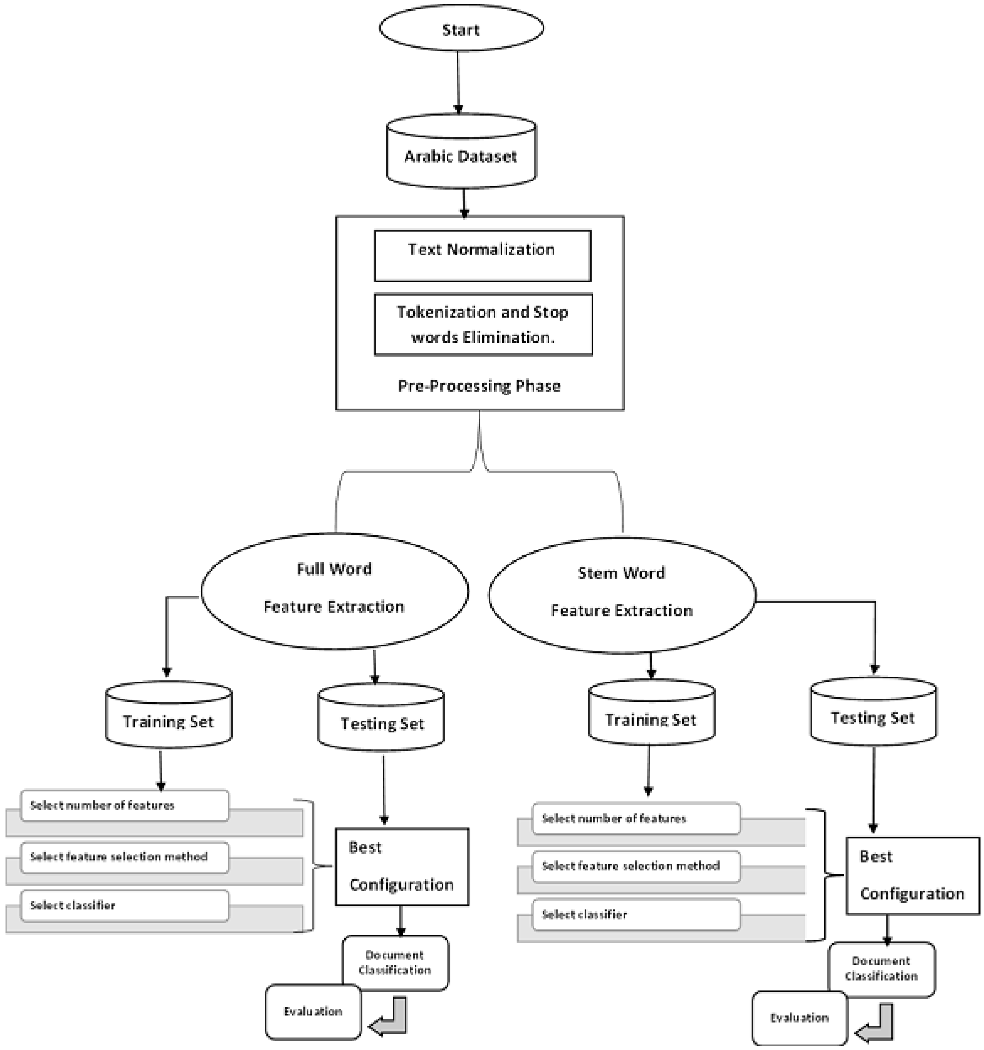

Fig. 1. The primary methodology of this research.

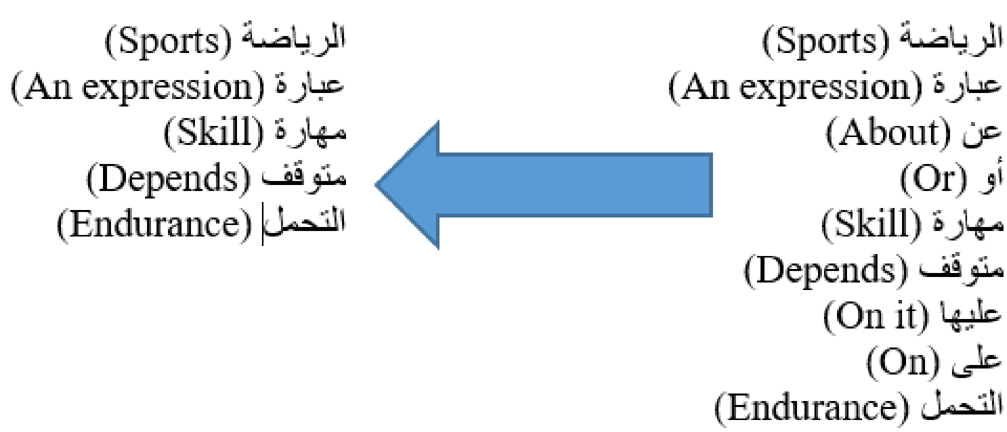

Fig. 2. Document sample words after the removal of the stop-words.

TABLE I. UTTERANCES OF AFFIXES INWORD(ليرونهم, (ليخادعونهم .

Postfix	Suffix	Root	prefix	Antefix
هم	ون	يرى	ي	ل
هم	ون	خدع	ي	ل
Pronoun meaning (they)	Termination of conjugation	يرى meaning (See) and خدع Meaning (Deceive)	A letter meaning the tense and the person of conjugation	Preposition meaning (to)

classification task can be achieved [52]. In this study, wrapper methods were examined.Fig. 3 illustrates the three primary factors used in wrapper methods for identifying the optimal feature set [53]: First, the ideal feature combination from the initial dataset is found using a search approach (for instance, an optimization technique). Second, a machine learning model (such a classifier) is used to assess the produced subset of features by comparing them to the validation dataset. Third, the dependability of the chosen feature subset is evaluated using an objective function criterion.

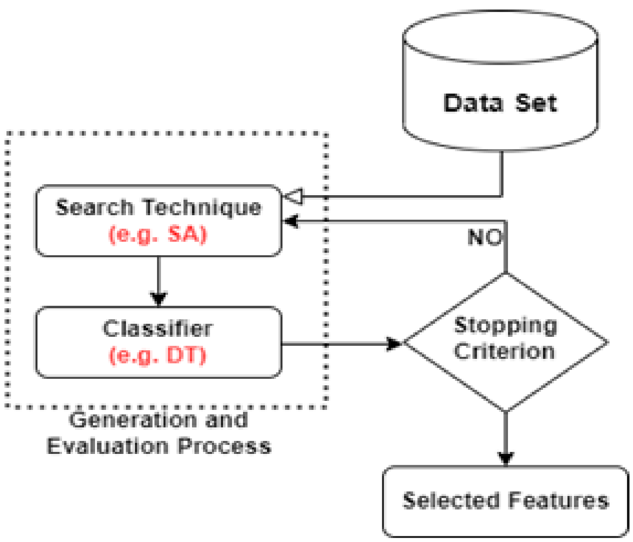

Fig. 3. Wrapper-Based feature selection method using SA and DT classifier.

As the increases in the amount of Arabic language text documents in classification, it is becoming increasingly difficult. Thus, FS techniques play a crucial part in deciding what features are the most relevant for the classification model while ignoring the redundant and the irrelevant ones. This study proposes a hybrid wrapper-based FS method for Arabic document classification, based on a combination of binary wind-driven optimization (WDO) and simulated annealing (SA) algorithms, namely BWDO-SA.

D. The Proposed BWDO-SA for Feature Selection

In this section, the proposed BAWDO-SA method is investigated, and its early stages of standard WDO, solution initialization, binary WDO, simulated annealing, and the proposed BAWDO-SA are discussed in the following:

1) Wind Driven Optimization (WDO): The Earth's atmosphere, where winds blow in an attempt to balance horizontal differences in air pressure, served as the inspiration for WDO [54]. The term "wind" refers to the horizontal movement of air, particularly in the troposphere, which is the lowest layer of Earth's atmosphere. The thickness of the troposphere varies with latitude and extends from the earth's crust to a height of about 18 km [55].

This phase aims to employ supervised WDO for feature selection. In this research, a supervised WDO approach was applied, using the accuracy of classifiers as the objective measure of performance for feature selection. The study suggested using three classifiers by combining feature selections to identify, detect document categories and assess the three classifiers' performance using WDO's supervised feature selection. In other words, this proposed method was designed to detect the most important features that would improve the classifier's performance while reducing the number of features selected. Fig. 4 shows the pseudo-code of WDO as feature selection [55].

The major strengths of the WDO high priority are that it [56]:

- Contains many transactions that facilitate the process of restricting the search field.
- Uses additional terms such as (gravitational and Coriolis force) in the equation of speed update.
- Classifies the population based on their pressure.
- Checks the boundaries to prevent any air expulsion from the search area.
- Affects problems related to both discrete and continuous value parameters [55].

2) Simulated annealing: The escape probability of a local optimum problem served as the basis for the development of the SA employing Hill Climbing-based (HC) techniques [56]. Using the simulated annealing method [57], is a heuristic solution based on the HC technique. Fig. 5 illustrates the algorithm. SA begins by establishing a random number from the available solutions and then generates neighboring solutions based on the same random number.

After generating the surrounding solutions, the algorithm verifies each one's credentials before attempting to select the most acceptable number for the next stage. With the probability defined by the Boltzmann probability value, P=e^(-θ/T), the method also accepts minimum solutions, where θ is the difference between the Solbest and Soltrial evaluations of the objective function. T is the temperature, which, in accordance with a cooling plan, drops periodically during the search phase. In [58], the authors used SA in order to solve numerous attribute reduction issues in SimRSAR. The solution took into account that each case corresponds to a certain set of attributes in SimRSAR, and a nearby arithmetic attribute can be found using the chosen attribute. $T(t+1) = 0.93 * T(t)$.

Set user parameter of the WDO: $P_{opt}, \alpha, g, RT,$ *and c and Maximum_ Iteration*

Create randomly initial population of feature selection using Eq (2, 3)

$$AirParcel\ feature = [X_1, X_2, X_3 \ldots] \quad (2)$$

$$\text{AirParcels of feature} = \begin{bmatrix} AirParcel_1 \\ AirParcel_2 \\ AirParcel_3 \\ \vdots \\ AirParcel_{Popt} \end{bmatrix} \begin{bmatrix} x_1^1 x_2^1 x_3^1 & \cdots & x_{Nfeatures}^1 \\ \vdots & \ddots & \vdots \\ x_1^{Popt} x_2^{Popt} x_3^{Popt} & \cdots & x_{Nfeatures}^{Popt} \end{bmatrix} \quad (3)$$

Randomize the velocity and the position of air parcels

Set the fitness function using Eq (4)

$$fitness = \beta\gamma_R(D) + \alpha \frac{|R|}{|N|} \quad (4)$$

Evaluate the population and find the minimum pressure value for air parcels ($P_0 = min(f(x_j))$)

Set global $P = P_0$

for all i **do**

for all j **do**

Update the velocity by Eq (5)

$$u_{new} = (1-\alpha)u_{cur} - gx_{cur} + \left(RT \left|\frac{1}{i} - 1\right| (x_{opt} - x_{cur})\right) + \left(\frac{cu_{cur}^{otherdim}}{i}\right) \quad (5)$$

Choose random dimension

Choose velocity based on random dimension

Check velocity by Eq (6)

$$u_{new}^* = \begin{cases} u_{max} & if\ u_{new} > u_{max} \\ -u_{max} & if\ u_{new} < -u_{max} \end{cases} \quad (6)$$

Update air parcel positions by Eq (4.27)

$$x_{new} = x_{cur} + (u_{new}\Delta t) \quad (7)$$

Call CMAES

Return the new set of WDO coefficients for $i+1$

Evaluate the new solution and update $P(i,j)$ **using Eq (4)**

if $P(i,j) < P0$ **then**

global $P = P(i,j)$

end if

end for

Rank air parcels and find the local optimum (*xlocal*)

end for

Rank air parcels and find the global optimum (*global*)

Return the values of *xglobal* **and** *fmin = min(f(xbest))*

End of the algorithm

Fig. 4. Pseudo-code of WDO as a feature selection.

```
ize (temperature T, random starting features selected)
 While cool_iteration <= max_iterations
      Cool_iteration = cool_iteration + 1
      Temp_iteration = 0
      While temp_iteration <- nrep
         Choose a new point from he tions.
         Compute the objective function of produced c
            cost_j = f(x^j_1t, x^j_feateatures@), j = 1,2,3, ..., P_opt
         δ= current_cost-previous_cost
         if δ < 0, approve neighbour
         else, approve with possibility exp(−δ/T)
         end while
       T = α · T (0 < α < 1)
   end while
end while
```

Fig. 5. Pseudo-code simulating annealing.

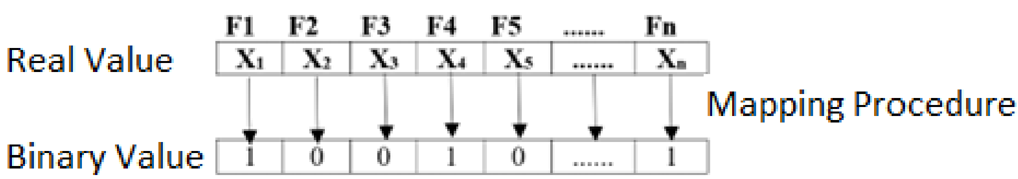

Fig. 6. Representation of the selected features.

3) Binary WDO for feature selection: The task of FS is a binary discrete problem in which features can be represented by a binary solution or vector. The values of this vector are either one for the selected features or zero for the omitted features, which are not selected, as illustrated in Fig. 6.

The basis of the WDO was initially implemented for continuous optimization tasks. Thus, to implement the WDO for the FS task, a transform function is required to convert the WDO to a binary WDO (BWDO). This study uses a sigmoid transform function, as shown in Fig. 7, to convert the solution to binary format (i.e., 0 or 1).

Eq. (2) illustrates the formula for the sigmoid function for a solution x.

$$T(X_i^j(k)) = \frac{1}{1 + exp(-X_i^j(k))} \tag{2}$$

where, $T(X^j{}_i(k))$represents the transform of the dimension j in the solution x_i (i.e., air particle) at iteration k.

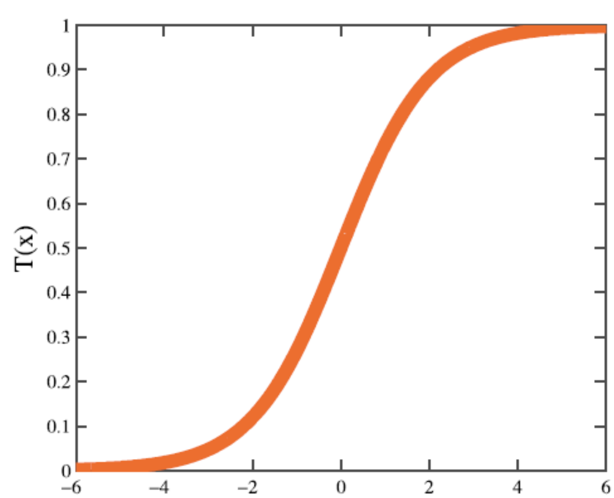

Fig. 7. Sigmoid function.

In the iteration, k, the location of each air particle xik of each pair particle i is converted into binary format using the following equation:

$$X_i^j(k) = \begin{cases} 1\, rand \;\geq\; T(X_i^j(k) \\ 0\, rand \;<\; T(X_i^j(k) \end{cases} \tag{3}$$

The above Eq. (3) is utilized in the FS approach in this study to determine the probability of changing the location of the elements in the solution.

The BWSO begins the optimization process with a randomly constructed initial population of N air particles. Each particle designates one solution with a binary vector of dimension D, which represents the candidate features, i.e., a subset of features. In this vector, dimension D characterizes the entire set of features in the original dataset. Fig. 4 shows the pseudo-code of WDO as a feature selection.

4) Evaluation using fitness function: The major objective of using the feature selection optimization approach is Reducing the number of selected features while getting the best classification accuracy. In other words, fewer selected features provide better opportunities to improve the solution classification performance.

All possible available solutions are evaluated according to the proposed witness selection procedure and the final publisher, which depends on three classifiers, namely KNN, SVM, and NB [59] to classify the answer correctly based on the amount of features chosen. To achieve the highest level of precision while maintaining an appropriate balance of all features chosen for each of the minimum solutions for feature selection, the fitness function (objective function) in Eq. (4) is utilized for each WDO as well as SA assessment techniques for agent searches.

$$fitness = \beta_{\gamma_R}(D) + \alpha\frac{|R|}{|N|} \tag{4}$$

The BWDO is a population-based metaheuristic method with high exploration performance. However, the main limitation of the BWDO is the degradation of its exploitation capabilities. This is because it employs a blind operator to enhance the local area surrounding the best solutions discovered thus far without regard for the quality of the present solutions. This study introduces an innovative FS method based on the hybridization of BWDO and SA, known as BWDO-SA. This method is applied to the FS task in the domain of Arabic text classification. The SA method, which is a local search algorithm, is utilized within the BWDO to replace the blind exploitation phase. Specifically, the SA seeks the best solution in the vicinity of the best-known and randomly selected solutions.Fig. 8 shows the flowchart of the proposed Hybrid of BWDO-SA algorithm.

E. Classification

We investigated NB, DT and KNN, which are frequently used for text classification and proven to give satisfactory results, to test the effectiveness of the proposed methodology [60]. Additionally, these classifiers meet nearly all of the difficulties encountered in this paper thanks to the following qualities: In supervised algorithms, the categorical structure of a given database is taken to be known beforehand. For the supervised algorithms to map documents into predefined labels, they need a set of labeled documents. As noted earlier, it can be difficult to determine the exact class and naming of training sets, especially in large databases. Therefore, the most popular supervised algorithms, including DT, KNN, and NB, will be the focus of this section.

1) KNN: KNN has been established as a well-liked instance-based learning method and has shown efficacy in a number of text classification tasks [60]. The algorithm flow can be summarized as follows: From the given training documents, KNN first determines the k-nearest neighbors. The flow of the algorithm can be summarized as follows: First, the KNN identifies the k-nearest neighbors from the given training documents. Second, based on the category labels of these neighbors, the category of the test document is established. The proven k-nearest method usually sets the test document with the name of the most common class. Each neighbor's contribution to the weighted KNN is determined by how close it is to the test document. This is an extension of the regular KNN. To determine the class score of a document, the similarities between neighbors in each class are added. Or to put it another way, the category score for document x can be represented as in Eq. (5):

$$score(c_j, x) = \sum\nolimits_{d_i \subset N(x)} \cos\left(x_i, d_j\right), y(d_i, c_j) \tag{5}$$

The training document is =d_i , a collection of the k training documents that are closest to x=N(x), similarity between x and d_i =cos(x,d_i) , value of a=1 in function y(di,c_j) if belongs to category, and otherwise 0. The results of both evaluation methods are presented in the results part of this study, which examined the original KNN as well as its weighted version by varying the values of the parameter k.

2) Naïve Bayesian (NB): NB was constructed as a probability-based model that uses the joint probability of terms and categories in a test document in order to calculate the probability of categories assigned to that document [60]. It is the classifier's assumption that all words in each class are conditionally independent of all words in the other class that gives rise to the "naive" feature of the classifier. In contrast to non-NB classifiers, computational procedures can be made simpler by learning parameter values for each word separately based on the idea of independence.

As per [61], Polynomial modeling and multivariate Bernoulli model were both major models used for text classification using NB. Bayes' rule is used by the two models to accomplish the categorization of documents [60] [see Eq. (6)]:

$$P(c_j|d_j) = \frac{P(c_j)\; P(d_j|c_j)}{d_i} \tag{6}$$

where, d_i= test document, C_j= class. Given d_i is the test document, c_j posterior probability for each category's, i.e., , $P(c_j|d_j)$and d_i assigned to the category with the highest probability. The calculation of $P(c_j|d_j)$ calls for an estimation

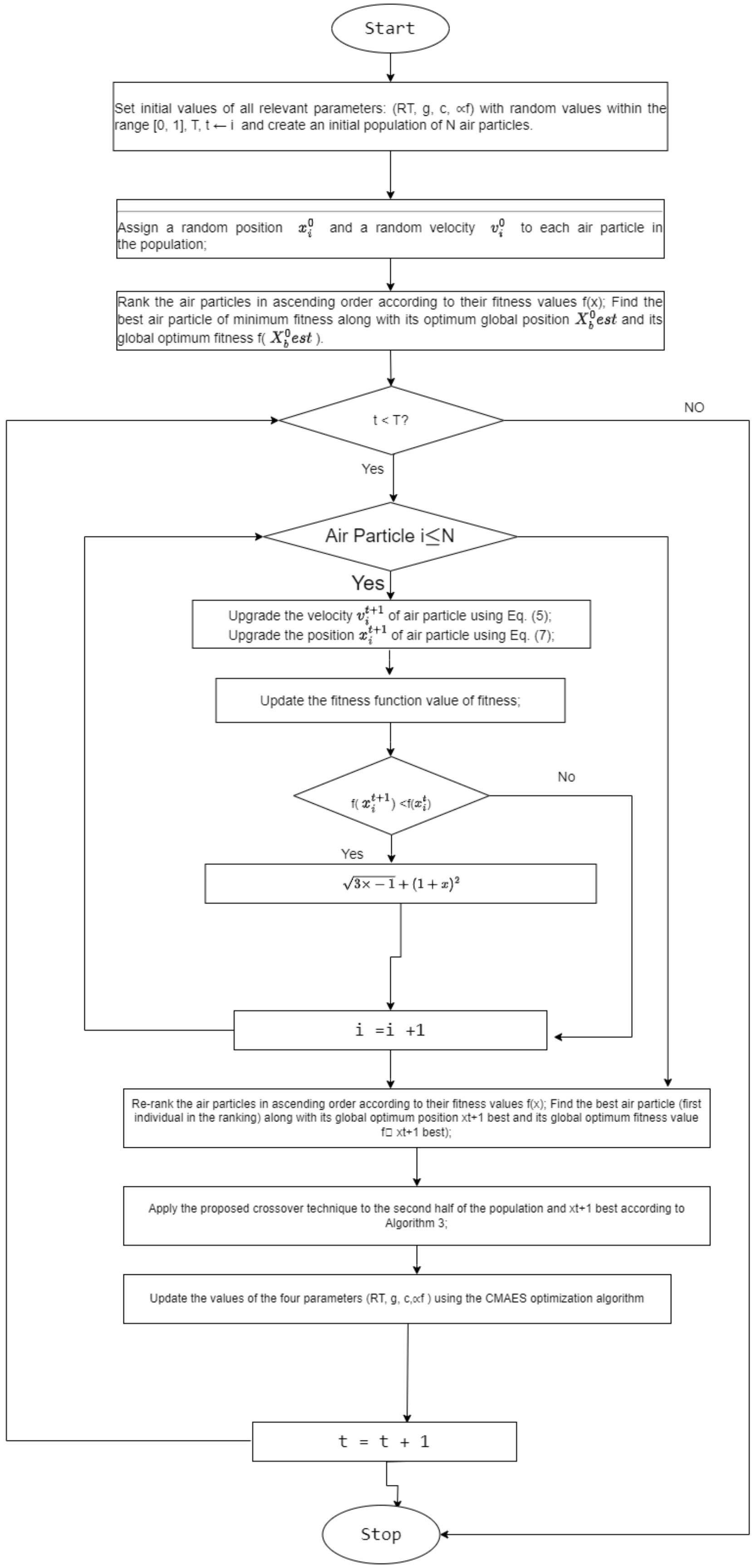

Fig. 8. Flowchart of the proposed hybrid of BWDO-SA algorithm.

of the P(Cj) and $P(c_j|d_j)$based on training set. It should be noted that P(Cj) is the same for each category. Hence, it can be eliminated from the calculation. In doing so, the prior probability of the category P(Cj) will be thus estimated as Eq. (7):

$$p\hat{c_j} = \frac{\sum_{i=1}^{N} y(d_i,\ c_j)}{N} \tag{7}$$

where, the amount of training set =N and thus y(di, $c_{j)}$estimated as Eq. (8):

$$y(d_i, c_j) = \begin{cases} d_i \in c_j \\ 0\ otherwise \end{cases} \tag{8}$$

Consequently, the percentage of training set which belongs to Cj is employed to determine the a priori probability of a class Cj. The multinomial model and multivariate Bernoulli model estimate $P(c_j|d_j)$ the parameters in different ways.

3) Decision Tree (DT): A Decision Tree text classifier is a non-parametric supervised learning method, represented as a tree in which the internal nodes are named terms and the leaves are tagged with classes [62]. The decision tree is one of the most popular and successful learning algorithms due to its many features, including lack of parameters, simplicity, ease of understanding, and ability to deal with different types of data. During the learning phase, a decision tree is built by recursively splitting the data based on attribute values and class labels, resulting in a large search space.

A decision tree is a greedy, top-down algorithm and iterative process from the training data entered until it reaches an empty tree. The feature that contains more information about the content is better defined to divide and is considered as the root division feature, Subsequently, subsets satisfying the partitioning features values are generated from the training data. This algorithm works on every subset repeatedly until all instances of a subset belong to the same category [63]. There are many applications in which the decision tree is configured, including its ability to deal with the problem of classification of incomplete data very well . In addition, it can replace statistical procedures to obtain data, extract text, find missing data, enable search engines, and use in many medical fields.

IV. Experiments

A. DataSet

The collection of data contains a set that was created to evaluate the creation of Arabic classifiers by extracting Arabic text, for which 800 documents have been chosen as a portion of TREC 2001 [64]. Realism and Criteria For such a set, TREC queries were developed: TREC 2001 included 25 queries, and relevant appropriate judgments were created using the grouping technique for the set of queries. Accordingly, Table II definition of the classifier's component of TREC includes four categories and the number of documents.

TABLE II. Summary Description of Arabic Data Set

Classes	# documents
1	200
2	200
3	200
4	200

B. Parameter Setting

Regarding the available optimal approach, it could be employed based on the three classification methods of the algorithm, considering this special method. For cross-validation, entire datasets are split. In conclusion, the validations are also categorized, using the evaluation strategy from [65]. While K-fold cross-validation and K-5 folds are utilized for both training and validation, the other folds are employed for testing.

The overall iterations of M could be applied to this process. After that, each optimizer unit could assess the times of K*M per dataset. To obtain the data used for training, the size of all parameters should be determined. Optimum results can be achieved when the number of repetitions reaches 50. The population size in this case should be 28. Each algorithm can be used for five total iterations. For feature selection, random feature selection should be used.

This section explores the algorithmic solution under several settings using 5 vital parameters: P_opt (population size), g (gravitational constant), RT coefficient, C (coriolis effect), and α (constants alpha). The vertical force is the gravitational force (g) directed from highest pressure to lowest pressure. The constant universal gas is represented by RT, where both R and T represent temperature. The constants C and α are utilized in velocity, while P_opt represents the number of Air Parcels. The gravitational force coefficient, g, and the friction coefficient, α, should only vary in the range [0, 1]. However, C and RT might include a wider range of values, such as [0, 5]. This section focused on the effects of variations in one of the parameters. This section focuses on the effects of variations in one of the parameters. Specifically, the three scenarios presented in Table III are evaluated. Furthermore, empirical studies have shown that the best results are achieved when there is a linear relation between P_opt and the number of features. Each scenario is evaluated for 50 iterations, with the cost value of the solution remaining as the fitness function value. The lowest fitness function value [Equation (8)] is considered the best fitness. Among the nine scenarios, the lowest fitness function value was related to scenario S2.3. The evaluation system used for wind-driven feature selection, as indicated in Section 3.3.3 (dmax = 1E03). Accordingly, case S2.3 was selected for conducting tests in this section, while other scenarios had higher fitness function scores. The parameters were set to P_{opt}=6, g=0.2, RT =3, C=0.4 and alph=0.4.

C. Performance Evaluation

The labeled test of the document component often determines the external quality measure, as part of its methodology, the classifiers that are produced are compared to the classes that have been labeled, and the extent to which documents

TABLE III. Parameter Settings of Arabic Data Sets

Scenario	Popt	g	RT	C	A
S1.1	2	0.1	1	0.2	0.1
S1.2	4	0.1	1	0.2	0.1
S1.3	6	0.1	1	0.2	0.1
S2.1	2	0.2	3	0.4	0.4
S2.2	4	0.2	3	0.4	0.4
S2.3	**6**	**0.2**	**3**	**0.4**	**0.4**
S3.1	2	0.3	5	0.6	0.8
S3.2	4	0.3	5	0.6	0.8
S3.3	6	0.3	5	0.6	0.8

from the same class are assigned to the same class is measured. In this work, the employed external quality measure is accuracy, as it is a commonly used quality measure in text mining, along with the number of features. The confusion matrix, also known as the evaluation measures in classification issues, is often defined from a matrix utilizing the number of instances that are properly and incorrectly identified for each class. The confusion matrix for a binary classification issue with just positive and negative classes was displayed in Table IV.

TABLE IV. Confusion Matrix

	The expected Class	
Real class	positive	real class
P	TP	TN
N	FP	FN

There, it is shown how FP, FN, TP, and TN is:

- False Positives (FP): Negative events that are incorrectly seen as positive.
- False Negatives (FN): Positive occurrences that are incorrectly interpreted as bad.
- True Positives (TP): The instances of positivity that are accurately anticipated to be positive.
- True Negatives (TN): The negative occurrences that are accurately predicted as negative.

Percentage Accuracy (ACC) is an evaluation measure widely used in practice that evaluates the effectiveness of a classifier based on the percentage of accurate predictions it makes. Thus, the ACC of the classifier is calculated as Eq. (9):

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} * 100 \tag{9}$$

V. Results and Discussion

The planned contribution of this section is to respond to the research queries raised by the study. The purpose of this section is to achieve the study objective of evaluating the performance of Arabic classifiers NB, DT, and KNN in terms of classification performance, whether with or without selecting features, as well as with and without stemming of words. Additionally, the proposed BWDO-SA algorithm is employed for feature selection to enhance the performance of Arabic classifiers by reducing the dimensions of their features. Finally, the proposed framework of the Arabic classifier evaluated by accuracy and the amount of features. The results of the proposed framework focus on the reduction of the number of features using the optimization FS approach and their enhancement. The results presented in this section highlight the significant implementations using a number of features and the accuracy as evaluation metrics. Evaluation methodology used is also explained, followed by a discussion of the obtained outcomes. In conclusion, The outcomes of the combined WDO and SA techniques are shown in this section. Table V and Table VI present the results of the three classifiers' accuracy with and without the stemmer as a pre-processing step. The results show that DT achieved higher accuracies compared to NB and K-NN in both cases. Without the stemmer, DT achieved an accuracy of 90%, which was higher than NB (33.83%) and K-NN (26.11%). With the stemmer, DT achieved an accuracy of 93%, higher than NB (35%) and K-NN (26.36%). When comparing between the use of stemmer and without stemmer, it can be observed that the stemmer resulted in better accuracy for the classifiers. In terms of feature reduction, without the stemming, the feature count evaluation showed that reducing the feature count was not effective, whereas with the stemmer, better reduction of the number of features was achieved, as shown in Fig. 9 and Fig. 10.

Fig. 11 and Fig. 12 displays the results of integrating various metaheuristic algorithms as feature selection methods with the DT classifier, in terms of precision and the total amount of features, both when a stemmer is used as a pre-processing step and when it is not. The outcomes show that our suggested algorithm, WDO with DT, and SA with DT using the stemmer achieved higher precision compared to other feature selection algorithms. Both WDO with DT and SA with DT obtained the same precision value of 97.01%, which was higher than WOA with DT (94.7%), GA with DT (94.5%), PSO with DT (91.79%), HS with DT (91.5%), and SA with DT (87.2%). Without a stemmer as a preprocessing step with feature selection, WDO with DT and GA with DT achieved the same precision value of 95.02%, higher than WDO with DT (92.2%). In addition, WOA with DT and HS with DT obtained the same precision value of 90.2%, PSO with DT (90%), and SA with DT (85.46%).

Regarding the number of features, using DT classifier, WDO as a feature selection technique method, and using a stemmer as a pre-processing step, achieved outstanding results by significantly reducing the number of features. Specifically, WDO with DT reduced the features to 4231 out of 4616, which was lower compared to our proposed algorithm WDO and SA with DT, where it reduced the features to 8909. HS with DT had 14175 features, SA with DT had 14937 features, PSO with DT had 34787 features, and GA with DT and WOA with DT had the same value of 46167 features without any change in the number of features. On the other hand, the findings revealed that our proposed algorithm WDO and SA with DT, without using a stemmer as a preprocessing step, succeeded in reducing the amount of features and achieved good results. When compared to other algorithms, it produced 7236 out of 91756 features, which was the best result. The distribution of results was as follows: WDO with DT reduced to 9289

TABLE V. Three Classifier Without Stemmer

Classifier	#of features before selected	# of features before selected	Acc.
DT	**91756**	**91756**	**90.298**
KNN	91756	91756	26.119
NB	91756	91756	33.830

TABLE VI. Three Classifier With Stemmer

Classifier	# of features after selected	# of features before selected	Acc.
DT	46167	46167	**93.781**
KNN	46167	46167	26.368
NB	46167	46167	35.074

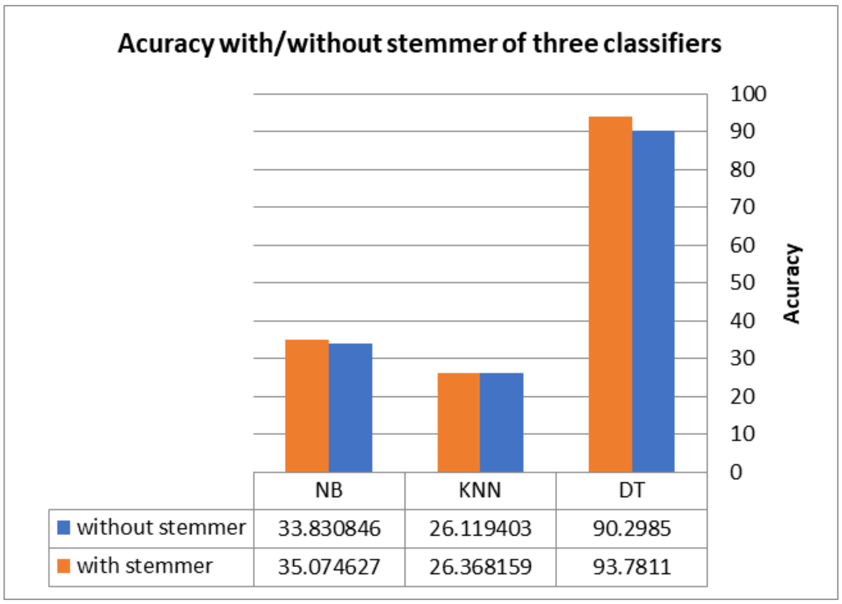

Fig. 9. Accuracy with/without stemmer of three classifiers.

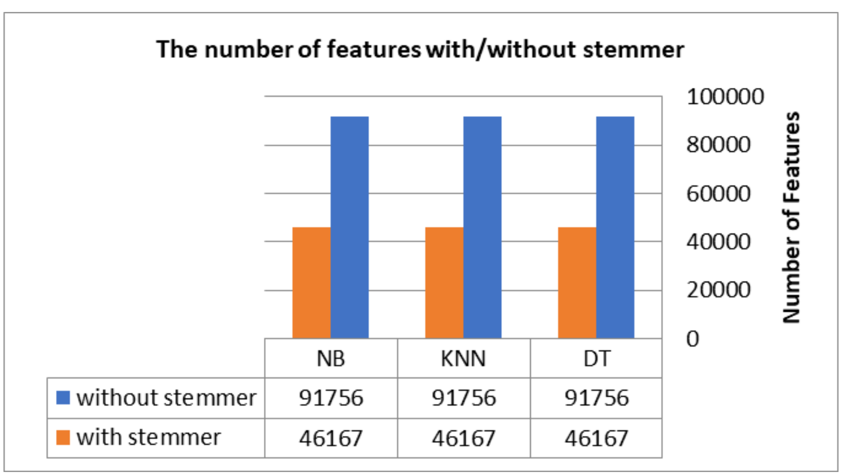

Fig. 10. Number of features with/without stemmer.

features, SA with DT to 19834 features, HS with DT to 28230 features, GA with DT to 45559 features, PSO with DT to 45711 features, and WOA with TD to 45989 features (see Table VII and Table VIII).

When comparing the integration of many metaheuristic algorithms as feature selection methods with DT, with and without stemmer, in terms of accuracy, the algorithms with DT classifier and stemmer achieved better accuracy than those without stemmer as show in Fig. 11.

However, the number of features is another measure for evaluation, It showed that the performance of DT classifier algorithms in feature count reduction was worse when they did not include a stemmer tool, while the algorithms with DT classifier and stemmer performed better, as shown in Fig. 12.

Fig. 13 and Fig. 14 present the results of hybridizing metaheuristic algorithms as feature selection methods with the KNN classifier, in terms of features size and accuracy, with using stemmer and without it as preprocessing. When stemmer was used, it is clear that our suggested algorithm, WDO and SA with KNN classifier, achieved better results compared to other algorithms. Specifically, WDO and SA with KNN obtained an accuracy that was 70.6% higher than WDO and SA with KNN, and HS with KNN, which had the same accuracy of 67.6%. GA with KNN and PSO with KNN had the same accuracy of 25.12%, WOA with KNN had an accuracy of 24.37%, and SA with KNN had an accuracy of 21.64%. However, when without stemmer was used as preprocessing with some optimization algorithms as feature selection, the results revealed that WDO with KNN classifier achieved an accuracy that was 67.6% higher than WDO and SA with KNN (61.9%), HS with KNN (41.5%), GA with KNN (24.6%), PSO with KNN (24.3%), WOA with KNN (24.1%), and SA with KNN (21.6%).

Regarding the size of features, the suggested algorithm WDO and SA with KNN classifier, when using stemmer as preprocessing, demonstrated superior performance as the results showed a significant reduction in the number of features. Specifically, SA with KNN and WDO reduced the features size to 519 out of 46167, which was lower than WDO with KNN (582 features), HS with KNN (633 features), SA with KNN (19528 features), PSO with KNN (23083 features), GA with KNN (23099 features), and WOA with KNN (23143 features). On the other hand, when stemmer was not used as preprocessing, the results showed that WDO and SA with KNN still provided good results in reducing the number of features, with some minor differences. Specifically, WDO and SA with KNN reduced the number of features to 472 out of 91756, while HS with KNN reduced to 243 out of 91756 features, WDO with KNN to 320, SA with KNN to 43859, PSO with KNN to 45454, WOA with KNN to 45939, and GA with KNN to 45994 (see Table IX and Table X).

When comparing different metaheuristic algorithms as feature selection with KNN classifier, using stemmer and without stemmer as pre-processing, it is observed that the algorithms with KNN classifier and stemmer achieved better accuracy, as shown in Fig. 13. However, the number of features is another evaluation measure, which indicates that the results without stemmer were worse in terms of reducing features size, while using stemming is more effective, as results in Fig. 14.

Fig. 15 and Fig. 16 present the results of hybridizing metaheuristic algorithms as feature selection with NB classifier for accuracy and number of features, both with and without the use of a stemmer for preprocessing. With stemmer, it is evident that the proposed algorithm WDO and SA with NB achieved better outcomes in comparison to other algorithms in terms of accuracy. Specifically, WDO and SA with NB showed 38.3% higher accuracy than GA with NB (37.56%), PSO with NB (36.3%), WDO with NB and SA with NB (34.3%), HS with NB (34.07%), and WOA with NB (24.37%).

On the other hand, without stemmer as preprocessing, our proposed algorithm WDO and SA with NB outperformed other algorithms, achieving an accuracy of 37.8%. The results for other algorithms were as follows: GA with NB (33.58%), PSO with NB (33.3%), WDO with NB and HS with NB (33.08%), SA with NB (31.34%), and WOA with NB (31.09%).

Regarding the number of features, the WDO and SA feature selection with NB classifier using stemmer performed well, as evidenced by the results, and reduced the number of features to 5354 out of 46167, which was lower than other algorithms. The results for other algorithms were as follows: HS with NB (15119), WDO with NB (17730), GA with NB (22965), SA with NB (23124), PSO with NB (23162), and WOA with DT (23162). On the other hand, the results without using stemmer showed that the WDO and SA with NB outperformed other algorithms, achieving 6899 features out of 91756, which was less than each of WDO with NB (25589), SA with NB (44358), PSO with NB (45666), GA with NB (45821), WOA with NB (46143), and HS with NB (91756) [see Table XI and Table XII].

In general, when comparing different metaheuristic algorithms as feature selection with NB classifier with and without stemmer, the algorithms with NB classifier with stemmer showed better accuracy, as depicted in Fig. 15. Moreover, the features size is another evaluation measure, which revealed that the results without stemmer were less effective in reducing the features size in comparison to those with stemmer, as shown in Fig. 15 and Fig. 16.

The three classification algorithms, namely NB, KNN, and DT, used in this study were discussed in detail. Additionally, the proposed algorithm, Wind Driven Feature Selection (WDFS), and simulated annealing (SA) were explained in full, including the modifications made to use them as feature selection techniques and the parameters employed in our proposed algorithm. Finally, the results of the research experiments were given, showcasing the impact of using stemmer and feature selection with the three classifiers on our dataset. Furthermore, a comparison was made between the results of WDFS and SA with the three classifiers separately, with/without using stemmer. The findings revealed that using WDFS and SA with stemmer positively affected the accuracy and number of features in classification, leading to improved classifier accuracy and reduced feature count.

VI. Conclusion and Future Work

Arabic text classification is a difficult computational task due to the high content of natural Arabic language text on the Internet. FS is a critical step in the Arabic TC process, as there are thousands of possible features to consider. This study

TABLE VII. DO and SA With DT Algorithms Without Using a Stemmer

DT	# of features after selected	# of features before selected	Acc.
GA+ DT	91756	45559	0.9502
HS+ DT	91756	28230	0.9029
PSO+ DT	91756	45711	0.9004
WDO+ DT	91756	9289	0.9228
SA+ DT	91756	19834	0.8546
WOA+ DT	91756	45989	0.9029
WDO+SA+DT	91756	**7236**	**0.9502**

TABLE VIII. WDO and SA With DT Algorithms With Using a Stemmer

Classifier	# of features after selected	# of features before selected	Acc.
GA+DT	46167	46167	0.945274
HS+DT	46167	14175	0.915423
PSO+DT	46167	34787	0.917910
WDO+DT	46167	**4231**	**0.9701**
SA+DT	46167	14937	0.872698
WOA+DT	46167	46167	0.947761
WDO+SA+DT	46167	**8909**	**0.9701**

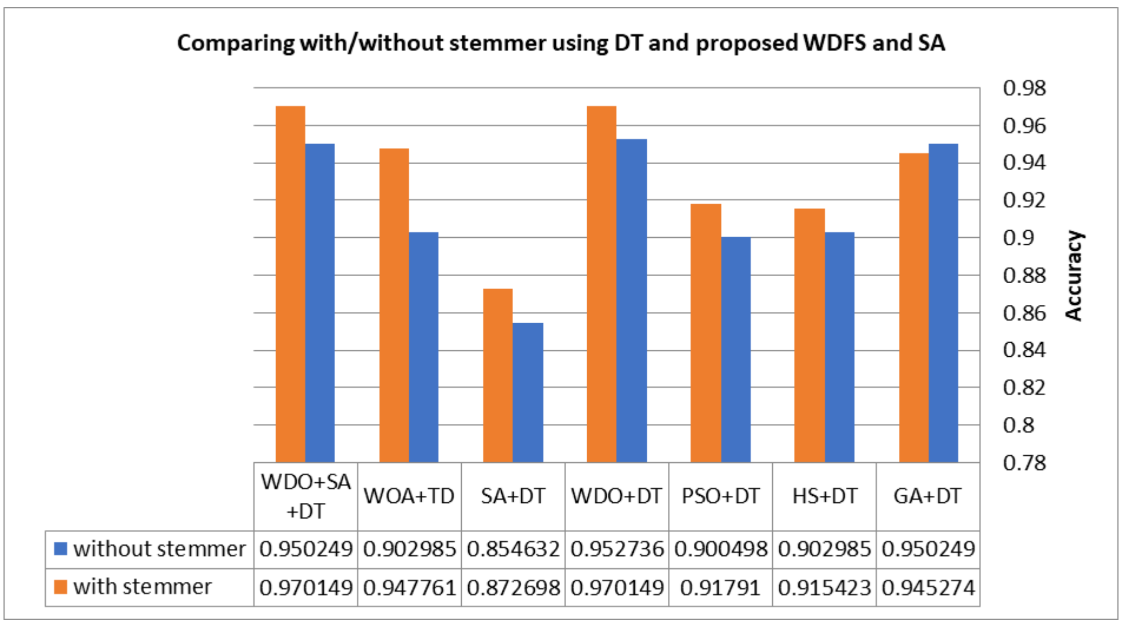

Fig. 11. Comparing with/without stemmer using DT and proposed WDFS and SA.

TABLE IX. WDO and SA With KNN Algorithms Without Using Stemmer

KNN	# of features after selected	# of features before selected	Acc.
GA+KNN	91756	45994	0.24626
HS+KNN	91756	**243**	**0.41542**
PSO+KNN	91756	45454	0.24378
WDO+KNN	91756	**320**	**0.67661**
SA+KNN	91756	43859	0.21641
WOA+KNN	91756	45939	0.24129
WDO+SA+KNN	91756	**472**	**0.61940**

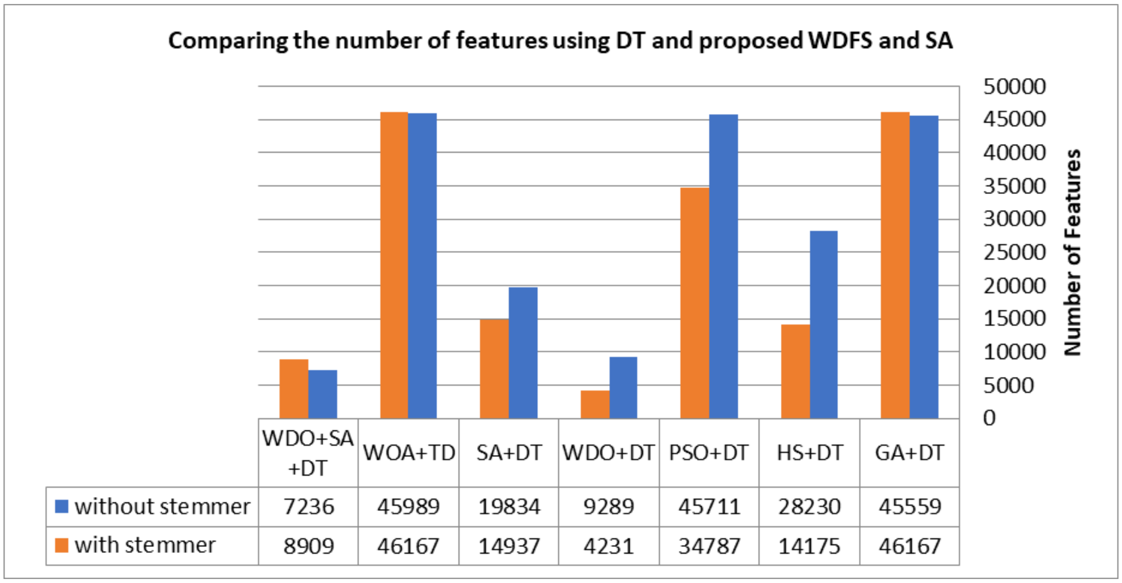

Fig. 12. Comparing the number of features using DT and proposed WDFS and SA.

TABLE X. WDO and SA With KNN Algorithms With Using Stemmer

KNN	# of features after selected	# of features before selected	Acc.
GA+KNN	46167	23099	0.2512
HS+KNN	46167	**633**	**0.6766**
PSO+KNN	46167	23083	0.2512
WDO+KNN	46167	**582**	**0.6766**
SA+KNN	46167	19528	0.2164
WOA+KNN	46167	23143	0.2437
WDO+SA+KNN	46167	**519**	**0.7064**

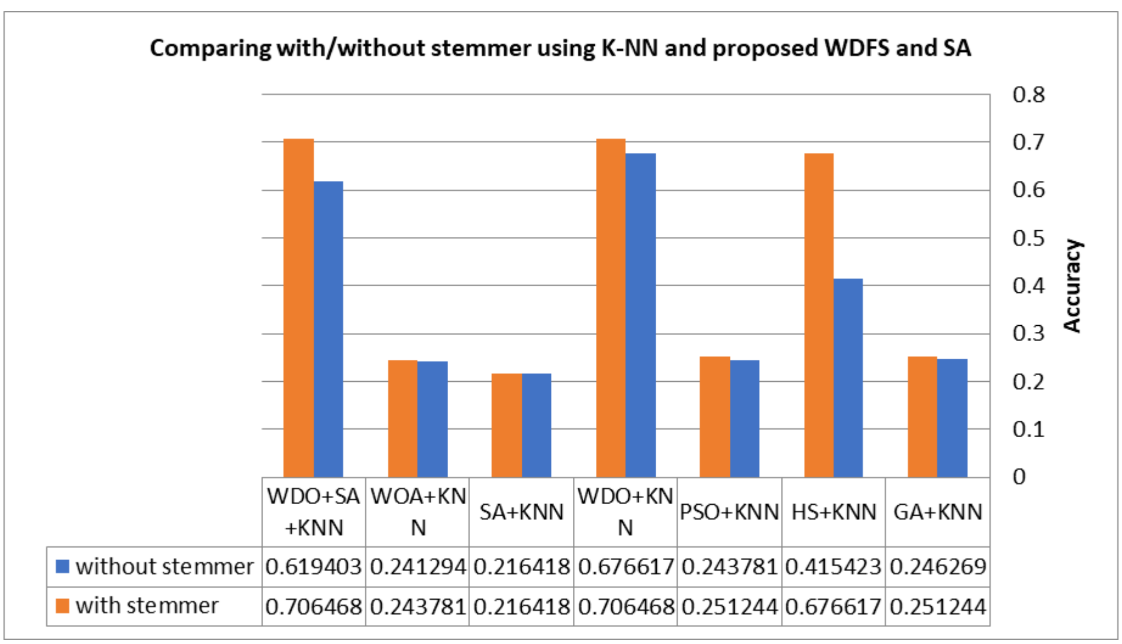

Fig. 13. Comparing with/without stemmer using K-NN and proposed WDFS and SA.

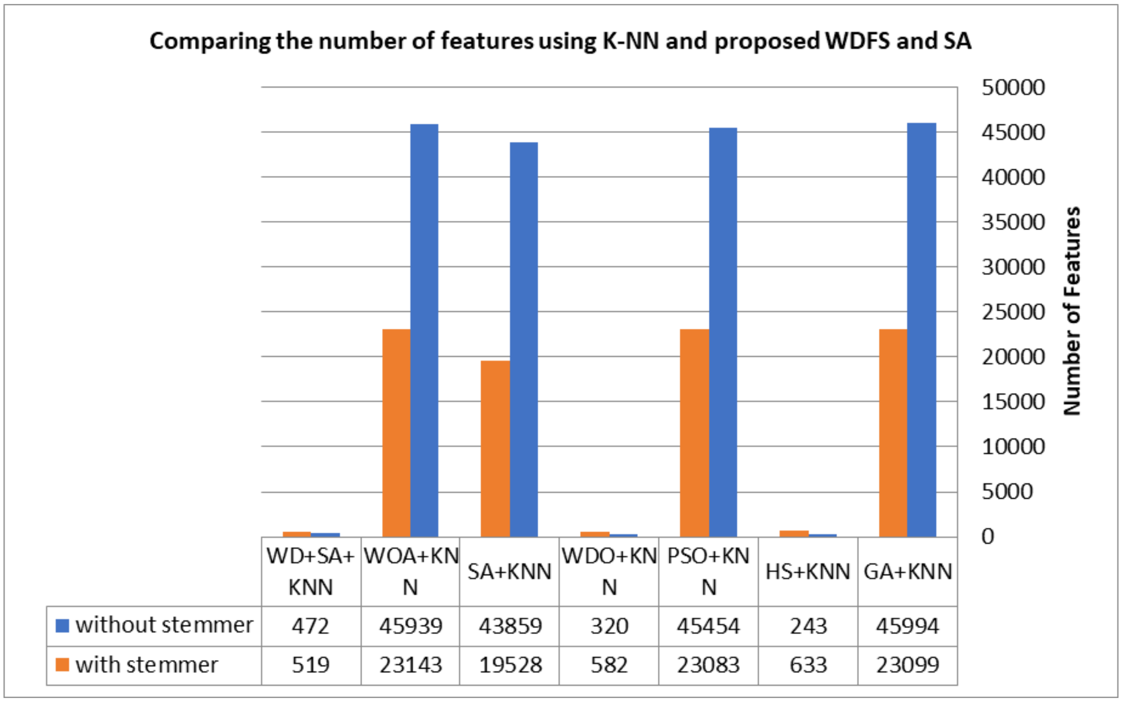

Fig. 14. Comparing the features size using proposed WDFS and SA and K-NN.

TABLE XI. WDO and SA with NB Algorithm Without Using Stemmer

NB	Number of features after selected is	Number of features before selected is	Acc.
GA+NB	91756	45821	0.335821
HS+NB	91756	91756	0.33085
PSO+NB	91756	45666	0.333333
WDO+NB	91756	25589	0.330846
SA+NB	91756	44358	0.313433
WOA+NB	91756	46143	0.310945
WDO+SA+NB	91756	**6899**	**0.378109**

TABLE XII. WDO and SA with NB Algorithm With Using Stemmer

NB	Number of features after selected is	Number of features before selected is	Acc.
GA+NB	46167	22965	0.375622
HS+NB	46167	15119	0.340796
PSO+NB	46167	23124	0.363184
WDO+NB	46167	17730	0.343284
SA+NB	46167	23124	0.343284
WOA+NB	46167	23162	0.243781
WDO+SA+NB	46167	**5354**	**0.383085**

introduces two wrapper FS methods based on the WDO algorithm for Arabic text classification. The first method uses the binary version of WDO, while the second method improves the binary WDO by hybridizing it with Simulated Annealing (SA) to create WDO-SA. The primary goal of this hybridization is to optimize the best found binary WDO solution after each major iteration. Three famed classifiers, K-NN, SVM, and Naive Bayes, it was used to assess the applicability of each subset of features to Arabic text classification. The proposed methods were validated on a complex Arabic benchmark dataset. The suggested WDO-SA with SVM classifier outperformed other approaches for the FS task in Arabic text categorization, according to the empirical study's findings. Furthermore, investigation results show that incorporating SA into binary WDO methods enhances the search ability to find feasible regions and exploit them. In the future, exploring ensemble machine learning with individual classifiers and investigating different hybridization optimization methods for Arabic text classification could be worthwhile research directions.

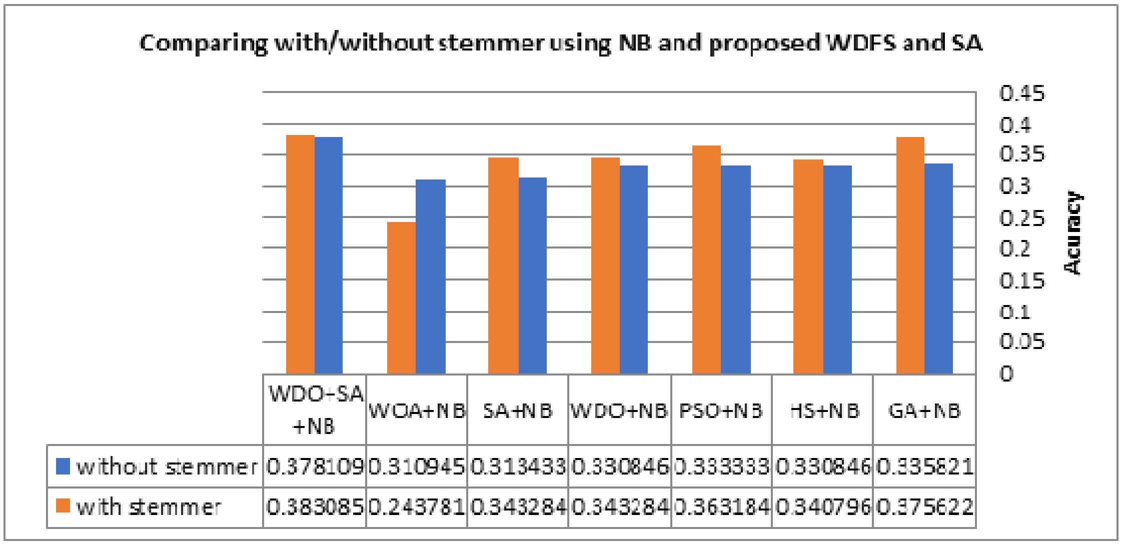

Fig. 15. Comparing with/without stemmer using NB and proposed WDFS and SA.

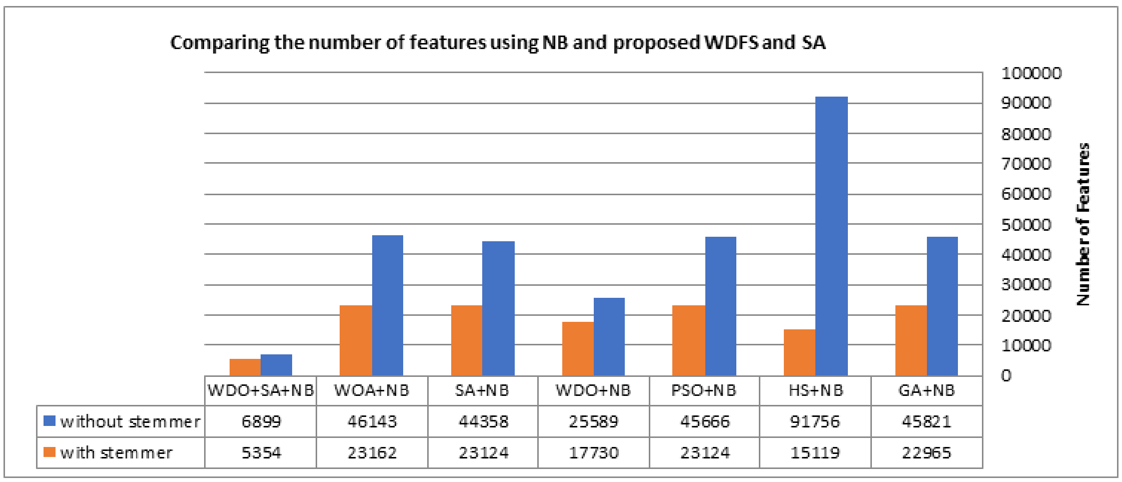

Fig. 16. Comparing the number of features using NB and proposed WDFS and SA.

References

[1] E. G. Dada, J. S. Bassi, H. Chiroma, A. O. Adetunmbi, and O. E. Ajibuwa, "Machine learning for email spam filtering: review, approaches and open research problems," *Heliyon*, vol. 5, no. 6, pp. 1802–1802, 2019.

[2] A. Wahdan, S. Hantoobi, S. A. Salloum, and K. Shaalan, "A systematic review of text classification research based on deep learning models in Arabic language," *Int. J. Electr. Comput. Eng*, vol. 10, no. 6, pp. 6629–6643, 2020.

[3] A. Qadi, L. E. Rifai, H. Obaid, S. Elnagar, and A, "Arabic text classification of news articles using classical supervised classifiers," *2nd International Conference on New Trends in Computing Sciences (ICTCS*, pp. 1–6, 2019.

[4] D. Alsaleh and S. Larabi-Marie-Sainte, "Arabic text classification using convolutional neural network and genetic algorithms," *IEEE Access*, vol. 9, pp. 91 670–91 685, 2021.

[5] A. Omar, T. M. Mahmoud, T. Abd-El-Hafeez, and A. Mahfouz, "Multi-label arabic text classification in online social networks," *Information Systems*, vol. 100, pp. 101 785–101 785, 2021.

[6] E. Al-Shammari and J. Lin, "A novel Arabic lemmatization algorithm," *Proceedings of the Second Workshop on Analytics for Noisy Unstructured Text Data*, pp. 113–118, 2008.

[7] H. Chantar, M. Mafarja, H. Alsawalqah, A. A. Heidari, I. Aljarah, and H. Faris, "Feature selection using binary grey wolf optimizer with elite-based crossover for Arabic text classification," *Neural Computing and Applications*, vol. 32, pp. 12 201–12 220, 2020.

[8] S. Fodeh, B. Punch, and P. N. Tan, "On ontology-driven document clustering using core semantic features," *Knowledge and Information Systems*, vol. 28, pp. 395–421, 2011.

[9] G. Jain, M. Sharma, and B. Agarwal, "Spam detection on social media using semantic convolutional neural network," *International Journal of Knowledge Discovery in Bioinformatics (IJKDB)*, vol. 8, no. 1, pp. 12–26, 2018.

[10] S. Saha, S. Dasgupta, and S. K. Das, "Spam mail detection using data mining: A comparative analysis," *Smart Intelligent Computing and Applications: Proceedings of the Second International Conference on SCI 2018*, vol. 1, pp. 571–580, 2019.

[11] F. Wang, T. Xu, T. Tang, M. Zhou, and H. Wang, "Bilevel feature extraction-based text mining for fault diagnosis of railway systems," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 1, pp. 49–58, 2016.

[12] A. El-Mageed, A. A. Gad, A. G. Sallam, K. M. Munasinghe, K. Abohany, and A. A, "Improved binary adaptive wind driven optimization algorithm-based dimensionality reduction for supervised classification," *Computers & Industrial Engineering*, vol. 167, pp. 107 904–107 904, 2022.

[13] F. Sebastiani, "Machine learning in automated text categorization," *ACM Computing Surveys (CSUR)*, vol. 34, no. 1, pp. 1–47, 2002.

[14] Y. A. Alhaj, J. Xiang, D. Zhao, M. A. A. Al-Qaness, M. A. Elaziz, and A. Dahou, "A study of the effects of stemming strategies on arabic document classification," *IEEE Access*, vol. 7, pp. 32 664–32 671, 2019.

[15] B. Masand, G. Linoff, and D. Waltz, "Classifying news stories using memory based reasoning," *Proceedings of the 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 59–65, 1992.

[16] K. Nigam, J. Lafferty, and A. Mccallum, "Using maximum entropy for text classification," *IJCAI-99 Workshop on Machine Learning for Information Filtering*, vol. 1, pp. 61–67, 1999.

[17] H. Drucker, D. Wu, and V. N. Vapnik, "Support vector machines for spam categorization," *IEEE Transactions on Neural Networks*, vol. 10, no. 5, pp. 1048–1054, 1999.

[18] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up? Sentiment classification using machine learning techniques," *arXiv preprint cs/0205070*, 2002.

[19] Y. A. Alhaj, W. U. Wickramaarachchi, A. Hussain, M. A. A. Al-Qaness, and H. M. Abdelaal, "Efficient feature representation based on the effect of words frequency for Arabic documents classification," *Proceedings of the 2nd International Conference on Telecommunications and Communication Engineering*, pp. 397–401, 2018.

[20] H. Gupta, M. S. Jamal, S. Madisetty, and M. S. Desarkar, "A framework for real-time spam detection in Twitter," *10th International Conference on Communication Systems & Networks (COMSNETS)*, pp. 380–383, 2018.

[21] W. Chebil, M. Wedyan, M. Alazab, R. Alturki, and O. Elshaweesh, "Improving semantic information retrieval using multinomial naive Bayes classifier and Bayesian networks," *Information*, vol. 14, no. 5, p. 272, 2023.

[22] P. Parveen and P. Halse, "Spam mail detection using classification," *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 5, pp. 347–349, 2016.

[23] S. Deepalakshmi and T. Velmurugan, "Empirical study of feature selection methods for high dimensional data," *Indian Journal of Science and Technology*, vol. 9, no. 39, pp. 1–6, 2016.

[24] M. Diale, C. V. D. Walt, T. Celik, and A. Modupe, "Feature selection and support vector machine hyper-parameter optimisation for spam detection," *Pattern Recognition Association of South Africa and Robotics and Mechatronics International Conference*, pp. 1–7, 2016.

[25] Y. Liu, Y. Wang, L. Feng, and X. Zhu, "Term frequency combined hybrid feature selection method for spam filtering," *Pattern Analysis and Applications*, vol. 19, pp. 369–383, 2016.

[26] G. Raho, R. Al-Shalabi, G. Kanaan, and A. Nassar, "Different classification algorithms based on Arabic text classification: feature selection comparative study," *Int. J. Adv. Comput. Sci. Appl*, vol. 6, no. 2, pp. 23–28, 2015.

[27] S. Al-Harbi, A. Almuhareb, A. Al-Thubaity, M. Khorsheed, and A. Al-Rajeh, "Automatic Arabic text classification," *Proceedings of the international conference on the statistical analysis of textual data*, pp. 77–83, 2008.

[28] M. Mafarja and S. Abdullah, "A fuzzy record-to-record travel algorithm for solving rough set attribute reduction," *International Journal of Systems Science*, vol. 46, no. 3, pp. 503–512, 2015.

[29] M. M. Mafarja and S. Mirjalili, "Hybrid whale optimization algorithm with simulated annealing for feature selection," *Neurocomputing*, vol. 260, pp. 302–312, 2017.

[30] M. Spitters, "Comparing feature sets for learning text categorization," *Content-Based Multimedia Information Access*, vol. 2, pp. 1124–1134, 2000.

[31] F. Yang, T. Sun, and C. Zhang, "An efficient hybrid data clustering method based on K-harmonic means and Particle Swarm Optimization," *Expert Systems with Applications*, vol. 36, no. 6, pp. 9847–9852, 2009.

[32] A. Sharaff, N. K. Nagwani, and K. Swami, "Impact of feature selection technique on email classification," *Int. J. Knowl. Eng*, vol. 1, no. 1, pp. 59–63, 2015.

[33] S. Bahassine, A. Madani, M. Al-Sarem, and M. Kissi, "Feature selection using an improved Chi-square for Arabic text classification," *Journal of King Saud University-Computer and Information Sciences*, vol. 32, no. 2, pp. 225–231, 2020.

[34] M. H. Aghdam, N. Ghasem-Aghaee, and M. E. Basiri, "Text feature selection using ant colony optimization," *Expert Systems with Applications*, vol. 36, no. 3, pp. 6843–6853, 2009.

[35] S. L. Marie-Sainte and N. Alalyani, "Firefly algorithm based feature selection for Arabic text classification," *Journal of King Saud University-Computer and Information Sciences*, vol. 32, no. 3, pp. 320–328, 2020.

[36] H. Majidpour and F. S. Gharehchopogh, "An improved flower pollination algorithm with AdaBoost algorithm for feature selection in text documents classification," *Journal of Advances in Computer Research*, vol. 9, no. 1, pp. 29–40, 2018.

[37] P. Shunmugapriya and S. Kanmani, "A hybrid algorithm using ant and bee colony optimization for feature selection and classification (AC-ABC Hybrid)," *Swarm and evolutionary computation*, vol. 36, pp. 27–36, 2017.

[38] E. A. H. Khalil, E. H. F. Houby, and H. K. Mohamed, "Deep learning for emotion analysis in Arabic tweets," *J Big Data*, vol. 8, pp. 136–136, 2021.

[39] I. Targio, M. Hashem, and Tubishat, "Hybrid Feature Selection Based on Principal Component Analysis and Grey Wolf Optimizer Algorithm for Arabic News Article Classification," *IEEE Access*, vol. 10, pp. 121 816–121 830, 2022.

[40] P. Dhal and C. Azad, "A fine-tuning deep learning with multi-objective-based feature selection approach for the classification of text," *Neural Comput & Applic*, 2023.

[41] I. Jamaleddyn, R. E. Ayachi, and M. Biniz, "An improved approach to Arabic news classification based on hyperparameter tuning of machine learning algorithms," *Journal of Engineering Research*, vol. 11, no. 2, pp. 2023–2023.

[42] A. Ayedh, G. Tan, K. Alwesabi, and H. Rajeh, "The effect of preprocessing on arabic document categorization," *Algorithms*, vol. 9, no. 2, pp. 27–27, 2016.

[43] B. Alhadidi and M. Alwedyan, "Hybrid Stop-Word Removal Technique for Arabic Language." *Egypt. Comput. Sci. J.*, vol. 30, no. 1, pp. 35–38, 2008.

[44] A. Jaffar, M. Wedyan, and H. Al-Zoubi, "Arabic text light stemmer," *Int. J. Comput. Acad. Res*, vol. 8, no. 2, pp. 17–23, 2019.

[45] J. Atwan, M. Wedyan, and H. Al-Zoubi, "Arabic text light stemmer," *Int. J. Comput. Acad. Res*, vol. 8, no. 2, pp. 17–23, 2019.

[46] J. Atwan, M. Wedyan, Q. Bsoul, A. Hamadeen, R. Alturki, and M. Ikram, "The effect of using light stemming for Arabic text classification," *International Journal of Advanced Computer Science and Applications*, no. 5, pp. 12–12, 2021.

[47] L. S. Larkey, L. Ballesteros, and M. E. Connell, "Light stemming for Arabic information retrieval," in *Arabic computational morphology: knowledge-based and empirical methods*, and others, Ed. Springer, 2007, pp. 221–243.

[48] M. A. H. Omer and S. Ma, "Stemming algorithm to classify Arabic documents," *Journal of Communication and Computer*, vol. 7, no. 9, pp. 1–5, 2010.

[49] G. Salton, A. Wong, and C. S. Yang, "A vector space model for automatic indexing," *Communications of the ACM*, vol. 18, no. 11, pp. 613–620, 1975.

[50] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," *Information Processing & Management*, vol. 24, no. 5, pp. 513–523, 1988.

[51] W. G. Indriyani and A. Rakhmadi, "Filter-wrapper approach to feature selection using PSO-GA for Arabic document classification with Naive Bayes multinomial," *IOSR Journal of Computer Engineering*, vol. 17, no. 6, pp. 45–51, 2015.

[52] K. C. Lin, K. Y. Zhang, Y. H. Huang, J. C. Hung, and N. Yen, "Feature selection based on an improved cat swarm optimization algorithm for big data classification," *The Journal of Supercomputing*, vol. 72, pp. 3210–3221, 2016.

[53] Y. Saeys, I. Inza, and P. Larrañaga, "A review of feature selection techniques in bioinformatics," *Bioinformatics*, vol. 23, no. 19, pp. 2507–2517, 2007.

[54] R. D. Thompson, "Atmospheric processes and systems," and others, Ed. Psychology Press, 2002.

[55] Z. Bayraktar, M. Komurcu, and D. H. Werner, "Wind Driven Optimization (WDO): A novel nature-inspired optimization algorithm and its application to electromagnetics," *IEEE Antennas and Propagation Society International Symposium*, pp. 1–4, 2010.

[56] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.

[57] F. Y. Vincent, S. W. Lin, W. Lee, and C. J. Ting, "A simulated annealing heuristic for the capacitated location routing problem," *Computers & Industrial Engineering*, vol. 58, no. 2, pp. 288–299, 2010.

[58] R. Jensen and Q. Shen, "Fuzzy-rough attribute reduction with application to web categorization," *Fuzzy Sets and Systems*, vol. 141, no. 3, pp. 469–485, 2004.

[59] N. S. Altman, "An introduction to kernel and nearest-neighbor non-parametric regression," *The American Statistician*, vol. 46, no. 3, pp. 175–185, 1992.

[60] T. M. Mitchell, "Machine Learning," in *Machine Learning*, vol. 1, 2007.

[61] A. Mccallum and K. Nigam, "A comparison of event models for naive bayes text classification," *AAAI-98 Workshop on Learning for Text Categorization*, vol. 752, pp. 41–48, 1998.

[62] P. Y. Pawar and S. H. Gawande, "A comparative study on different types of approaches to text categorization," *International Journal of Machine Learning and Computing*, vol. 2, no. 4, pp. 423–423, 2012.

[63] J. Su and H. Zhang, "A fast decision tree learning algorithm," in *Aaai*, and others, Ed., vol. 6. McGraw-Hill Science, 2006, pp. 500–505.

[64] D. Graff and K. Walker, "Linguistic Data Consortium (LDC)," 2001. [Online]. Available: Https://Catalog.Ldc.Upenn.Edu/LDC2001T55

[65] T. Hastie, R. Tibshirani, and J. H. Friedman, "The Elements of Statistical Learning: Data Mining, Inference, and Prediction," *Biometrics*, vol. 66, no. 4, p. 746, 2010.