future internet

MDPI

Article

A Novel Text Classification Technique Using Improved Particle Swarm Optimization: A Case Study of Arabic Language

Yousif A. Alhaj [1], Abdelghani Dahou [2], Mohammed A. A. Al-qaness [3,4], Laith Abualigah [5,6], Aaqif Afzaal Abbasi [7], Nasser Ahmed Obad Almaweri [1], Mohamed Abd Elaziz [8,9,10] and Robertas Damaševičius [11,]*

[1] Sanaa Community College, Sanaa 5695, Yemen; yalhag@gmail.com (Y.A.A.); dr.nasser@scc.edu.ye (N.A.O.A.)
[2] Mathematics and Computer Science Department, Ahmed Draia University, Adrar 01000, Algeria; dahou.abdghani@univ-adrar.edu.dz
[3] State Key Laboratory for Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, Wuhan 430079, China; alqaness@whu.edu.cn or m.alqaness@eng-su.edu.ye
[4] Faculty of Engineering, Sana'a University, Sana'a 12544, Yemen
[5] Faculty of Information Technology, Middle East University, Amman 11831, Jordan; aligah.2020@gmail.com
[6] Faculty of Computer Sciences and Informatics, Amman Arab University, Amman 11953, Jordan
[7] Department of Software Engineering, Foundation University Islamabad, Islamabad 44000, Pakistan; aaqif.afzaal@fui.edu.pk
[8] Faculty of Computer Science and Engineering, Galala University, Suez 435611, Egypt; abd_el_aziz_m@yahoo.com
[9] Artificial Intelligence Research Center (AIRC), College of Engineering and Information Technology, Ajman University, Ajman P.O. Box 346, United Arab Emirates
[10] Department of Mathematics, Faculty of Science, Zagazig University, Zagazig 44519, Egypt
[11] Department of Applied Informatics, Vytautas Magnus University, 44404 Kaunas, Lithuania
* Correspondence: robertas.damasevicius@vdu.lt

check for updates

Citation: Alhaj, Y.A.; Dahou, A.; Al-qaness, M.A.A.; Abualigah, L.; Abbasi, A.A.; Almaweri, N.A.O.; Elaziz, M.A.; Damaševičius, R. A Novel Text Classification Technique Using Improved Particle Swarm Optimization: A Case Study of Arabic Language. *Future Internet* **2022**, *14*, 194. https://doi.org/10.3390/fi14070194

Academic Editor: Paulo Quaresma

Received: 31 May 2022
Accepted: 22 June 2022
Published: 27 June 2022

Abstract: We propose a novel text classification model, which aims to improve the performance of Arabic text classification using machine learning techniques. One of the effective solutions in Arabic text classification is to find the suitable feature selection method with an optimal number of features alongside the classifier. Although several text classification methods have been proposed for the Arabic language using different techniques, such as feature selection methods, an ensemble of classifiers, and discriminative features, choosing the optimal method becomes an NP-hard problem considering the huge search space. Therefore, we propose a method, called Optimal Configuration Determination for Arabic text Classification (OCATC), which utilized the Particle Swarm Optimization (PSO) algorithm to find the optimal solution (configuration) from this space. The proposed OCATC method extracts and converts the features from the textual documents into a numerical vector using the Term Frequency-Inverse Document Frequency (TF–IDF) approach. Finally, the PSO selects the best architecture from a set of classifiers to feature selection methods with an optimal number of features. Extensive experiments were carried out to evaluate the performance of the OCATC method using six datasets, including five publicly available datasets and our proposed dataset. The results obtained demonstrate the superiority of OCATC over individual classifiers and other state-of-the-art methods.

Keywords: text classification; feature selection; feature extraction; particle swarm optimization

1. Introduction

Recently, the internet has witnessed a massive accumulation of valuable information growing exponentially every day. Most of this information is an unstructured text which creates challenges for humans to manage and process this information and extract proper knowledge [1]. A new research field in text mining called text classification (TC) emerged to overcome this problem. TC is a machine learning challenge that tries to classify new written content into a conceptual group from a predetermined classification collection [1]. It is crucial in a variety of applications, including sentiment analysis [2,3], spam email filtering [4,5], hate speech detection [6], text summarization [7], website classification [8], authorship attribution [9], information retrieval [10], medical diagnostics [11], emotion detection on smart phones [12], online recommendations [13], fake news detection [14,15], crypto-ransomware early detection [16], semantic similarity detection [17], part-of-speech tagging [18], news classification [19], and tweet classification [20].

Several primary stages are needed to build TC systems [21], namely, the preprocessing stage (tokenization, stop word removal, normalization, and stemming), document modeling stage (feature extraction and feature selection), and the final document classification and evaluation stage.

Compared to other languages, such as English, TC in Arabic is resource-poor. However, Arabic is the world's fifth most widely spoken language, with around 4.5% of its population using it as their first language [1]. The Arabic alphabet consists of 28 letters and directions from right to left. As Arabic words do not begin with a capital letter like they do in English, distinguishing correct names, acronyms, and shortcuts can be challenging. There are also diacritics, which are symbols put above or below letters to give different sounds and grammatical formulations that can alter the definition of a sentence [22]. The shape and construction of the same letter vary depending on where it appears in the sentence [23]. The Arabic language requires a variety of preprocessing methods before classification due to several obstacles, including the Arabic language's strong affixation character and the scarcity of freely available Arabic datasets [23], as well as the scarcity of standard Arabic morphological analysis software.

The ATC system uses a robust feature selection (FS) method and classifier (CA) to enhance the performance [24]. The former executes a classification process, whereas the former select useful features to decrease the high dimensionality of the feature space. Additionally, incorporating FS in TC systems will help reduce classification complexity and processing demands [25,26]. Over the past years, researchers have been challenged with finding robust FS methods, relevant features and classifiers to enhance the performance of the TC system. This problem occurs due to many available FS methods and techniques in the literature.

Obviously, executing this trail is a complicated and time-consuming task. Therefore, this necessitates the development of a technique to find an optimal solution, among others.

Recently, optimization techniques such as Particle Swarm Optimization (PSO) [27] have been used to solve selected problems across several domains [28]. These techniques are known to emulate natural evolution in their operations. PSO is a comparative evolutionary algorithm (EA) based on swarm intelligence, which is considered one of the most efficient search methods over various proposed method in the literature. Moreover, it is not very expensive and can converge faster in comparison with other EA [29]. This is the main motivation behind using the PSO method. Additionally, it has been successfully applied in feature selection [30], ensemble learning [31], and clustering [32]. Therefore, this work provides a new technique for ATC that uses a meta-heuristic algorithm to find the best answer from a variety of feature selection methods and classifiers using a set of features. This configuration is determined using the PSO [33].

The proposed method, called OCATC, consists of three phases covering data preparation, experiment, and testing and evaluation. In the first phase, OCATC starts by preparing a given Arabic dataset using several preprocessing tasks, including tokenization, normalization, stop word removal, and stemming, then extracts features from the dataset using the TF–IDF approach. In the second phase, the dataset is divided into train and test sets using 10-fold cross-validation, whereby the training set is used during the learning of PSO to find the optimal configuration. This step is considered the main contribution, where the PSO begins by generating a set of solutions, and each solution represents a configuration of three elements: the feature selection method, number of features, and classifier. Then, the fitness function for each solution is computed to determine the best configuration. After that, it updates the position and velocity of each solution until the prerequisites for stopping are met. Finally, the testing set is used to evaluate the quality of the best configuration to build the classification system using the optimal FS method (first element), the optimal number of features (second element), and the optimal classifier (third element). The recommended approach, to the best of our knowledge, aims to find an effective solution for the ATC system to automatically select the optimal solution from a set of elements such as feature selection methods, features, and classifiers, which has not been applied before.

The purpose of this study is to suggest an alternate way, namely OCATC, based on a PSO algorithm to come up with the best solution for the ATC job. OCATC finds the optimal solution among three elements sets, including feature selection methods, number of features, and classifiers. Then, we evaluated the proposed method using publicly available datasets. Finally, a series of experiments were performed to show the effectiveness of the proposed method.

The key contributions of this research might be summarized as follows:

- Classification is presented as a single objective optimization problem where the search region is divided into three elements set covering feature selection methods, number of features, and classifiers. The optimal combination of elements from these separate classes can be retrieved by PSO algorithm;
- An alternative method (OCATC) is proposed using improved Particle Swarm Optimization (PSO) algorithm to find the optimal combination (solution) for ATC;
- The optimal solution for a set of publicly available datasets is presented.

The remainder of the paper is laid out as follows. The precise research goals are stated in Section 2. Section 3 contains the literature review. The suggested work's technique is presented in Section 4. The Section 5 discusses experimental work and assessment. In Section 5, we come to a conclusion regarding our findings.

2. Literature Review

Several studies have looked at the issue of automatic TC, providing various methodologies and answers. This is primarily true for English, but it also applies to other languages, such as Arabic, it is still in an early stage [1]. The authors of [34] used a variety of categorization techniques to investigate the impact of removing stop words on ATC. They discovered that a support vector machine (SVM) classifier with little sequential optimization had the best error rate and accuracy. In [35], the Naive Bayes (NB) classifier is used to examine the impact of a light stemmer, Khoja stemmer, and root extractor on Arabic document categorization. The authors came to the conclusion that a root extractor combined with a position tagger would yield the most outstanding results.

Chi-square (Chi2), Information Gain (IG), NG-Goh-Low (NGL), and Galavotti–Sebastiani–Simi Coefficient (GSS) coefficients were used to determine the essential features and the influence of feature reduction approaches on Arabic document categorization [22]. They also employed feature weighting approaches based on inverse document frequency (IDF), such as Term frequency (TF–IDF), the location of a word's first occurrence (FAiDF), and the compactness of the word (WCF) (CPiDF). The classification model was established using the SVM classifier. Thus, when the TF–IDF, CPiDF, and FAiDF feature weighting methods were combined, the GSS outperformed the other feature selection strategies. The feature selection approach for ATC by [36] was binary practical swarm optimization with a KNN classifier (BPSO-KNN). The Alj-News dataset was utilized to develop the classification model, and the best results were obtained utilizing SVM and NB classifiers.

On Arabic document classification, Sabbah et al. [37] tested a number of feature selection approaches, including Chi2, IG, Correlation (Corr), and SVM-based Feature Ranking Method (SVM-FRM). The classification model was constructed using an SVM classifier. In their research, they used the BBC [38] and Abuaiadah [39] databases. They arrived at the conclusion that SVM-FRM performs well in a balanced dataset, but not so well in an imbalanced dataset.

A novel feature selection method, namely, improved chi-square (ImpCHI), was presented by [40] to enhance the ATC. Three standard features selection methods, namely, IG, Chi2, and Mutual Information (MI), were compared with ImpCHI. SVM and DT classifiers were used to evaluate the performance using CNN dataset [38]. Experimental results demonstrate that the most beneficial result was obtained using the ImpCHI FS method using the SVM classifier.

In [41], the authors presented the Frequency Ratio Accumulation Method, a revolutionary text categorization technique for the Arabic language (FRAM). The characteristics were extracted using a bag of words (BoW). Chi2, MI, Odds Ratio (OR), and GSS-coefficient were among the feature selection approaches used to exclude unnecessary characteristics (GSS). According to the results, the FRAM outperformed the NB, Multi-variant Bernoulli Naive Bayes (MBNB), and Multinomial Naive Bayes models (MNB) classifiers. The macro-f-measure value was 95.1 percent for the unigram word-level representation approach.

The Polynomial Networks (PNs) classifier used by [42] on Arabic text classification using the Alj-News dataset. They compared the performance of the PNs classifier with other classifiers, such as SVM, NB, and DT classifiers. Their results showed that the performance of the PNs classifier was not the best for all the categories in the dataset, but was very competitive. The authors in [43] claimed to be the first to utilize Logistic Regression (LR) in Arabic Text categorization with the Alj-News dataset. The results of the experiments showed that Logistic Regression is beneficial for ATC.

The effects of eight supervising learning algorithms on Arabic document classification Studied by [44]. Several feature representation approaches were used to extract features from the Abuaiadah dataset [39]. The authors concluded that superior results were obtained when gathering an LSVM classifier with the IDFT approach.

Abdelaal et al. [45] proposed an automatic classification model for Hadith. The proposed model was used to organize Arabic text Hadith into related categories: Sahih, Hasan, Da'if, and Maudu. Several classifiers, such as LSVC, SGD, and LR, were investigated to build a classification model. IG, Chi2, and Gain Ratio (GR) are feature selection methods to remove irrelevant features. The outcomes showed that LSVC outperforms other classifiers. Moreover, in [46], the authors evaluated the Hadith dataset using DT, RF, and NB classifiers, and they isolated redundant features using IG and Chi2. Binary boolean algebra and TF–IDF were used to extract features. Experimental results demonstrated that the best classifier investigated was DT.

Elnagar et al. [1] performed an extensive analysis to evaluate the effectiveness of Deep Neural Networks (DNN) models and a word2vec embedding model on new constructed large corpora for Arabic document classification. The corpus was SANAD (Single-label-Arabic- News- Articles Dataset), collected by Einea et al. [47]. The evaluation experiments showed the effectiveness of the proposed models on single-label categorization and multi-label categorization.

Alhaj et al. [48] studied the effects of stemming strategies on ATC. Several classifiers, including NB, KNN, and SVM, were used to build the classification model. Chi2 was used to extract essential features in different sizes. An available public dataset, namely, CNN, was used to evaluate the classification model. The outcomes demonstrated that the SVM classifier gathered with ARLStem stemmer outperforms other classifiers when increasing the features. Moreover, in [49], the authors studied the effects of stop word removal in several classifiers and feature extractions using the CNN public dataset. Chi2 was applied as a feature selection method. The TF–IDF and BoW methods were used to extract features. They concluded that the best results were achieved when removing stop words gathering with TF–IDF and the VM classifier.

Researchers reported several classifiers and feature selection methods to improve the ATC system from the literature. Therefore, our approach is used to automatically find an effective solution to improve the ATC system among this literature, such as choosing the optimal feature selection method with an optimal number of features alongside the classifier using supervised machine learning. Furthermore, to clear up the differences between existed works with our approach, we conclude these differences in Table 1. This table includes preprocessing, normalization, stop word removal, stemming technique, feature selection, and classification algorithms, abbreviated as PR, NR, SR, ST, FS, and CA, respectively.

Table 1. Comparison with previous studies.

Study	PR	FS	Dataset	CA
[34]	SR	Not reported	In-house Collected	NB, SVM, DT
[35]	SR, NR, ST	Term Frequency	BBC	NB
[22]	SR, NR, ST	Chi2, IG, NGL, GSS	In-house Collected	SVM
[36]	SR	BPSO-KNN	Alj-News	SVM, NB, DT
[37]	SR, ST	Chi2, IG, Corr, SVM-FRM	Abuaiadah, BBC	SVM
[40]	SR, NR, ST	IG, Chi2, MI,ImpCHI	CNN	SVM
[41]	SR, NR, ST	Chi2, MI, OR , GSS	In-house Collected	FRAM, NB, MBNB, MNB
[42]	SR, ST	Chi2	Alj-News	PNs
[43]	SR, ST	Chi2	Alj-News	LR
[44]	SR, ST	Not reported	Abuaiadah	KNN, NB,LSVM, DT, BN, RF, RT, RC
[45]	NR, SR, ST	IG,Chi2,GR	In-house Collected	LSVM, SGD, LR
[46]	NR, SR, ST	Chi2,IG	In-house Collected	DT, BN, RF
[1]	NR	word2vec	SANAD (Arabiya)	DNN
[48,49]	NR, SR, ST	Chi2	CNN	KNN, NB, SVM
Proposed method	**NR, SR, ST**	**PCA, SVD, IG** **Chi2, RFF**	**CNN, BBC, Alj-News** **SANAD (Arabiya)** **Abuaiadah ,our dataset**	**LR, RF, KNN, DT** **SVC, LSVC, SGD** **NN**

3. Methodology

3.1. Arabic Text Classification

Preprocessing, document structuring, and document categorization are the three key steps of the ATC system. Document conversion, tokenization, normalization, stop word removal, and stemming methods are all part of the preprocessing step. Feature extraction and feature selection are part of the document modeling step. Finally, the document classification phase. All the phases are explained in detail as follows.

3.1.1. Preprocessing

Preprocessing is considered the vital part of TC [23], which starts by dividing the text into sequences of tokens. Secondly, normalization transforms the characters into a standard format, remove (all non-Arabic characters, diacritics, numbers, and punctuation). Then, 1057 stop words lists were prepared to be removed from all the documents [49]. Finally, stemming was applied by reducing the word into their root/stem [50] using a novel stemmer [51].

3.1.2. Document Modeling

This procedure is known as feature extraction (representation) and it comprises the following stages.

The method of extracting features from text is known as feature extraction and represents it as a numerical matrix value to be suitable for machine learning algorithms. The rows in this matrix relate to documents, and the columns to terms (words). There are several methods for deciding each matrix element's value. One of the known schemes is TF–IDF [52]. The relevance of a word within a document is measured by TF, whereas the global significance of a term within a dataset is measured by DF [53]. Specficially, we consider a group of documents D containing N documents, such that D = $d_0, \ldots, d_n - 1$ every document that contains a group of terms t is going to be depicted as a vector in vector space model (VSM) as follows.

$$d_{1j} = (t_{i1}, \ldots, t_{iy}), j = 1, \ldots, y \tag{1}$$

where y is the total number of certain words within the given document (d_i).

The $TF - -IDF$ strategy employs two main techniques, TF and DF, to determine the importance (weight) of a term in a document by using Equations (2) and (3):

$$IDF(d,t) = log\frac{N}{DF(d,t)} \tag{2}$$

$$TF\_IDF = TF(t,d) * IDF(d,t) \tag{3}$$

where $TF(t,d)$ represents the number of times word t appears in document d, $DF(d,t)$ represents the number of documents that include term t, and N represents the total number of documents in the training set.

3.2. Feature Selection (FS)

Feature selection can be called attribute selection or variable selection, which is defined as the procedure of selecting a subset of relevant features to construct the classification model [54]. Moreover, it is known as reducing dimensionality by creating new combinations of attributes. For various reasons, including the performance of classification algorithms that are adversely affected when dealing with the large number of dimensions features, VSM has a large number of terms (dimensions) that are irrelevant to the classification assignment and can be removed without affecting classification accuracy. Furthermore, the over-fitting problem may occur when the classification model is trained on all features. Finally, some features are standard and happen in all or most classes. Thus, feature selection methods are important to minimize the feature dimensions to improve the machine learning models. In reality, utilizing feature selection or reduction is the key to improving the classifiers' performance. Several feature selection and reduction methods are used in TC; it is tough to say which approach is typically better for others because the method's success is dependent on several factors. As a result of this research, a set of feature selection approaches such as chi-square testing were used (Chi2) [55], Information Gain (IG) [56], Relief (RFF) [57], Singular Value Decomposition (SVD) [58], Principal Component Analysis (PCA) [59] are addressed and described as follows:

3.2.1. Chi-Square (Chi2)

Chi-square analysis χ^2 is a unique data hypothesis testing technique from statistics that assess whether variables are correlated or associated by measuring the correlation among them. Chi-square statistics show how important each element is to the category. Equations (4) and (5) can be used to determine the value of χ^2 for each characteristic t in a category c.

$$\chi^2(t_k, c_i) = \frac{|Tr| \cdot [p(\bar{t_k}, \bar{c_i}) * p(t_k, c_i) - p(\bar{t_k}, c_i) * p(t_k, \bar{c_i})]^2}{p(t_k) * p(\bar{t_k}) * p(c_i) * p(\bar{c_i})} \tag{4}$$

Moreover, is estimated using

$$\chi^2(t, \mu) = \frac{N * (\alpha\omega - \mu\beta)^2}{(\alpha + \mu) * (\beta + \omega) * (\mu + \omega) * (\alpha + \beta)} \tag{5}$$

where:

α = The recurrence of t and μ occurrences;
β = The recurrence of t occurrences without μ;
μ = The recurrence of μ without t;
ω = The recurrence of non-occurrence of both μ and t and N is the quantity of document.

3.2.2. Information Gain (IG)

In machine learning, information gain (IG) is commonly used as a feature quality standard. By identifying the existence or absence of a phrase in a document, IG evaluates the quantity of information obtained for category prediction [22]. The goal of IG is to establish which attributes reveal the most information about the category. The following is the definition of the IG feature t:

$$IG(t, c_i) = \sum_{i=1}^{i=m} p(t, c) \cdot log \frac{p(t, c_i)}{p(c_i) \cdot p(t)} + \sum_{i=1}^{i=m} p(\bar{t}, c_i) \cdot log \frac{p(\bar{t}, c_i)}{p(\bar{t}) \cdot p(c_i)} \tag{6}$$

Moreover, it is estimated using:

$$IG(t) = \sum_{i=1}^{i=m} \alpha \cdot log \frac{\alpha}{(\alpha + \beta)(\alpha + \mu)} + \sum_{i=1}^{i=m} \beta \cdot log \frac{\beta}{(\alpha + \beta)(\beta + \omega)} \tag{7}$$

3.2.3. Relief (RFF)

Relief (RFF) is a frequently used feature ranking method that determines the significance of features based on how effectively they evaluate and distinguish the sampled item from its nearest hit (an instance of the same category) and nearest miss (an instance of a different category) (different categories). The Relief feature selection method picks feature instances at random from the training data. For every sampled occurrence, the nearest hit and miss are identified. Suppose a characteristic differs across various classes and has a comparative assessment for examples of the same classes. In that case, it is given a valuable weight. It tries to obtain the most accurate assessment from the probability below to use as the weight for each word feature f [57].

3.2.4. Singular Value Decomposition (SVD)

Singular Value Decomposition (SVD) is a well-known approach in the information retrieval study area developed in [58] to lower the dimensionality in applications such as text classification, where documents are presented with vectors.

From a linear algebra-based theorem, the SVD technique can decompose a matrix A of size $m \times n$ into the product of three matrices: an $m \times m$ orthogonal matrix u, an $m \times n$ diagonal matrix S, and the transpose of an orthogonal matrix V of size $n \times n$. SVD formula is usually presented as the following:

$$A_{n \times m} = U_{m \times m} \ \times \ S_{m \times n} \ \times \ V^T_{n \times n} \tag{8}$$

Dimensionality reduction is applied to the data to remove the noise done by deleting rows from the bottom of matrices U and S. The left columns from matrices S and V^T are also removed.

3.2.5. Principal Component Analysis (PCA)

PCA is a widely used technique to perform dimensionality reduction for reducing the number of features considerably in massive data sets while preserving much of the information from the original data set. PCA is often applied as the first step before moving to another technique, such as multiple regression, cluster analysis, and/or discriminant analysis. The data are initially centered in PCA by subtracting the mean (in practice, the mean is estimated as the average value of the vector in a sample). The vector is then turned into a new array, maybe of a smaller dimension, where the development work members are uncorrelated, using a linear transformation. The differences of the data projections on the new coordinate axes are increased, with the first axis corresponding to the greatest variance. The second axis represents the greatest variation opposite to the first axis.

In mathematical terms, an orthonormal basis vector is generated using PCA, which maximizes the scatter of objects. Considering $X = [x_1, x_2, \ldots, x_N]$ represents the set of vectors, where N is the number of objects. The next step in PCA is to normalize the X to become a unit norm. Then, the average is subtracted to produce a new set of vectors $Y = [y_1, y_2, \ldots, y_N]$. The next step is to compute the covariance matrix of Y as:

$$\sum_Y = \frac{1}{N} Y Y^T \tag{9}$$

After that, the eigenvalue and the eigenvector are computed as in the following equation:

$$\sum_Y \Phi = \Phi \Lambda \tag{10}$$

where Φ and Λ represent the eigenvector and eigenvalue, respectively. Then, the eigenvalue is sorted and select the most relevant eigenvectors Φ_r are selected which correspond to the best eigenvalues and reduce the data as in the following:

$$Tar = \Phi_r^T Y \tag{11}$$

where T represents the reduced dataset.

3.3. Document Classification

Document classification is known as the process which builds and evaluates the model of classification. In this study, a set of classifiers were utilized, namely, Logistic Regression (LR) [60], Random Forest(RF) [61], k-nearest neighbors algorithm (KNN) [61], Decision Tree (DT) [62], Neural Networks (NN) [63], Support Vector Machine (SVM) [61], Linear Support Vector Machine (LSVM) [64], and Stochastic Gradient Descent (SGD) [65].

Particle Swarm Optimization (PSO)

Particle Swarm Optimization Algorithm (PSO) is generally known as a population-based intelligence algorithm which was suggested by [33]. PSO is grounded on the interaction and the elementary behavior among agents (swarm individuals) to find the optimal solution and solve complex problems. PSO begins with a swarm of random particles. Through the optimizing procedure for a particular challenge with N variables, the i-th particle has a specific velocity vector, and a position vector presented as $V_i = [V_{i1}, V_{i2}, \ldots, V_{iD}]$ and $X_i = [X_{i1}, X_{i2}, \ldots, X_{iN}]$, respectively. The vector X_i is considered as a candidate solution to solve the given problem while the vector V_i is considered as the particle's search direction and step size. During the process of optimization every particle decides its trajectory according to its personal historical best position $Pb_i = [Pb_{i1}, Pb_{i2}, \ldots, Pb_{iN}]$ and the global best-so-far position $Gb = [gb_1, gb_2, \ldots, gb_N]$. In the canonical PSO, the update rules of V_i and X_i are defined as (12) and (13), respectively.

$$V_{ij}^{t+1} = Y \times V_{ij}^{t} + c_1 \times rnd_1 \times (pb_{ij}^{t} - X_{ij}^{t}) + c_2 \times rnd_2 \times (gb_{j}^{t} - X_{ij}^{t}) \tag{12}$$

$$X_{ij}^{t+1} = X_{ij}^{t} + V_{ij}^{t+1} \tag{13}$$

where Y is an inertia weight presenting how the previous velocity is preserved; c_1 and c_2 are known as two acceleration coefficients determining relative learning weights for Pb_i and Gb, referred to as "self-cognitive" and "social learning", respectively; rnd_1 and rnd_2 are two random numbers uniformly distributed over $[0, 1]$.

Algorithm 1 presents the pseudo-code of PSO.

Algorithm 1 Pseudo-code of the PSO algorithm.

```
Begin
Initialize parameters: $Y = 0.9, c_1 = c_2 = 2.0$;
Generate initial population's positions $X_i$ and velocity $V_i$ where $i = 1, 2, \ldots, N$;
Evaluate all $X_i$ by computing the fitness function (fit);
Determine $Pb_i$ and $Gb$ according to the fitness function values;
While (not meet the stop conditions)
  For $i = 1 : N$
    Update $V_i$ and $X_i$ according to Equations (12) and (13), respectively ;
    If fit ($X_i$) $<$ fit ($Pb_i$)
      $Pb_i$ = $X_i$ ; fit ($pb_i$) = fit ($X_i$);
       If fit ($X_i$) $<$ fit($Gb$)
         $Gb$ = $X_i$ ; fit ($Gb$) = fit($X_i$);
       End If
    End If
  End For
$Y = 0.9 - 0.5$
End While
Return $Gb$
End
```

3.4. Proposed Method

Figure 1 depicts the overall conceptual model of OCATC, divided into three phases. Preprocessing procedures are used to prepare the dataset in the initial phase (as discussed in Section 1). The features from the collection are then extracted using TF–IDF. Using 10-fold cross-validation, the extracted features from the dataset are partitioned into different sets in the second stage. After that, the training set is used by PSO during the search for the best configuration. At this phase, PSO generates a set of random solutions where each one of them represents one configuration. The quality of each configuration is evaluated by computing its objective, which is defined based on the accuracy of the classification and the number of selected features. Then, the best personal solution and best global solution to update the solutions in the next iterations in the next step are determined. The process of updating the quality of the solution is repeated until the stopping conditions are met. In the third phase, the testing set assesses the best solution that represents the best configuration. The OCATC phases are discussed in the following.

Phase 1: Data preparation.

OCATC receives the raw dataset as input to the preprocessing module that applies tokenization, normalization, stop word removal, and stemming technique to prepare the raw dataset. Then, the dataset is represented into the numerical vectors using the TF–IDF approach.

Phase 2: Optimal Select configuration using PSO.

PSO is used to determine the best configuration from a set of feature selection methods, classifiers, and the number of selected features where this method contains the following steps.

(a) Initialization step

The PSO starts by constructing a set of N solutions/configurations X using the following Equation:

$$X_{ij} = floor(LB_j + rand \times (UB_j - LB_j)), j = 1, 2 \ldots, J \quad (14)$$

where LB_j and UB_j represent the lower and upper boundaries of the jth dimension of the current solution X_i. In this study, we set $J = 3$, since only three elements (dimension) are selected to construct the configuration, the first dimension is a classifier, the second dimension is the feature selection method. In contrast, the last dimension is the number of selected features. Therefore, the value of the lower boundary at the first dimension is one $LB_1 = 1$ (also for the second dimension $LB_2 = 1$), while the third dimension is set to zeros ($LB_3 = 0$). The $UB_1 = N_C, UB_2 = N_{FS}$ and $UB_3 = D$, where N_C, N_{FS}, and D are the number of classifiers, feature selection methods, and dimensions of the given dataset, respectively. For further clarification, the representation of each solution is listed in Table 2 where we consider $N_C = 4$, $N_{FS} = 6$ and the dimension for the given dataset is $D = 1500$.

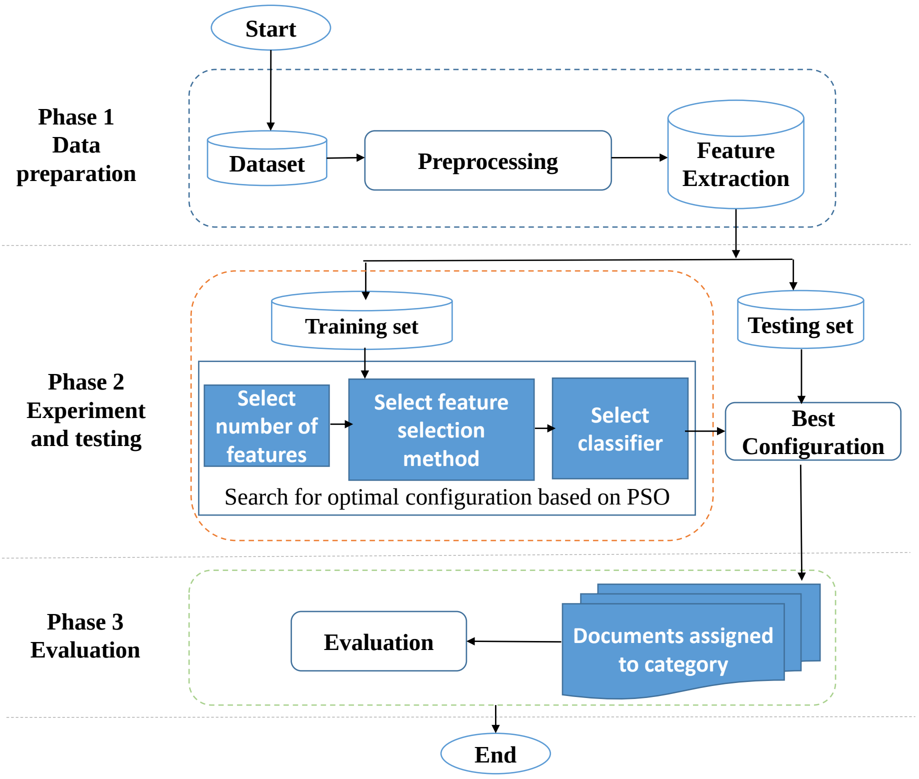


Figure 1. General Framework of OCATC.

Table 2. Representation of the solution.

	X_{i1}	X_{i2}	X_{i3}
Configuration1 X_1	2	4	1000
Configuration2 X_2	3	2	500
Configuration3 X_3	1	5	20

From Table 2, it can see that the first configuration X_1 uses the second classifier and the fourth feature selection method, and 1000 features. The second configuration X_2 uses the third classifier, second feature selection method, and 500 features. In the third configuration, X_3 uses the first classifier, the fifth feature selection method, and selects 20 features.

(b) Updating step

In this step, the fitness function of each solution is computed using the following Equation:

$$Fit = \alpha \times Y_{X_i 1} + (1 - \alpha) \times \frac{X_{i3}}{D} \tag{15}$$

where $Y_{X_{i1}}$ represents the accuracy of the used classifier in X_{i1}, α is a random number that balances between the accuracy and the number of selected features.

(c) Best Solution Approach step

This step finds the best personal solution Pb_i and the global best solution Gb (best configuration). Then, the set of solutions X is updated using the Equations (12) and (13) and sets the value of each dimension inside its boundary. After that, the fitness function for each solution is computed again, and the process of updating X is repeated until the terminal conditions are reached.

Phase 3: Evaluation of the best configuration

This stage starts by using the testing set to evaluate the quality of the selected best configuration. Here, the testing set features are reduced to the selected number of features represented by the third value of Gb using the feature selection method determined by the second value in Gb. The first value of Gb represents the selected classifier applied to predict the labels of the testing set and compute the performance using different measures.

4. Experimental Results

The section presents experiments conducted using OCATC for Arabic text classification.

4.1. Dataset Description

Five datasets have been used to evaluate the performance of OCATC. These datasets, including DatasetA, which was created by Abuaiadah et al. [39], contain 2700 documents distributed over nine categories. In our experiments, we used the same distribution for DatasetA as in [44]. DatasetB, collected by Saad et al. [38] and known as CNN dataset, contains 5070 documents and six main categories. DatasetC, also known as the BBC dataset and created by Saad et al. [38], contains 4763 documents and seven main categories. DatasetD (Alj-News dataset), collected by Al-Tahrawi et al. [42], contains 1500 documents distributed across five categories. DatasetE (Alarabiya dataset) was collected by Einea et al. [47] and we selected the same distribution of [1], in which every category contains 3700 documents distributed into five categories. In addition, we manually collected 1000 documents from different newspaper websites, and classified them into six categories to construct DatasetF. It is worth noting that DatasetA, DatasetD, and DatasetF are balanced datasets where DatasetA, DatasetD contain 300 documents, and DatasetF contains 3700 documents. Other datasets' document distributions are shown in Figure 2. Furthermore, we set the maximum number of features to be extracted from each dataset using TF–IDF to 1000 to avoid the existence of too many rare words.

4.2. Evaluation Criteria

Ten-fold cross-validation is used in our studies, which splits the dataset into ten subgroups (folds). Each fold includes a tenth of the documents in the collection. The remaining folds are used as training sets, while one is utilized as a test set. The findings are reported using micro-f1 (also known as F1) and macro-f1 measurements.

Micro-averaging considers all categorization judgments in the dataset without discriminating between classes. If a dataset's categories are skewed, large categories will outnumber tiny ones. The following formula is used to calculate this metric:

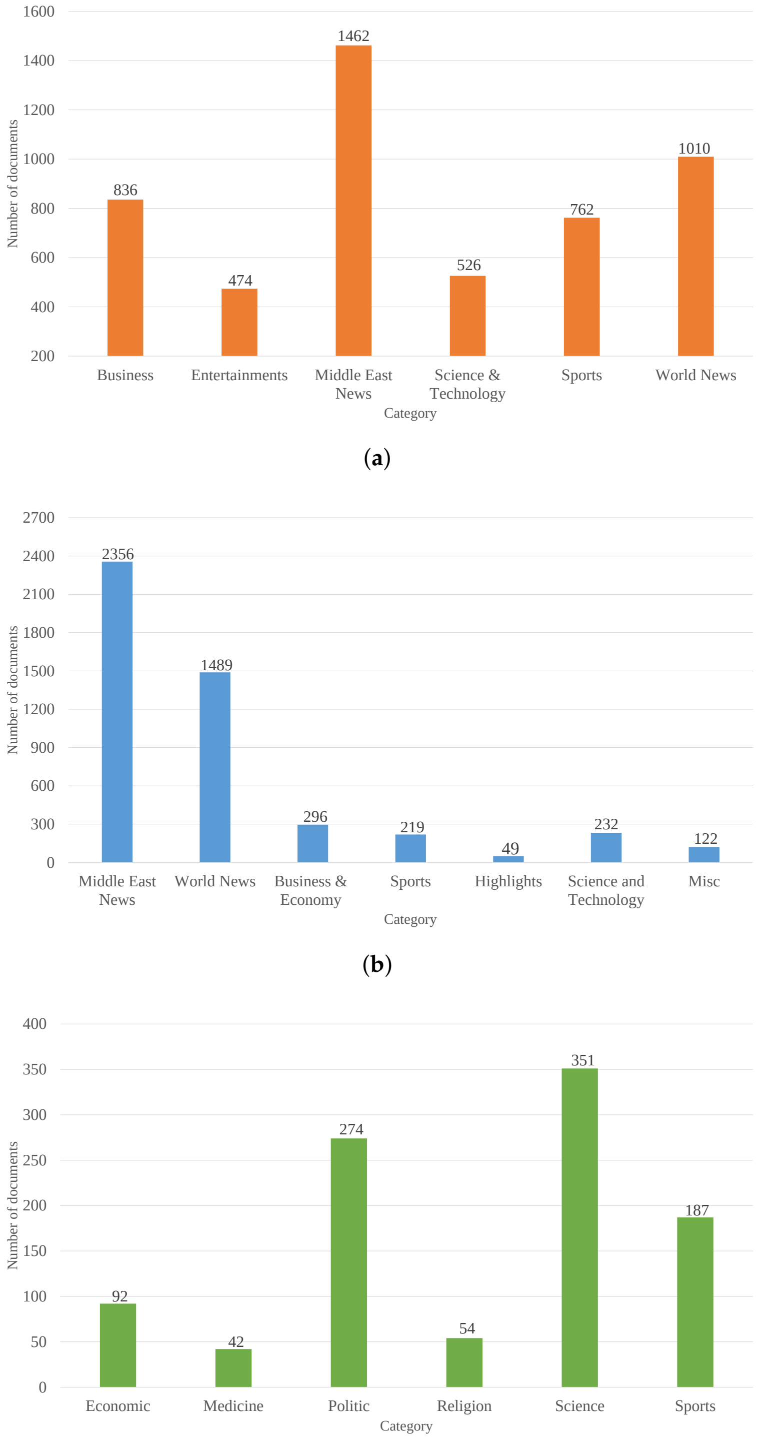
$$F1 = \frac{2 \times P \times R}{P + R} \tag{16}$$

where a pair of (r_k,p_k) represents the recall and precision of categories k, respectively.

Macro-averaging is calculated for each category among the dataset, and so the average overall categories are found. In this manner, equal weight is assigned to each category without considering the categories distributions. The equation of macro-f1 can be written as

$$F1 = \frac{\sum_{j=1}^{c} F_j}{C},\ F_j = \frac{2 \times P \times P_j \times r_j}{P_j + r_j} \tag{17}$$

where a pair of (r_k,p_k) represents recall and precision of class k, respectively.



(**a**)

(**b**)

(**c**)

Figure 2. Document distribution for: (**a**) DatasetB; (**b**) DatasetC; (**c**) DatasetE.

4.3. Parameters Setup of PSO and Classification Algorithms

PSO uses various parameters during its configuration. During the experiments, the number of particles is set to 50, and the number of iterations is set to 100. "c_1" and "c_2", which are PSO acceleration coefficient values, are set to 2. Moreover, classifiers' training parameters were configured according to the default parameters, and as listed in Table 3.

Table 3. Classification algorithms parameters.

Classifier	Parameters
LR	random_state = False.
RF	n_estimators = 50, random_state = False.
KNN	n_neighbors = 6.
DT	random_state = False.
NN	solver = ’lbfgs’, alpha = 1e−5 , hidden_layer_sizes = (15), random_state = False.
SVC	random_state = False, kernel = ‘linear’, C = 1, gamma = 1, degree = 5.
LSVC	random_state = False, max_iter = 5.

4.4. Experimental Series 1: Comparison with Single Classifiers

In this subsection, the performance of OCATC on different datasets is compared to classifiers including LR, RF, KNN, DT, NN, SVC, LSVM, and SGD. The results are listed in Tables 4 and 5. For instance, Table 4 presents the optimal configurations for each dataset obtained by OCATC, where CA stands for the selected classifier, FS method represents the feature selection method, and FN represents the number of features. Table 5 reports the results of OCATC and comparison of different evaluated single classifiers.

Table 4. Results of the optimal configurations obtained by OCATC.

Dataset	CA	FS Method	FN	Micro-f1	Macro-f1
DatasetA	LSVM	RFF	876	0.975	0.975
DatasetB	LSVM	PCA	436	0.936	0.929
DatasetC	RF	IG	36	0.972	0.925
DatasetD	SVC	SVD	289	0.967	0.967
DatasetE	SVC	Chi2	942	0.975	0.975
DatasetF	LSVM	PCA	189	0.962	0.909

As shown in Table 4, the LSVM classifier shows the best performance across three datasets, namely, DatasetA, DatasetB, and DatasetF. On the other hand, PCA seems to work well with LSVM in selecting the optimal features and boosting the performance of LSVM on DatasetB and DatasetF. Considering the selected number of features (FN) per dataset, the IG algorithm only selected 36 relevant features for DatasetC from a total number of 1000 features. In contrast, RFF, PCA, SVD, and Chi2 selected a relatively larger number of features for the rest of the datasets, which seems to provide the highest performance alongside SVM-based classifiers.

Table 5. Comparison of OCATC (optimal configurations) and single classifiers.

Dataset	Measure	OCATC	LR	RF	KNN	DT	NN	SVC	LSVM	SGD
DatasetA	micro-f1	**0.975**	0.971	0.96	0.948	0.847	0.971	0.973	0.973	0.968
	macro-f1	**0.975**	0.971	0.96	0.948	0.847	0.971	0.973	0.973	0.968
DatasetB	micro-f1	**0.936**	0.927	0.893	0.902	0.769	0.912	0.934	0.933	0.932
	macro-f1	**0.929**	0.918	0.872	0.89	0.738	0.903	0.927	0.925	0.922
DatasetC	micro-f1	**0.972**	0.865	0.95	0.841	0.94	0.878	0.887	0.896	0.876
	macro-f1	**0.925**	0.714	0.869	0.763	0.891	0.834	0.848	0.844	0.821
DatasetD	micro-f1	**0.967**	0.963	0.945	0.921	0.827	0.956	0.957	0.959	0.935
	macro-f1	**0.967**	0.963	0.945	0.921	0.826	0.956	0.958	0.959	0.935
DatasetE	micro-f1	**0.974**	0.973	0.961	0.948	0.901	0.963	0.974	0.973	0.973
	macro-f1	**0.974**	0.973	0.961	0.948	0.901	0.963	0.974	0.973	0.973
DatasetF	micro-f1	**0.962**	0.928	0.934	0.931	0.835	0.935	0.955	0.956	0.948
	macro-f1	**0.909**	0.791	0.817	0.87	0.764	0.872	0.887	0.897	0.889

From the balanced datasets perspective, as shown in Table 5, DT did not perform well on balanced datasets such as DatasetA, DatasetD, and DatasetF, compared with other classifiers and OCATC. In single classifier training, it should be noted that the total number of 1000 features were used. LR and DT show the worst performance in terms of macro-f1 on DatasetC and DatasetF, respectively. The same low performance in terms of macro-f1 can be noticed for DT when classifying DatasetB. Thus, it can be concluded that LR and DT suffer from low performance when dealing with imbalanced datasets in this case. Overall, OCATC shows the best results in micro-f1 and macro-f1 compared with all single classifiers over all datasets.

Figure 3 shows the average of precision, recall, and f1-score overall ten folds for each dataset obtained by OCATC. Concerning DatasetA and DatasetD, which are balanced datasets, the Politics category in both datasets had the lowest classification results. In the case of DatasetA, politics documents have been misclassified into Law and Economic categories. Whereas, in DatasetD, Politics documents have been misclassified into Economy and Art categories. Moreover, in the balanced dataset, DatasetE, the lowest classification results were in the Finance category. Entertainments, Highlights, and Medicine categories possess the lowest classification results in terms of recall and f1-score in DatasetB, DatasetC, and DatasetF, receptively. For the Entertainments category in DatasetB, their documents have been misclassified over the rest of the categories. In DatasetC, documents in the Highlights category have been misclassified into Middle East News and World News categories with a percentage of almost 24% and 16%, respectively. In DatasetF, almost 52% of Medicine documents have been misclassified into the Science category.

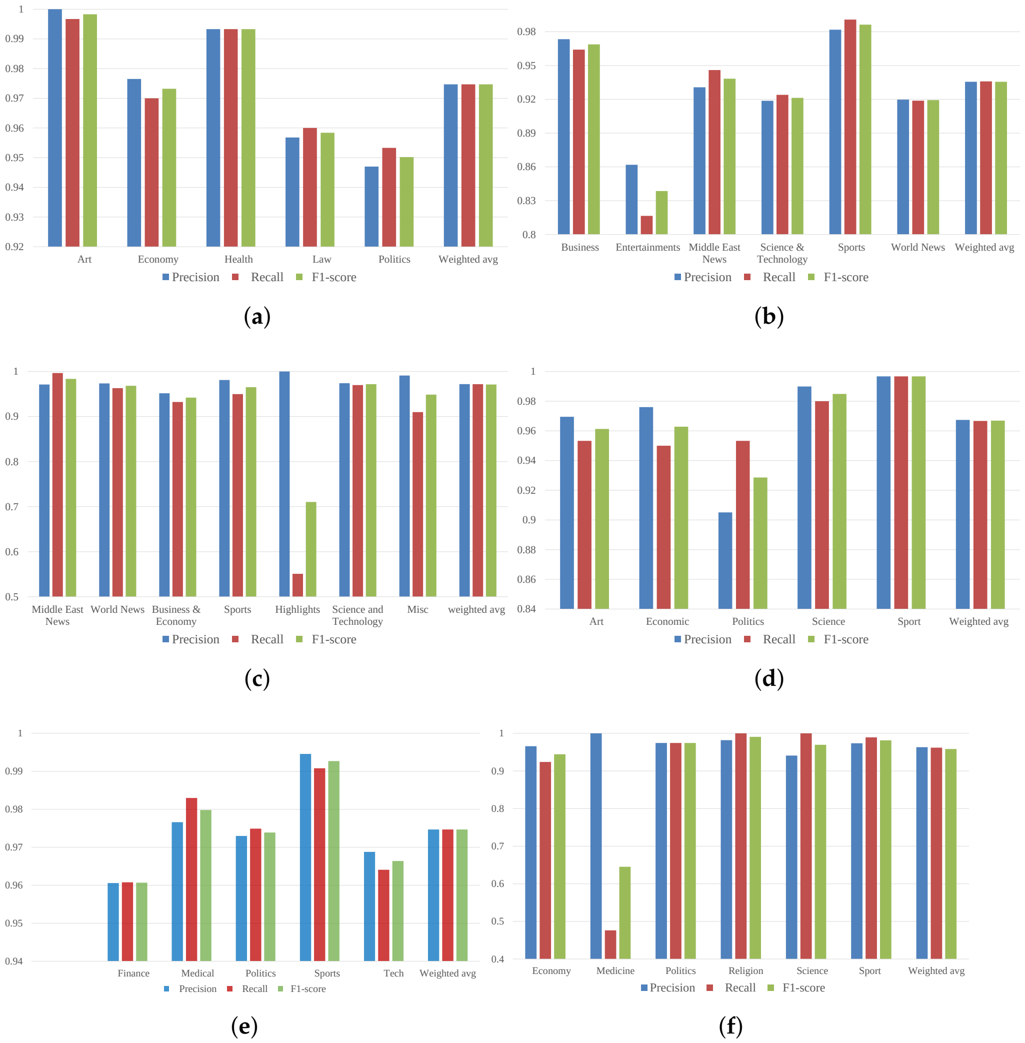
4.5. Experimental Series 2: Comparison with Other Models

To further show the effectiveness of OCATC, we provide a comparison between OCATC and state-of-the-art methods, as shown in Table 6.

Table 6. OCATC results compared to state-of-the-art methods.

Dataset	Study	Previous Study				OCATC			
		Classifier	FS Method	FS	Accuracy	Classifier	FS Method	FS	Accuracy
DatasetA	[37]	SVM	SVM-FRM	4000	0.974	LSVM	RFF	876	0.975
DatasetB	[48]	NB, SVM, KNN	Chi2	1000	0.933	LSVM	PCA	436	0.936
	[49]	NB, SVM, KNN	Chi2	1000	0.909				
	[40]	SVM	ImpCHI	900	0.905				
DatasetC	[37]	SVM	Chi2	3000	0.875	RF	IG	36	0.972
DatasetD	[43]	LR	Chi2	1%	0.865	SVC	SVD	289	0.967
	[36]	SVM	BPSO-KNN	2967	0.931				
	[42]	PNs	Chi2	1%	0.90				
DatasetE	[1]	DNN	Word embedding	128	0.974	SVC	Chi2	942	0.975

From Table 6, we observed that OCATC gives highly competitive results compared with state-of-the-art methods over all datasets. OCATC shows higher performance for DatasetC when compared to the previous study in [37] on this dataset. Thus, OCATC has a better advantage in selecting the suitable classifier, FS method, and the number of features for all datasets, especially for DatasetC. Moreover, the selected number of features by OCATC for all datasets is lower than those in other methods, which helps to achieve better results and saves computation resources. Selecting the correct number of features by discarding redundant and irrelevant features improves the classifier's training and performance. Furthermore, SVM-based classifiers were selected by most of the previously conducted studies and by OCATC. This proves the ability of OCATC to automatically select the most suitable classifier for each dataset without human intervention. Overall, OCATC yields the best results compared to single classifiers and state-of-the-art approaches. During the analysis, we showed the importance of adopting PSO as an optimization algorithm to solve the problem of TC and select the optimal configuration that maximizes the classification performance. Overall, the proposed method obtained better results in comparison with other similar methods, which proved the highest ability of the proposed method in resolving complicated text problems.



(**a**) (**b**)

(**c**) (**d**)

(**e**) (**f**)

Figure 3. Precision, recall, and f1-score for: (**a**) DatasetA; (**b**) DatasetB; (**c**) DatasetC; (**d**) DatasetD; (**e**) DatasetE; and (**f**) DatasetF.

5. Conclusions

The internet has recently seen an extensive collection of useful knowledge rising exponentially daily. Most of these data are unstructured language, making it difficult for people to manage and analyze it and extract valuable knowledge. A new text mining field, called text classification (TC), was introduced to solve a machine learning problem in which new textual material is classified into a theoretical group using a specified classification dataset.

This paper introduces an alternative method for Arabic text classification based on a meta-heuristic algorithm. The proposed method, called Optimal Configuration Determination for Arabic Text Classification (OCATC), uses the Particle Swarm Optimization (PSO) algorithm to find the optimal solution from three elements, namely, feature selection methods, machine learning classifiers, and a pool of features. The PCA, SVD, IG, Chi2, and RFF were used as feature selection methods. Additionally, we used LR, RF, KNN, DT, NN, SVC, LSVC, and SGD as machine learning classifiers. Our proposed method has investigated the optimal configuration (solution) for several Arabic text classification datasets. Compared with single classifiers and other state-of-the-art approaches, experimental findings show that the suggested method is more successful.

In future work, other stochastic optimization strategies might be investigated, as well as establishing how the suggested technique performs for text categorization in different domains and languages. Additionally, the proposed method can be investigated to solve various problems in the domain of computer science and language engineering.

Author Contributions: Conceptualization, A.D. and M.A.A.A.-q.; Data curation, Y.A.A. and M.A.E.; Formal analysis, Y.A.A., A.D., L.A. and R.D.; Funding acquisition, R.D.; Investigation, M.A.A.A.-q., L.A., A.A.A. and N.A.O.A.; Methodology, Y.A.A.; Software, Y.A.A.; Supervision, M.A.A.A.-q.; Validation, Y.A.A., A.D., M.A.A.A.-q., A.A.A., N.A.O.A., M.A.E. and R.D.; Visualization, Y.A.A.; Writing—original draft, Y.A.A.; Writing—review and editing, M.A.A.A.-q., L.A. and R.D. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by LIESMARS Special Research Funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data are publicly available as described in the main text.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Elnagar, A.; Al-Debsi, R.; Einea, O. Arabic text classification using deep learning models. *Inf. Process. Manag.* **2020**, *57*, 102121. [CrossRef]
2. Al-Ayyoub, M.; Khamaiseh, A.A.; Jararweh, Y.; Al-Kabi, M.N. A comprehensive survey of arabic sentiment analysis. *Inf. Process. Manag.* **2019**, *56*, 320–342. [CrossRef]
3. Al-Smadi, M.; Al-Ayyoub, M.; Jararweh, Y.; Qawasmeh, O. Enhancing Aspect-Based Sentiment Analysis of Arabic Hotels' reviews using morphological, syntactic and semantic features. *Inf. Process. Manag.* **2019**, *56*, 308–319. [CrossRef]
4. Dada, E.G.; Bassi, J.S.; Chiroma, H.; Abdulhamid, S.M.; Adetunmbi, A.O.; Ajibuwa, O.E. Machine learning for email spam filtering: Review, approaches and open research problems. *Heliyon* **2019**, *5*, e01802. [CrossRef] [PubMed]
5. Shrivas, A.K.; Dewangan, A.K.; Ghosh, S.M.; Singh, D. Development of proposed ensemble model for spam e-mail classification. *Inf. Technol. Control* **2021**, *50*, 411–423. [CrossRef]
6. Aldjanabi, W.; Dahou, A.; Al-Qaness, M.A.A.; Elaziz, M.A.; Helmi, A.M.; Damaševičius, R. Arabic offensive and hate speech detection using a cross-corpora multi-task learning model. *Informatics* **2021**, *8*, 69. [CrossRef]
7. Sun, G.; Wang, Z.; Zhao, J. Automatic text summarization using deep reinforcement learning and beyond. *Inf. Technol. Control* **2021**, *50*, 458–469. [CrossRef]
8. Li, Y.; Nie, X.; Huang, R. Web spam classification method based on deep belief networks. *Expert Syst. Appl.* **2018**, *96*, 261–270. [CrossRef]
9. Kapociute-Dzikiene, J.; Venckauskas, A.; Damasevicius, R. A comparison of authorship attribution approaches applied on the Lithuanian language. In Proceedings of the 2017 Federated Conference on Computer Science and Information Systems, FedCSIS 2017, Prague, Czech Republic, 3–6 September 2017; pp. 347–351.
10. Xu, B.; Lin, H.; Lin, Y.; Xu, K.; Wang, L.; Gao, J. Incorporating semantic word representations into query expansion for microblog information retrieval. *Inf. Technol. Control* **2019**, *48*, 626–636. [CrossRef]
11. Omoregbe, N.A.I.; Ndaman, I.O.; Misra, S.; Abayomi-Alli, O.O.; Damaševičius, R. Text messaging-based medical diagnosis using natural language processing and fuzzy logic. *J. Healthc. Eng.* **2020**, *2020*, 8839524. [CrossRef]
12. Ghosh, S.; Hiware, K.; Ganguly, N.; Mitra, B.; De, P. Emotion detection from touch interactions during text entry on smartphones. *Int. J. Hum.-Comput. Stud.* **2019**, *130*, 47–57. [CrossRef]
13. Ji, Z.; Pi, H.; Wei, W.; Xiong, B.; Wozniak, M.; Damasevicius, R. Recommendation Based on Review Texts and Social Communities: A Hybrid Model. *IEEE Access* **2019**, *7*, 40416–40427. [CrossRef]
14. Alonso, M.A.; Vilares, D.; Gómez-Rodríguez, C.; Vilares, J. Sentiment analysis for fake news detection. *Electronics* **2021**, *10*, 1348. [CrossRef]
15. Tesfagergish, S.G.; Damaševičius, R.; Kapočiūtė-Dzikienė, J. *Deep Fake Recognition in Tweets Using Text Augmentation, Word Embeddings and Deep Learning*; Springer: Cham, Switzerland, 2021; Volume 12954, pp. 523–538.
16. Al-rimy, B.A.S.; Maarof, M.A.; Shaid, S.Z.M. Crypto-ransomware early detection model using novel incremental bagging with enhanced semi-random subspace selection. *Future Gener. Comput. Syst.* **2019**, *101*, 476–491. [CrossRef]
17. Mansoor, M.; Ur Rehman, Z.; Shaheen, M.; Khan, M.A.; Habib, M. Deep learning based semantic similarity detection using text data. *Inf. Technol. Control* **2020**, *49*, 495–510. [CrossRef]

18. Tesfagergish, S.G.; Kapočiūtė-Dzikienė, J. Part-of-speech tagging via deep neural networks for northern-Ethiopic languages. *Inf. Technol. Control* **2020**, *49*, 482–494.
19. Alfonse, M.; Gawich, M. A novel methodology for Arabic news classification. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* **2022**, *12*, e1440. [CrossRef]
20. Alruily, M. Classification of arabic tweets: A review. *Electronics* **2021**, *10*, 1143. [CrossRef]
21. Uysal, A.K.; Gunal, S. The impact of preprocessing on text classification. *Inf. Process. Manag.* **2014**, *50*, 104–112. [CrossRef]
22. Ayedh, A.; Tan, G.; Rajeh, H. The Impact of Feature Reduction Techniques on Arabic Document Classification. *Int. J. Database Theory Appl.* **2016**, *9*, 67–80. [CrossRef]
23. Ayedh, A.; TAN, G.; Alwesabi, K.; Rajeh, H. The Effect of Preprocessing on Arabic Document Categorization. *Algorithms* **2016**, *9*, 27. [CrossRef]
24. Kou, G.; Yang, P.; Peng, Y.; Xiao, F.; Chen, Y.; Alsaadi, F.E. Evaluation of feature selection methods for text classification with small datasets using multiple criteria decision-making methods. *Appl. Soft Comput.* **2019**, *86*, 105836. [CrossRef]
25. Larkey, L.S.; Ballesteros, L.; Connell, M.E. Improving stemming for Arabic information retrieval. In Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Tampere, Finland, 11–15 August 2002; p. 275. [CrossRef]
26. Al-Anzi, F.S.; AbuZeina, D. Beyond vector space model for hierarchical Arabic text classification: A Markov chain approach. *Inf. Process. Manag.* **2018**, *54*, 105–115. [CrossRef]
27. Kohler, M.; Vellasco, M.M.; Tanscheit, R. PSO+: A new particle swarm optimization algorithm for constrained problems. *Appl. Soft Comput.* **2019**, *85*, 105865. [CrossRef]
28. Al-qaness, M.A.; Ewees, A.A.; Fan, H.; AlRassas, A.M.; Abd Elaziz, M. Modified aquila optimizer for forecasting oil production. *Geo-Spat. Inf. Sci.* **2022**, 1–17. [CrossRef]
29. Unler, A.; Murat, A. A discrete particle swarm optimization method for feature selection in binary classification problems. *Eur. J. Oper. Res.* **2010**, *206*, 528–539. [CrossRef]
30. Engelbrecht, A.P.; Grobler, J.; Langeveld, J. Set based particle swarm optimization for the feature selection problem. *Eng. Appl. Artif. Intell.* **2019**, *85*, 324–336. [CrossRef]
31. Malhotra, R.; Khanna, M. Particle swarm optimization-based ensemble learning for software change prediction. *Inf. Softw. Technol.* **2018**, *102*, 65–84. [CrossRef]
32. Janani, R.; Vijayarani, S. Text document clustering using Spectral Clustering algorithm with Particle Swarm Optimization. *Expert Syst. Appl.* **2019**, *134*, 192–200. [CrossRef]
33. Eberhart, R.C.; Kennedy, J.A. New Optimizer Using Particle Swarm. In Proceedings of the Sixth International Symposium on Micro Machine and Human Science, Nagoya, Japan, 4–6 October 1995; pp. 39–43.
34. Al-Shargabi, B.; Al-Romimah, W.; Olayah, F. A comparative study for Arabic text classification algorithms based on stop words elimination. In Proceedings of the 2011 International Conference on Intelligent Semantic Web-Services and Applications, Amman, Jordan, 18–20 April 2011; p. 11.
35. Yousif, S.A.; Samawi, V.W.; Elkabani, I. Enhancement of Arabic Text Classification Using Semantic Relations with Part of Speech Tagger. *Adv. Electr. Comput. Eng.* **2015**, 195–201. [CrossRef]
36. Chantar, H.K.; Corne, D.W. Feature subset selection for Arabic document categorization using BPSO-KNN. In Proceedings of the 2011 Third World Congress on Nature and Biologically Inspired Computing, Salamanca, Spain, 19–21 October 2011; pp. 546–551. [CrossRef]
37. Sabbah, T.; Ayyash, M.; Ashraf, M. Support Vector Machine based Feature Selection Method for Text Classification. In Proceedings of the International Arab Conference on Information Technology, Yassmine Hammamet, Tunisia, 22–24 December 2017.
38. Saad, M.; Ashour, W. OSAC: Open Source Arabic Corpora. In Proceedings of the 6th ArchEng International Symposiums, EEECS'10 the 6th International Symposium on Electrical and Electronics Engineering and Computer Science, Lefke, North Cyprus, 25–26 November 2010; pp. 118–123. [CrossRef]
39. Abuaiadah, D.; El Sana, J.; Abusalah, W. On the impact of dataset characteristics on arabic document classification. *Int. J. Comput. Appl.* **2014**, *101*, 31–38. [CrossRef]
40. Bahassine, S.; Madani, A.; Al-Sarem, M.; Kissi, M. Feature selection using an improved Chi-square for Arabic text classification. *J. King Saud Univ. Comput. Inf. Sci.* **2018**, *32*, 225–231. [CrossRef]
41. Sharef, B.T.; Omar, N.; Sharef, Z.T. An automated arabic text categorization based on the frequency ratio accumulation. *Int. Arab J. Inf. Technol.* **2014**, *11*, 213–221.
42. Al-Tahrawi, M.M.; Al-Khatib, S.N. Arabic text classification using Polynomial Networks. *J. King Saud Univ.-Comput. Inf. Sci.* **2015**, *27*, 437–449. [CrossRef]
43. Al-Tahrawi, M.M. Arabic Text Categorization Using Logistic Regression. *Int. J. Intell. Syst. Appl.* **2015**, *7*, 71–78. [CrossRef]
44. Sammouda, R. A comparative study of effective supervised learning methods on arabic text classification. *Int. J. Comput. Sci. Netw. Secur.* **2017**, *17*, 130–133.
45. Abdelaal, H.M.; Youness, H.A.; Ahmed, A.M.; Ghribi, W. Knowledge Discovery in the Hadith according to the reliability and memory of the reporters using Machine learning techniques. *IEEE Access* **2019**, *7*, 157741–157755. [CrossRef]
46. Abdelaal, H.M.; Elemary, B.R.; Youness, H.A. Classification of Hadith According to Its Content Based on Supervised Learning Algorithms. *IEEE Access* **2019**, *7*, 152379–152387. [CrossRef]

47. Einea, O.; Elnagar, A.; Debsi, R.A. SANAD: Single-label Arabic News Articles Dataset for automatic text categorization. *Data Brief* **2019**, *25*, 104076. [CrossRef]
48. Alhaj, Y.A.; Xiang, J.; Zhao, D.; Al-Qaness, M.A.; Elaziz, M.A.; Dahou, A. A Study of the Effects of Stemming Strategies on Arabic Document Classification. *IEEE Access* **2019**, *7*, 32664–32671. [CrossRef]
49. Alhaj, Y.A.; Wickramaarachchi, W.U.; Hussain, A.; Al-Qaness, M.A.; Abdelaal, H.M. Efficient Feature Representation Based on the Effect of Words Frequency for Arabic Documents Classification. In Proceedings of the 2nd International Conference on Telecommunications and Communication Engineering, Beijing, China, 28–30 November 2018; pp. 397–401. [CrossRef]
50. Flores, F.N.; Moreira, V.P. Assessing the impact of Stemming Accuracy on Information Retrieval—A multilingual perspective. *Inf. Process. Manag.* **2016**, *52*, 840–854. [CrossRef]
51. Abainia, K.; Ouamour, S.; Sayoud, H. A novel robust Arabic light stemmer. *J. Exp. Theor. Artif. Intell.* **2017**, *29*, 557–573. [CrossRef]
52. Karisani, P.; Rahgozar, M.; Oroumchian, F. A query term re-weighting approach using document similarity. *Inf. Process. Manag.* **2016**, *52*, 478–489. [CrossRef]
53. Salton, G.; Buckley, C. Improving retrieval performance by relevance feedback. *J. Am. Soc. Inf. Sci.* **1990**, *41*, 288–297. [CrossRef]
54. Wang, H.; Hong, M. Supervised Hebb rule based feature selection for text classification. *Inf. Process. Manag.* **2019**, *56*, 167–191. [CrossRef]
55. Rehman, A.; Javed, K.; Babri, H.A. Feature selection based on a normalized difference measure for text classification. *Inf. Process. Manag.* **2017**, *53*, 473–489. [CrossRef]
56. Liu, N.; Qi, E.S.; Xu, M.; Gao, B.; Liu, G.Q. A novel intelligent classification model for breast cancer diagnosis. *Inf. Process. Manag.* **2019**, *56*, 609–623. [CrossRef]
57. Liu, Y.; Tian, J.; Feng, G.; Hu, Z. A relief supplies purchasing model via option contracts. *Comput. Ind. Eng.* **2019**, *137*, 106009. [CrossRef]
58. Tuncer, T.; Dogan, S.; Acharya, U.R. Automated detection of Parkinson’s disease using minimum average maximum tree and singular value decomposition method with vowels. *Biocybern. Biomed. Eng.* **2019**, *40*, 211–220. [CrossRef]
59. Deun, K.V.; Thorrez, L.; Coccia, M.; Hasdemir, D.; Westerhuis, J.A.; Smilde, A.K.; Mechelen, I.V. Weighted sparse principal component analysis. *Chemom. Intell. Lab. Syst.* **2019**, *195*, 103875. [CrossRef]
60. Al-Salemi, B.; Ayob, M.; Kendall, G.; Noah, S.A.M. Multi-label Arabic text categorization: A benchmark and baseline comparison of multi-label learning algorithms. *Inf. Process. Manag.* **2019**, *56*, 212–227. [CrossRef]
61. Follett, L.; Geletta, S.; Laugerman, M. Quantifying risk associated with clinical trial termination: A text mining approach. *Inf. Process. Manag.* **2019**, *56*, 516–525. [CrossRef]
62. Dev, V.A.; Eden, M.R. Formation lithology classification using scalable gradient boosted decision trees. *Comput. Chem. Eng.* **2019**, *128*, 392–404. [CrossRef]
63. Bharath Bhushan, S.N.; Danti, A. Classification of text documents based on score level fusion approach. *Pattern Recognit. Lett.* **2017**, *94*, 118–126. [CrossRef]
64. Wang, D.; Zhang, X.; Fan, M.; Ye, X. Hierarchical mixing linear support vector machines for nonlinear classification. *Pattern Recognit.* **2016**, *59*, 255–267. [CrossRef]
65. Sharma, A. Guided Stochastic Gradient Descent Algorithm for inconsistent datasets. *Appl. Soft Comput.* **2018**, *73*, 1068–1080. [CrossRef]