Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2023.034196

*Article*

# Modified Dragonfly Optimization with Machine Learning Based Arabic Text Recognition

**Badriyya B. Al-onazi[1], Najm Alotaibi[2], Jaber S. Alzahrani[3], Hussain Alshahrani[4], Mohamed Ahmed Elfaki[4], Radwa Marzouk[5], Mahmoud Othman[6] and Abdelwahed Motwakel[7,*]**

[1]Department of Language Preparation, Arabic Language Teaching Institute, Princess Nourah bint Abdulrahman University, P.O. Box 84428, Riyadh, 11671, Saudi Arabia

[2]Prince Saud AlFaisal Institute for Diplomatic Studies, Riyadh, Saudi Arabia

[3]Department of Industrial Engineering, College of Engineering at Alqunfudah, Umm Al-Qura University, Makkah, Saudi Arabia

[4]Department of Computer Science, College of Computing and Information Technology, Shaqra University, Shaqra, Saudi Arabia

[5]Department of Information Systems, College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University, P.O. Box 84428, Riyadh, 11671, Saudi Arabia

[6]Department of Computer Science, Faculty of Computers and Information Technology, Future University in Egypt, New Cairo, 11835, Egypt

[7]Department of Computer and Self Development, Preparatory Year Deanship, Prince Sattam bin Abdulaziz University, AlKharj, Saudi Arabia

*Corresponding Author: Abdelwahed Motwakel. Email: a.ismaeil@psau.edu.sa

Received: 08 July 2022; Accepted: 27 August 2022; Published: 30 August 2023

**Abstract:** Text classification or categorization is the procedure of automatically tagging a textual document with most related labels or classes. When the number of labels is limited to one, the task becomes single-label text categorization. The Arabic texts include unstructured information also like English texts, and that is understandable for machine learning (ML) techniques, the text is changed and demonstrated by numerical value. In recent times, the dominant method for natural language processing (NLP) tasks is recurrent neural network (RNN), in general, long short term memory (LSTM) and convolutional neural network (CNN). Deep learning (DL) models are currently presented for deriving a massive amount of text deep features to an optimum performance from distinct domains such as text detection, medical image analysis, and so on. This paper introduces a Modified Dragonfly Optimization with Extreme Learning Machine for Text Representation and Recognition (MDFO-EMTRR) model on Arabic Corpus. The presented MDFO-EMTRR technique mainly concentrates on the recognition and classification of the Arabic text. To achieve this, the MDFO-EMTRR technique encompasses data pre-processing to transform the input data into compatible format. Next, the ELM model is utilized for the representation and recognition of the Arabic text. At last, the MDFO algorithm was exploited for optimal tuning of the parameters related to the ELM method and thereby accomplish enhanced classifier results. The experimental result analysis of the MDFO-EMTRR system was performed on benchmark datasets and attained maximum accuracy of 99.74%.

**Keywords:** Arabic corpus; dragonfly algorithm; machine learning; text mining; extreme learning machine

## 1 Introduction

The text mining issue grabs more interest and it turns out to be an important research field due to the boom of textual applications namely social networking gates and news article portals [1]. Text classification was one such great difficulty because of the rising quantity of documents. Many works in text classification were made in English [2]. Inappropriately, several research works utilizing Arabic dataset becomes rarer now and requires larger examination [3]. There were numerous text classifier techniques like decision tree (DT), Naïve Bayes (NB), k-nearest neighbour (K-NN), and support vector machine (SVM). Text classification is a processing method which regularly allocates a file which is given to its category or class [4]. There were several studies in publications which deals with classification of text from many languages such as Russian, English, Chinese, and many other languages [5]. Nearly 447 billion individuals speak Arabic as a native speaker meanwhile higher than this number non-Arab however it requires Arabic language as a religious language. Moreover, it becomes official language of twenty-two nations [6,7]. In addition, several other languages consider Arabic as root language. There were twenty-nine characters or letters in the Arabic language that should be written in right to left [8,9]. This was the 5th most famous language worldwide. Arabic owns an ironic geomorphology and a complicated orthography that generates distinct words with various meanings [10]. This language contains particular words stated as Arab vowels (أ-ي-و) that mandate an unusual mechanism of grammar and morphology. The remaining are known as consonant words [11].

In Arabic language, the character method changes in accordance with its place across the globe. It is written disconnected or connected at the end, located inside the word, or discovered at the opening [12]. Also, short vowels or diacritics control word phonology and modify its sense. Such features indulge difficulties in word representation and embedding. Additional difficulties for Arabic language processing were stemming, dialects, phonology, and orthography [13]. Additionally, the Arabic nature relevant difficulties, the efficacy of word embedding was task-based and is impacted by plenty of task-based words. Thus, a suitable Arabic text depiction was needed for manipulating such exceptional features [14]. Text detection was one such great difficulty of the natural language processing (NLP) that has several restrictions to operate with, particularly for the Arabic language than English [15]. Deep learning (DL) implies various structures can learn features which were internally identified at the training time.

This paper introduces a Modified Dragonfly Optimization with Extreme Learning Machine for Text Representation and Recognition (MDFO-EMTRR) model on Arabic Corpus. The presented MDFO-EMTRR technique mainly concentrates on the recognition and classification of Arabic text. To achieve this, the MDFO-EMTRR technique encompasses data pre-processing to transform the input data into compatible format. Next, the ELM model is utilized for the representation and recognition of the Arabic text. At last, the MDFO algorithm was exploited for optimal tuning of the parameters related to the ELM method and thereby accomplish enhanced classifier results. The experimental result study of MDFO-EMTRR algorithm is performed on benchmark dataset.

The rest of the paper is organized as follows. Section 2 offers the literature review and Section 3 introduces the proposed model. Next, Section 4 provides experimental validation and Section 5 concludes the work.

## 2 Literature Review

Azmi et al. [13] introduced a spell checking tool which identifies and corrects real-word errors for the Arabic language. It eliminates pre-defined confusion sets employed by several models on large scale real time errors. Ahmed et al. [14] presented a detailed review on Arabic cursive scene text recognition. The presented techniques following a DL model which is equally appropriate to design Arabic cursive scene text recognition system. In [15], an effective Bidirectional long short term memory Network (BiLSTM) is examined for enhancing Arabic Sentiment Analysis using Forward-Backward summarize contextual data from Arabic feature sequences.

Butt et al. [16] provided a convolutional neural network-recurrent neural network (CNN-RNN) method having an attention system for Arabic image text detection. The method uses an input image and produces feature series via a CNN. Such series were transmitted to a bi-directional RNN for obtaining feature series in order. The bi-directional RNN could miss certain pre-processing of text segmentation. Thus, a bi-directional RNN having an attention system can be utilized for generating output, allowing the method for selecting related data from the feature series. An attention system applies end-to-end training via a standard back propagation (BP) technique. Muaad et al. [17] suggested a method for representing and recognizing Arabic text at character phase on the basis of the ability of a deep convolutional neural network (CNN). This mechanism has been authenticated by making use of fivefold cross-validation (CV) tests for Arabic text file classifier. And utilized suggested presented mechanism for evaluating Arabic text. The Arabic text computer-aided recognition (ArCAR) mechanism displays its ability for classifying Arabic text at character level. Saleem et al. [18] suggested a novel effective segmentation method by altering an artificial neural network (ANN) method and making appropriate for binarization level depending on blocks. This altered technique was integrated with a novel effectual rotation method for achieving precise segmentation via the histogram analysis of binary images. In addition, recommends a novel structure for precise text rotation which is helpful in establishing a segmentation technique which could ease the text extraction from its background.

In [19], a new DL Arabic text computer-aided recognition (ArCAR) was suggested for representing and recognizing Arabic text at character phase. The input Arabic text was quantized in 1D vector format for every Arabic character for indicating a 2D range for ArCAR mechanism. The ArCAR mechanism was authenticated over fivefold computer vision (CV) tests for 2 applications they are Arabic sentiment analysis and Arabic text document classifications. Ahmad et al. [20] provided a multi-stage Hidden Markov Model (HMM)-related text recognition mechanism for handwritten Arabic. Moreover, contextual HMM modeling using such sub-core shapes was provided that illustrates sub-core shapes as methods that enhance contextual HMM mechanism when compared to a contextual HMM mechanism using the standard Arabic character shapes as algorithms, and its results in suggestively compact recognizer in the meantime.

## 3 The Proposed Model

In this study, a new MDFO-EMTRR technique was established for the recognition and classification of Arabic text. To achieve this, the MDFO-EMTRR technique encompasses data pre-processing to transform the input data into compatible format. Next, the ELM model is utilized for the representation and recognition of the Arabic text. At last, the MDFO algorithm is exploited for optimal tuning of the parameters related to the ELM technique and thereby accomplish enhanced classifier results. Fig. 1 demonstrates the block diagram of MDFO-EMTRR approach.

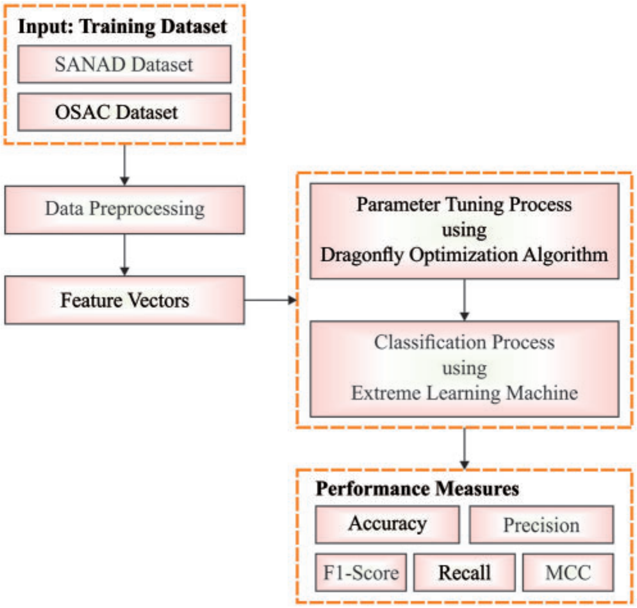


**Figure 1:** Block diagram of MDFO-EMTRR approach

### 3.1 Data Pre-processing

Arabic text pre-processing indicates the initial level of any text classifier workflow. It can be utilized for cleaning and making the unstructured text dataset for improvising entire execution of the suggested ArCAR mechanism [21]. The Arabic language nature contains more structural complexities than English, where it requires several extra pre-processing efforts, like difficult morphology, stemming, and normalization of numerous roots detected for a particular single word. These complexities made Arabic text depiction really a complex task and affect complete correctness of classifiers. For minimizing these complexities, indicate the Arabic text related to the character level in place of sentence level or word level for both applications they are Arabic files Arabic sentiment analysis, and text classification. Therefore, normalization and stemming were not needed, and this was a key to facilitating Arabic text preparation. Next, a lookup table can be produced as a 2D tensor of size (f0, l) which comprises embedded of l characters, and f0 is indicated as the RGB image dimension of input text.

### 3.2 ELM Based Text Recognition

In this study, the ELM method can be employed for the representation and recognition of the Arabic text. Essentially, ELM was a single hidden layer feedforward neural network (SLFN) approach. The variation amongst ELM and SLFN are in weight of output and hidden layers were updated [22]. In SLFN, weight of output and input neurons are arbitrarily initialized, and weight of layer is upgraded through BP mechanism. Consider trained dataset by $\left(x_j, t_j\right)$ where $x_j = [x_{j1}, x_{j2}, \ldots, x_{jN}]^T$ characterizes the input vector and $t_j$ symbolizes the resultant vector. The output of $j^{th}$ hidden layers are characterized by $g\left(w_i, b_i, x_j\right)$, where $w_i$ specifies weight vector interconnected with input layer to $i^{th}$ hidden state $b_i$ indicates the bias of $i^{th}$ hidden layers and $g$ represents the activation function. Each hidden state of ELM is linked to each resultant layer with connected weight, they characterize the weight interconnecting the $i^{th}$ hidden layers with resultant layer as $\beta_i$. The mathematical expression can be given in the following,

$$\sum\nolimits_{i=1}^{L} \beta_i g\left(w_i, b_i, x_j\right) = t_j \tag{1}$$

From the equation, $L$ indicates the hidden layer count, and $j$ shows the output or input samples of $N$ training instance. The abovementioned equation is written as follows:

$$H\beta = T \tag{2}$$

Let $m$ output node is regarded as

$$\beta = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_L^T \end{bmatrix}_{L\times m} \quad and\ T = \begin{bmatrix} t_1^T \\ \vdots \\ t_N^T \end{bmatrix}_{N\times m} \tag{3}$$

$H$ represents the output matrix of hidden state as:

$$H = \begin{bmatrix} g\left(w_1, b_1, x_1\right) & \cdots & g\left(w_{\mathrm{L}}, b_{\mathrm{L}}, x_1\right) \\ \vdots & \ddots & \vdots \\ g\left(w_1, b_1, x_N\right) & \cdots & g\left(w_{\mathrm{L}}, b_{\mathrm{L}}, x_N\right) \end{bmatrix} \tag{4}$$

The minimal norm least square of (2) can be

$$\hat{\beta} = H^{+}T \tag{5}$$

In Eq. (5), $H^{+}$ the Moore-Penrose generalized inverse of matrix $H$. $H^{+}$ is estimated using orthogonal projection method, singular value decomposition (SVD), orthogonalization model, and QR methodology. Fig. 2 depicts infrastructure of ELM. It should regulate the system (to evade overfitting), the optimization issue becomes:

$$\min\left(\|\beta\|^2 + C\sum\nolimits_{i=1}^{N} \|\xi_i^T\|^2\right) \tag{6}$$

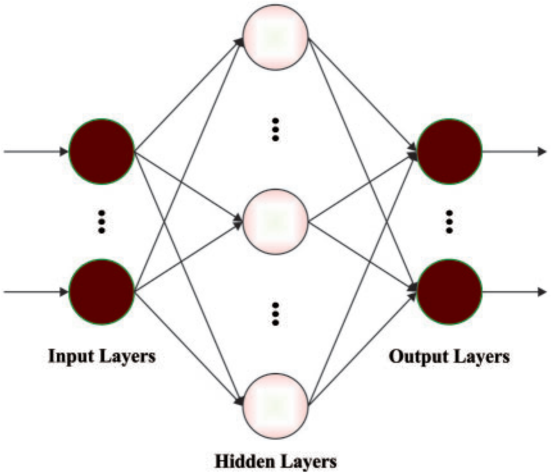


**Figure 2:** Structure of ELM

In Eq. (6), $\xi_i = t_i^T - h(x_i)\beta$ signifies the trainable error of $i^{th}$ instances and $C$ represent the penalty factor. It converts the problem to its binary format and generates the Lagrangian function by:

$$F = \|\beta\|^2 + C\sum\nolimits_{i=1}^{N} \|\xi\|^2 - \sum\nolimits_{i=1}^{N}\sum\nolimits_{j=1}^{L} \alpha_{ij}\left(h(x_i)\beta_j - t_{ij} + \xi_{ij}\right) \tag{7}$$

Considers the partial differential coefficient of abovementioned expression and employ Karush–Kuhn–Tucker (KKT). If $L < N$ the size of matrix $H^T H$ was lower than $HH^T$,

$$\beta = H^+ T = \left(\frac{I}{C} + H^T H\right)^{-1} H^T T \tag{8}$$

Therefore, the final output of ELM can be

$$f(x) = h(x)\beta = h(x)\left(\frac{I}{C} + H^T H\right)^{-1} H^T T \tag{9}$$

If $L \succ N$, the size of matrix $HH^T$ is lower than that of matrix $H^T H$, the solution of formula is denoted as follows:

$$\beta = H^+ T = H^T\left(\frac{I}{C} + HH^T\right)^{-1} T \tag{10}$$

As a result, the final resultant of ELM is

$$f(x) = h(x)\beta = h(x)H^T\left(\frac{I}{C} + HH^T\right)^{-1} T \tag{11}$$

For the dual classification issues, decision function of ELM is exhibited as follows:

$$f(x) = sign(h(x)\beta) \tag{12}$$

For multi-class sample, the class label of sample is written as follows:

$$label\ (x) = \underset{1\le i\le m}{\arg\max}\{f_i(x)\} \tag{13}$$

Then

$$f(x) = r_1(x), f_2(x), f_3(x), \cdots, f_n(x)]^T \tag{14}$$

ELM is applied to the classification and predictive tasks in different areas.

### *3.3 Parameter Optimization Process*

At this stage, the MDFO algorithm was exploited for optimal tuning of the parameters related to the ELM technique and thereby accomplish enhanced classifier results. Mirjalili in [23] established DFO algorithm which is a metaheuristic approach motivated by the static and dynamic strategies of dragonflies (DFs). There exist 2 considerable levels of optimization: exploitation and exploration. The two phases are modelled through DF, similarly static or dynamic searching for food or avoiding the enemy. There exist 2 cases whereby SI performs in DFs: migration and feeding. The migration is demonstrated as dynamic swarm; feeding can be presented by a static swarm. The swarm exist three certain strategies: cohesion, separation, and alignment. Now, the separation denotes that separate in swarm prevent static collision with neighbor. Alignment indicates the speed at the agent was respective to neighboring individual. Finally, the cohesion demonstrates the tendency of individual neighboring centers of herd. The reason to add the performance to this approach is that significant drive of each swarm is for enduring. Therefore, each individual move nearby to the food source, which prevents the enemy in same time period. Each performance is arithmetically processed by the following equation:

$$S_i = -\sum_{j=1}^{N} X - X_j, \tag{15}$$

$$A_i = \frac{\sum_{j=1}^{N} V_j}{N}, \tag{16}$$

$$C_i = \frac{\sum_{j=1}^{N} X_j}{N} - X, \tag{17}$$

$$F_j = X^{+} - X, \tag{18}$$

$$E_i = X^{-} + X. \tag{19}$$

Here, $X$ indicates the instant location of individual however $X_j$ represents instant place of $j^{\text{th}}$ individual. $N$ denotes the amount of contiguous individuals but $V_j$ illustrates the speed of $j^{\text{th}}$ nearby individual. $X^{+}$ and $X^{-}$ indicates the place of food and enemy sources.

For upgrading location of artificial dragonfly in the searching region and simulating the motion, 2 vectors are taken into account: step $(\Delta X)$ and place $(X)$. Also, the step vector is considered as speed that indicates the method of DF movement. Then, compute the step vector, and the location vector can be upgraded as [24]:

$$\nabla X_{t+1} = \left(sS_i + aA_i + cC_i + fF_j + eE_i\right) + w\nabla X_t, \tag{20}$$

$$X_{t+1} = X_t + \nabla X_{t+1}, \tag{21}$$

From the equations, the value of, $a$, and $c$ indicate the separation, alignment, and cohesion coefficients, $f$, $e$, $w$ and $t$ demonstrates the food factor, enemy factor, inertia coefficient, and iterative number. The stated factors and coefficients allow to carry out exploratory and exploitative efficiencies in the optimization. In the static movement, the alignment was low however the fit to attack the enemy was greater. Fig. 3 demonstrates the steps involved in DFA. In the dynamic swarm, DF is inclined to support the flight. Therefore, in the exploitation stage, the co-efficient of alignment was minimal and co-efficient of cohesion was maximal; the coefficient of alignment was maximal and the cohesion coefficient was minimal in the exploration stage. The MDFO algorithm was developed by utilize of Levy flight concept in the DFO algorithm. Levy Flight was a kind of artificial and natural phenomenon, characterized by Levy statistics. It was a type of stochastic non-Gaussian walk whereby the step length values should be disseminated according to the Levy uniform distribution:

$$Levy\,(\beta)\ \sim u = t^{-1-\beta},\ 0 < \beta \leq 2 \tag{22}$$

In Eq. (22), $\beta$ indicates a Levy index to alter steadiness. The Levy random value is defined in the following:

$$Levy\,(\beta) \sim \frac{\varphi \times \mu}{|\mathrm{v}|^{1/\beta}} \tag{23}$$

In Eq. (23) $\mu$ & $v$ indicates uniform distribution, $\Gamma$ represents uniform Gamma function, $\beta = 1.5$, & $\varphi$ is characterized by:

$$\varphi = \left[ \frac{\Gamma (1+\beta) \times \sin\left(\pi \times \frac{\beta}{2}\right)}{\Gamma\left(\left(\frac{1+\beta}{2}\right) \times \beta \times 2^{\frac{\beta-1}{2}}\right)} \right]^{\frac{1}{\beta}} \tag{24}$$

In order to accomplish tradeoff among exploration and exploitation abilities of AOA, LF technique is applied for updating the location of the search agent as:

$$X_i^{levy} = X_i + r \oplus levy\,(\beta) \tag{25}$$

In Eq. (25), $X_i^{levy}$ indicates recently accomplished location of *i-th* search agent $X_i$ and $r$ designates arbitrary vector and $\oplus$ denotes dot product.

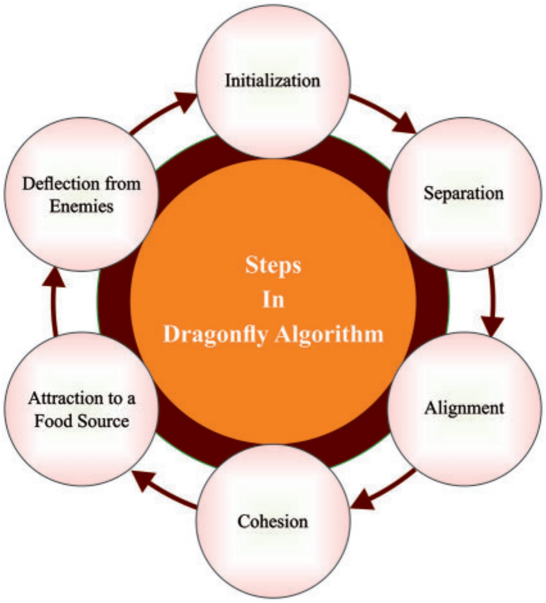


**Figure 3:** Steps involves in DFA

## 4 Experimental Validation

The experimental validation of the MDFO-EMTRR method is tested by utilizing the SANAD dataset and OSAC dataset. Table 1 defines the detailed description of SANAD dataset under seven distinct classes. The dataset holds 45500 documents.

**Table 1:** Details on SANAD dataset

| Label | Class | No. of documents |
|---|---|---|
| C-1 | Finance | 6500 |
| C-2 | Sports | 6500 |

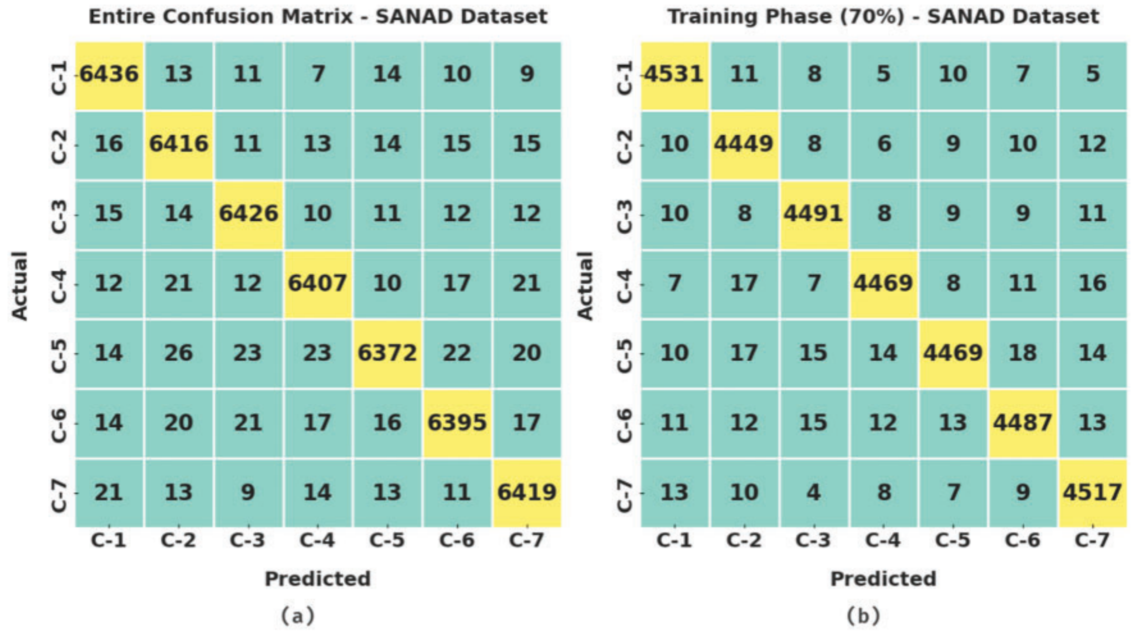
(Continued)

**Table 1 (continued)**

| Label | Class | No. of documents |
|---|---|---|
| C-3 | Culture | 6500 |
| C-4 | Technology | 6500 |
| C-5 | Politics | 6500 |
| C-6 | Medical | 6500 |
| C-7 | Religion | 6500 |
| Total number of documents | | 45500 |

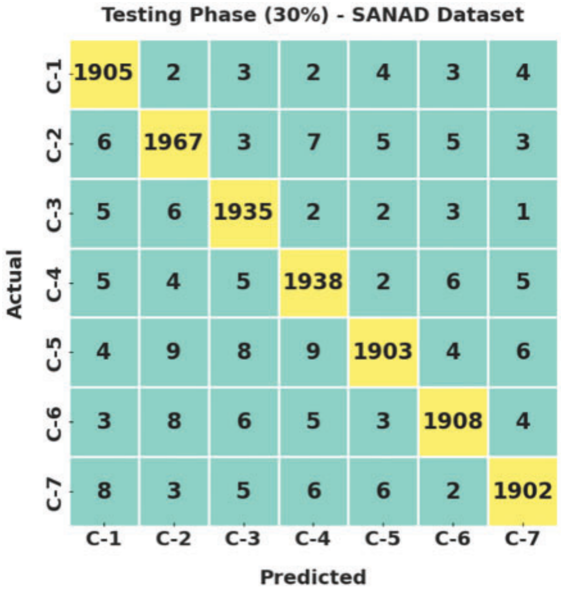
Fig. 4 depicts the confusion matrices produced by the MDFO-EMTRR technique on the test SANAD dataset. The figures demonstrated the MDFO-EMTRR approach has shown effectual outcomes on the applied SANAD dataset.

Table 2 and Fig. 5 exemplify the classification performance of the MDFO-EMTRR algorithm on test SANAD dataset. The outcomes demonstrated the MDFO-EMTRR model has resulted in effectual results. For instance, on entire dataset, the MDFO-EMTRR model has provided average $accu_y$ of 99.61%, $prec_n$ of 98.62%, $reca_l$ of 98.62%, $F_{score}$ of 98.62%, and Mathew Correlation Coefficient (MCC) of 98.39%. Also, on 70% of TR data, the MDFO-EMTRR method has offered average $accu_y$ of 99.61%, $prec_n$ of 98.63%, $reca_l$ of 98.63%, $F_{score}$ of 98.63%, and MCC of 98.40%. In addition, on 30% of TS data, the MDFO-EMTRR technique has rendered average $accu_y$ of 99.60%, $prec_n$ of 98.60%, $reca_l$ of 98.59%, $F_{score}$ of 98.59%, and MCC of 98.36%.



(a) (b)

**Figure 4:** (Continued)



| | C-1 | C-2 | C-3 | C-4 | C-5 | C-6 | C-7 |
|---|---|---|---|---|---|---|---|
| C-1 | 1905 | 2 | 3 | 2 | 4 | 3 | 4 |
| C-2 | 6 | 1967 | 3 | 7 | 5 | 5 | 3 |
| C-3 | 5 | 6 | 1935 | 2 | 2 | 3 | 1 |
| C-4 | 5 | 4 | 5 | 1938 | 2 | 6 | 5 |
| C-5 | 4 | 9 | 8 | 9 | 1903 | 4 | 6 |
| C-6 | 3 | 8 | 6 | 5 | 3 | 1908 | 4 |
| C-7 | 8 | 3 | 5 | 6 | 6 | 2 | 1902 |

(c)

**Figure 4:** Confusion matrices of MDFO-EMTRR approach under SANAD dataset (a) entire dataset, (b) 70% of TR data, and (c) 30% of TS data

**Table 2:** Result analysis of MDFO-EMTRR approach with various measures under SANAD dataset

| Labels | Accuracy | Precision | Recall | F-Score | MCC |
|---|---|---|---|---|---|
| | | | Entire dataset | | |
| C-1 | 99.66 | 98.59 | 99.02 | 98.80 | 98.60 |
| C-2 | 99.58 | 98.36 | 98.71 | 98.53 | 98.29 |
| C-3 | 99.65 | 98.66 | 98.86 | 98.76 | 98.56 |
| C-4 | 99.61 | 98.71 | 98.57 | 98.64 | 98.41 |
| C-5 | 99.55 | 98.79 | 98.03 | 98.41 | 98.15 |
| C-6 | 99.58 | 98.66 | 98.38 | 98.52 | 98.28 |
| C-7 | 99.62 | 98.56 | 98.75 | 98.66 | 98.43 |
| Average | 99.61 | 98.62 | 98.62 | 98.62 | 98.39 |
| | | Training phase (70%) | | | |
| C-1 | 99.66 | 98.67 | 98.99 | 98.83 | 98.64 |
| C-2 | 99.59 | 98.34 | 98.78 | 98.56 | 98.32 |
| C-3 | 99.65 | 98.75 | 98.79 | 98.77 | 98.56 |
| C-4 | 99.63 | 98.83 | 98.54 | 98.69 | 98.47 |
| C-5 | 99.55 | 98.76 | 98.07 | 98.41 | 98.15 |
| C-6 | 99.56 | 98.59 | 98.33 | 98.46 | 98.21 |
| C-7 | 99.62 | 98.45 | 98.88 | 98.67 | 98.44 |
| Average | 99.61 | 98.63 | 98.63 | 98.63 | 98.40 |

(Continued)

**Table 2 (continued)**

| Labels | Accuracy | Precision | Recall | F-Score | MCC |
|---|---|---|---|---|---|
| | | Testing phase (30%) | | | |
| C-1 | 99.64 | 98.40 | 99.06 | 98.73 | 98.52 |
| C-2 | 99.55 | 98.40 | 98.55 | 98.47 | 98.21 |
| C-3 | 99.64 | 98.47 | 99.03 | 98.75 | 98.54 |
| C-4 | 99.58 | 98.43 | 98.63 | 98.53 | 98.28 |
| C-5 | 99.55 | 98.86 | 97.94 | 98.40 | 98.13 |
| C-6 | 99.62 | 98.81 | 98.50 | 98.66 | 98.43 |
| C-7 | 99.61 | 98.81 | 98.45 | 98.63 | 98.40 |
| Average | 99.60 | 98.60 | 98.59 | 98.59 | 98.36 |

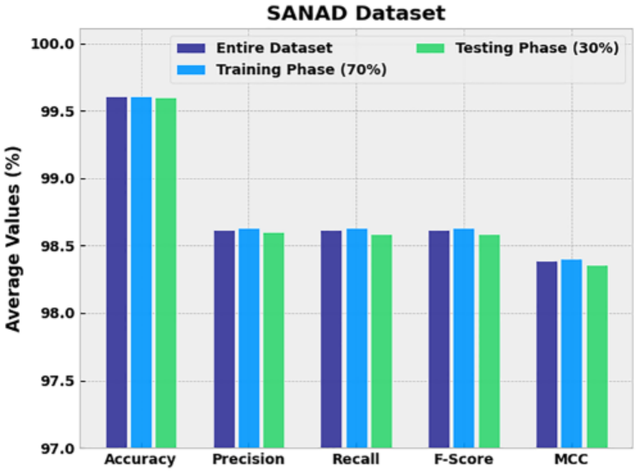


**Figure 5:** Average analysis of MDFO-EMTRR approach under SANAD dataset

The training accuracy (TA) and validation accuracy (VA) acquired by the MDFO-EMTRR method on SANAD Dataset is exhibited in Fig. 6. The experimental outcome denoted the MDFO-EMTRR method has attained maximal values of TA and VA. Particularly the VA is greater than TA.

The training loss (TL) and validation loss (VL) attained by the MDFO-EMTRR approach on SANAD Dataset were shown in Fig. 7. The experimental outcome signified the MDFO-EMTRR algorithm has established least values of TL and VL. To be Specific, the VL is lesser than TL.

A comparative inspection of the results offered by the MDFO-EMTRR model on SANAD dataset is provided in Table 3 and Fig. 8. The outcomes identified that the KELM model has shown lower $accu_y$ of 96.40% whereas the SVM model has resulted in slightly increased $accu_y$ of 97.91%. Moreover, the ELM, DT, and ArCAR models have demonstrated reasonable $accu_y$ of 98.38%, 98.07%, and 98.83% respectively. However, the MDFO-EMTRR model has shown maximum $accu_y$ of 99.60%.

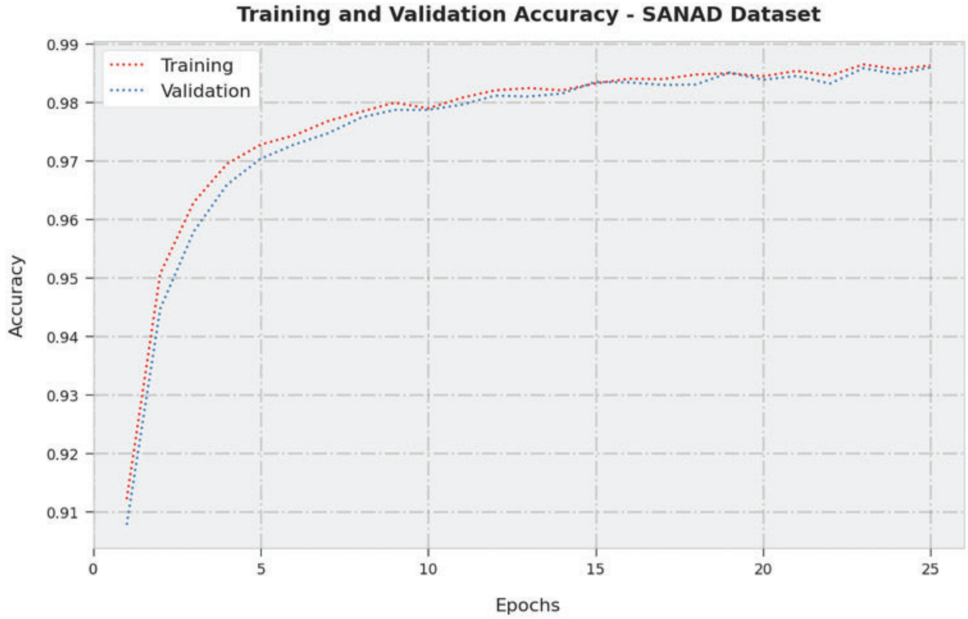


**Figure 6:** TA and VA analysis of MDFO-EMTRR approach under SANAD dataset

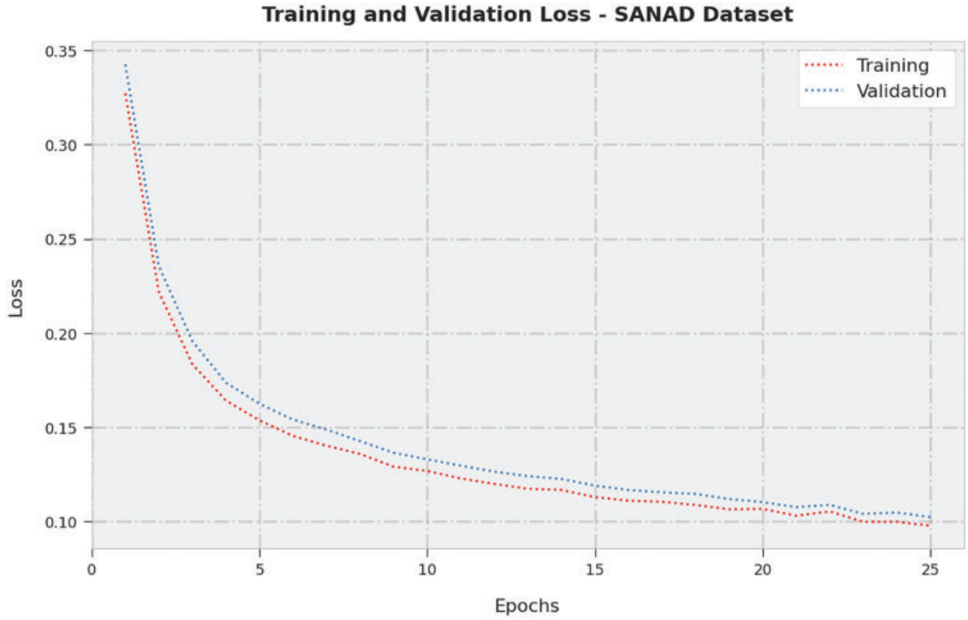


**Figure 7:** TL and VL analysis of MDFO-EMTRR approach under SANAD dataset

**Table 3:** Comparative analysis of MDFO-EMTRR approach with existing algorithms under SANAD dataset

| Methods | Accuracy (%) |
|---|---|
| MDFO-EMTRR | 99.60 |
| SVM model | 97.91 |
| ELM model | 98.38 |
| Decision tree classifier | 98.07 |
| KELM model | 96.40 |
| ArCAR model | 98.83 |

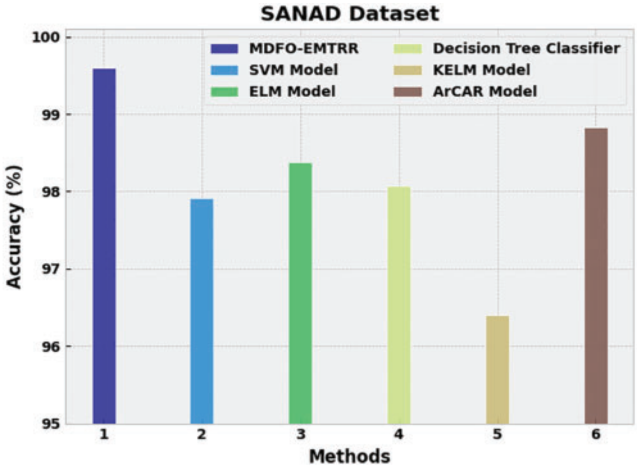


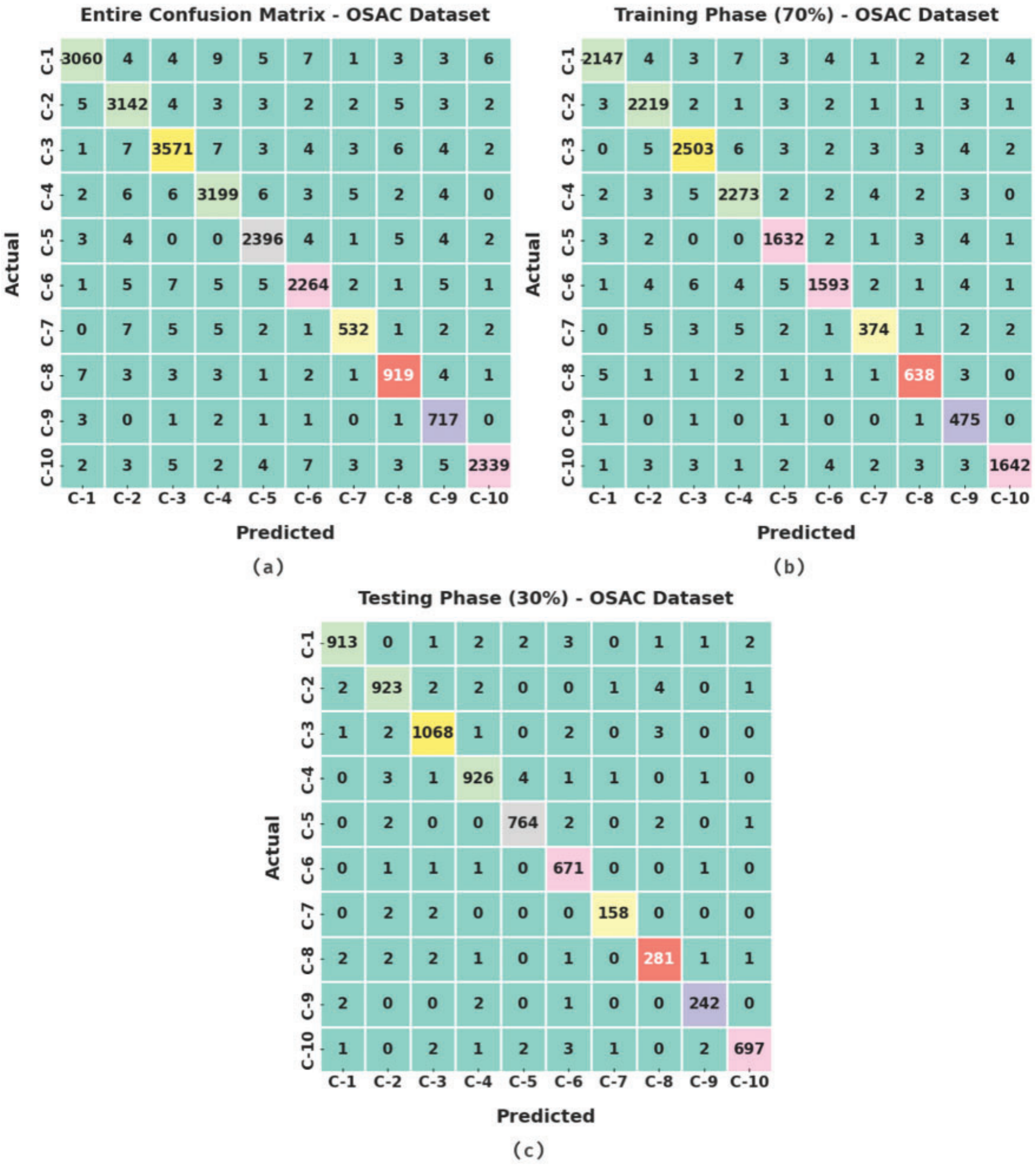
**Figure 8:** Comparative analysis of MDFO-EMTRR approach under SANAD dataset

Table 4 defines the detailed description of OSAC dataset under ten distinct classes. The dataset holds 22429 documents.

**Table 4:** Details on OSAC dataset

| Label | Class | No. of documents |
|---|---|---|
| C-1 | Economy | 3102 |
| C-2 | Religion | 3171 |
| C-3 | Education and family | 3608 |
| C-4 | History | 3233 |
| C-5 | Sports | 2419 |
| C-6 | Health | 2296 |
| C-7 | Astronomy | 557 |
| C-8 | Low | 944 |
| C-9 | Stories | 726 |
| C-10 | Cooking recipes | 2373 |
| Total number of documents | | 22429 |

Fig. 9 portrays the confusion matrices produced by the MDFO-EMTRR method on the test OSAC dataset. The figures exhibited the MDFO-EMTRR algorithm has shown effectual outcomes on the applied OSAC dataset.



Entire Confusion Matrix - OSAC Dataset

| Actual \ Predicted | C-1 | C-2 | C-3 | C-4 | C-5 | C-6 | C-7 | C-8 | C-9 | C-10 |
|---|---|---|---|---|---|---|---|---|---|---|
| C-1 | 3060 | 4 | 4 | 9 | 5 | 7 | 1 | 3 | 3 | 6 |
| C-2 | 5 | 3142 | 4 | 3 | 3 | 2 | 2 | 5 | 3 | 2 |
| C-3 | 1 | 7 | 3571 | 7 | 3 | 4 | 3 | 6 | 4 | 2 |
| C-4 | 2 | 6 | 6 | 3199 | 6 | 3 | 5 | 2 | 4 | 0 |
| C-5 | 3 | 4 | 0 | 0 | 2396 | 4 | 1 | 5 | 4 | 2 |
| C-6 | 1 | 5 | 7 | 5 | 5 | 2264 | 2 | 1 | 5 | 1 |
| C-7 | 0 | 7 | 5 | 5 | 2 | 1 | 532 | 1 | 2 | 2 |
| C-8 | 7 | 3 | 3 | 3 | 1 | 2 | 1 | 919 | 4 | 1 |
| C-9 | 3 | 0 | 1 | 2 | 1 | 1 | 0 | 1 | 717 | 0 |
| C-10 | 2 | 3 | 5 | 2 | 4 | 7 | 3 | 3 | 5 | 2339 |

(a)

Training Phase (70%) - OSAC Dataset

| Actual \ Predicted | C-1 | C-2 | C-3 | C-4 | C-5 | C-6 | C-7 | C-8 | C-9 | C-10 |
|---|---|---|---|---|---|---|---|---|---|---|
| C-1 | 2147 | 4 | 3 | 7 | 3 | 4 | 1 | 2 | 2 | 4 |
| C-2 | 3 | 2219 | 2 | 1 | 3 | 2 | 1 | 1 | 3 | 1 |
| C-3 | 0 | 5 | 2503 | 6 | 3 | 2 | 3 | 3 | 4 | 2 |
| C-4 | 2 | 3 | 5 | 2273 | 2 | 2 | 4 | 2 | 3 | 0 |
| C-5 | 3 | 2 | 0 | 0 | 1632 | 2 | 1 | 3 | 4 | 1 |
| C-6 | 1 | 4 | 6 | 4 | 5 | 1593 | 2 | 1 | 4 | 1 |
| C-7 | 0 | 5 | 3 | 5 | 2 | 1 | 374 | 1 | 2 | 2 |
| C-8 | 5 | 1 | 1 | 2 | 1 | 1 | 1 | 638 | 3 | 0 |
| C-9 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 475 | 0 |
| C-10 | 1 | 3 | 3 | 1 | 2 | 4 | 2 | 3 | 3 | 1642 |

(b)

Testing Phase (30%) - OSAC Dataset

| Actual \ Predicted | C-1 | C-2 | C-3 | C-4 | C-5 | C-6 | C-7 | C-8 | C-9 | C-10 |
|---|---|---|---|---|---|---|---|---|---|---|
| C-1 | 913 | 0 | 1 | 2 | 2 | 3 | 0 | 1 | 1 | 2 |
| C-2 | 2 | 923 | 2 | 2 | 0 | 0 | 1 | 4 | 0 | 1 |
| C-3 | 1 | 2 | 1068 | 1 | 0 | 2 | 0 | 3 | 0 | 0 |
| C-4 | 0 | 3 | 1 | 926 | 4 | 1 | 1 | 0 | 1 | 0 |
| C-5 | 0 | 2 | 0 | 0 | 764 | 2 | 0 | 2 | 0 | 1 |
| C-6 | 0 | 1 | 1 | 1 | 0 | 671 | 0 | 0 | 1 | 0 |
| C-7 | 0 | 2 | 2 | 0 | 0 | 0 | 158 | 0 | 0 | 0 |
| C-8 | 2 | 2 | 2 | 1 | 0 | 1 | 0 | 281 | 1 | 1 |
| C-9 | 2 | 0 | 0 | 2 | 0 | 1 | 0 | 0 | 242 | 0 |
| C-10 | 1 | 0 | 2 | 1 | 2 | 3 | 1 | 0 | 2 | 697 |

(c)

**Figure 9:** Confusion matrices of MDFO-EMTRR approach under OSAC dataset (a) entire dataset, (b) 70% of TR data, and (c) 30% of TS data

Table 5 demonstrates the classification performance of the MDFO-EMTRR method on the test OSAC dataset. The results exhibited the MDFO-EMTRR methodology has resulted in effectual outcomes. For example, on entire dataset, the MDFO-EMTRR model has provided average $accu_y$ of 99.74%, $prec_n$ of 98.20%, $reca_l$ of 98.35%, $F_{score}$ of 98.27%, and MCC of 98.13%. In addition, on 70% of TR data, the MDFO-EMTRR technique has offered average $accu_y$ of 99.74%, $prec_n$ of 98.20%, $reca_l$ of 98.35%, $F_{score}$ of 98.27%, and MCC of 98.13%. Moreover, on 30% of TS data, the MDFO-EMTRR model has rendered average $accu_y$ of 99.74%, $prec_n$ of 98.44%, $reca_l$ of 98.43%, $F_{score}$ of 98.43%, and MCC of 98.29%.

**Table 5:** Result analysis of MDFO-EMTRR approach with various measures under OSAC dataset

| Labels | Accuracy | Precision | Recall | F-Score | MCC |
|---|---|---|---|---|---|
| | | Entire dataset | | | |
| C-1 | 99.71 | 99.22 | 98.65 | 98.93 | 98.76 |
| C-2 | 99.70 | 98.77 | 99.09 | 98.93 | 98.75 |
| C-3 | 99.68 | 99.03 | 98.97 | 99.00 | 98.81 |
| C-4 | 99.69 | 98.89 | 98.95 | 98.92 | 98.74 |
| C-5 | 99.76 | 98.76 | 99.05 | 98.91 | 98.77 |
| C-6 | 99.72 | 98.65 | 98.61 | 98.63 | 98.47 |
| C-7 | 99.81 | 96.73 | 95.51 | 96.12 | 96.02 |
| C-8 | 99.77 | 97.15 | 97.35 | 97.25 | 97.13 |
| C-9 | 99.81 | 95.47 | 98.76 | 97.09 | 97.00 |
| C-10 | 99.78 | 99.32 | 98.57 | 98.94 | 98.82 |
| Average | 99.74 | 98.20 | 98.35 | 98.27 | 98.13 |
| | | Entire dataset | | | |
| C-1 | 99.71 | 99.22 | 98.65 | 98.93 | 98.76 |
| C-2 | 99.70 | 98.77 | 99.09 | 98.93 | 98.75 |
| C-3 | 99.68 | 99.03 | 98.97 | 99.00 | 98.81 |
| C-4 | 99.69 | 98.89 | 98.95 | 98.92 | 98.74 |
| C-5 | 99.76 | 98.76 | 99.05 | 98.91 | 98.77 |
| C-6 | 99.72 | 98.65 | 98.61 | 98.63 | 98.47 |
| C-7 | 99.81 | 96.73 | 95.51 | 96.12 | 96.02 |
| C-8 | 99.77 | 97.15 | 97.35 | 97.25 | 97.13 |
| C-9 | 99.81 | 95.47 | 98.76 | 97.09 | 97.00 |
| C-10 | 99.78 | 99.32 | 98.57 | 98.94 | 98.82 |
| Average | 99.74 | 98.20 | 98.35 | 98.27 | 98.13 |
| | | Training phase (70%) | | | |
| C-1 | 99.71 | 99.26 | 98.62 | 98.94 | 98.77 |
| C-2 | 99.72 | 98.80 | 99.24 | 99.02 | 98.86 |
| C-3 | 99.67 | 99.05 | 98.89 | 98.97 | 98.77 |
| C-4 | 99.69 | 98.87 | 99.00 | 98.93 | 98.75 |
| C-5 | 99.76 | 98.67 | 99.03 | 98.85 | 98.71 |
| C-6 | 99.71 | 98.88 | 98.27 | 98.58 | 98.41 |
| C-7 | 99.77 | 96.14 | 94.68 | 95.41 | 95.29 |
| C-8 | 99.80 | 97.40 | 97.70 | 97.55 | 97.45 |

(Continued)

**Table 5 (continued)**

| Labels | Accuracy | Precision | Recall | F-Score | MCC |
|---|---|---|---|---|---|
| C-9 | 99.80 | 94.43 | 99.16 | 96.74 | 96.67 |
| C-10 | 99.79 | 99.33 | 98.68 | 99.01 | 98.89 |
| Average | 99.74 | 98.08 | 98.33 | 98.20 | 98.06 |
| Testing phase (30%) | | | | | |
| C-1 | 99.70 | 99.13 | 98.70 | 98.92 | 98.74 |
| C-2 | 99.64 | 98.72 | 98.72 | 98.72 | 98.51 |
| C-3 | 99.70 | 98.98 | 99.16 | 99.07 | 98.90 |
| C-4 | 99.69 | 98.93 | 98.83 | 98.88 | 98.70 |
| C-5 | 99.78 | 98.96 | 99.09 | 99.03 | 98.90 |
| C-6 | 99.75 | 98.10 | 99.41 | 98.75 | 98.61 |
| C-7 | 99.90 | 98.14 | 97.53 | 97.83 | 97.78 |
| C-8 | 99.70 | 96.56 | 96.56 | 96.56 | 96.41 |
| C-9 | 99.84 | 97.58 | 97.98 | 97.78 | 97.69 |
| C-10 | 99.75 | 99.29 | 98.31 | 98.80 | 98.66 |
| Average | 99.74 | 98.44 | 98.43 | 98.43 | 98.29 |

A comparative review of the results provided by the MDFO-EMTRR method on OSAC dataset is given in Table 6 [14]. The results identified that the KELM model has shown lower $accu_y$ of 95.84% whereas the support vector machine (SVM) method has resulted in slightly increased $accu_y$ of 96.33%. Further, the ELM, decision tree (DT), and ArCAR models have demonstrated reasonable $accu_y$ of 96.21%, 96.01%, and 98.26% correspondingly. But, the MDFO-EMTRR method has shown maximum $accu_y$ of 99.74%.

**Table 6:** Comparative analysis of MDFO-EMTRR approach with existing algorithms under OSAC dataset

| Methods | Accuracy (%) |
|---|---|
| MDFO-EMTRR | 99.74 |
| SVM model | 96.33 |
| ELM model | 96.21 |
| Decision tree classifier | 96.01 |
| KELM model | 95.84 |
| ArCAR model | 98.26 |

## 5 Conclusion

In this study, a novel MDFO-EMTRR approach was established for the recognition and classification of Arabic text. To achieve this, the MDFO-EMTRR technique encompasses data pre-processing to transform the input data into compatible format. Next, the ELM model is utilized for the representation and recognition of the Arabic text. At last, the MDFO algorithm is exploited for optimal tuning of the parameters connected to the ELM method and thereby accomplishing enhanced classifier results. The experimental outcome study of the MDFO-EMTRR approach was performed on benchmark dataset and the outcomes described the supremacy of the MDFO-EMTRR technique over recent approaches. Thus, the MDFO-EMTRR system was regarded as an effectual tool for Arabic text classification. In upcoming years, the efficacy of the MDFO-EMTRR method was enhanced by the inclusion of fusion models.

**Funding Statement:** Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2022R263), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia. The authors would like to thank the Deanship of Scientific Research at Umm Al-Qura University for supporting this work by Grant Code: 22UQU4340237DSR35.

**Conflicts of Interest:** The authors declare that they have no conflicts of interest to report regarding the present study.

## References

[1] I. S. A. Sheikh, M. Mohd and L. Warlina, "A review of Arabic text recognition dataset," *Asia-Pacific Journal of Information Technology and Multimedia*, vol. 9, no. 1, pp. 69–81, 2020.

[2] L. Oulladji, K. Feraoun, M. Batouche and A. Abraham, "Arabic text detection using ensemble machine learning," *International Journal of Hybrid Intelligent Systems*, vol. 14, no. 4, pp. 233–238, 2018.

[3] M. A. Ahmed, R. A. Hasan, A. H. Ali and M. A. Mohammed, "The classification of the modern Arabic poetry using machine learning," *Telkomnika*, vol. 17, no. 5, pp. 2667, 2019.

[4] F. N. Al-Wesabi, "Entropy-based watermarking approach for sensitive tamper detection of Arabic text," *Computers, Materials & Continua*, vol. 67, no. 3, pp. 3635–3648, 2021.

[5] O. Zayene, S. M. Touj, J. Hennebert, R. Ingold and N. E. B. Amara, "Open datasets and tools for Arabic text detection and recognition in news video frames," *Journal of Imaging*, vol. 4, no. 2, pp. 32, 2018.

[6] F. N. Al-Wesabi, A. Abdelmaboud, A. A. Zain, M. M. Almazah and A. Zahary, "Tampering detection approach of Arabic-text based on contents interrelationship," *Intelligent Automation & Soft Computing*, vol. 27, no. 2, pp. 483–498, 2021.

[7] F. R. Alharbi and M. B. Khan, "Identifying comparative opinions in Arabic text in social media using machine learning techniques," *SN Applied Sciences*, vol. 1, no. 3, pp. 213, 2019.

[8] F. N. Al-Wesabi, "Proposing high-smart approach for content authentication and tampering detection of Arabic text transmitted via internet," *IEICE Transactions on Information and Systems*, vol. E103.D, no. 10, pp. 2104–2112, 2020.

[9] F. N. Al-Wesabi, "A hybrid intelligent approach for content authentication and tampering detection of Arabic text transmitted via internet," *Computers, Materials & Continua*, vol. 66, no. 1, pp. 195–211, 2021.

[10] F. N. Al-Wesabi, "A smart english text zero-watermarking approach based on third-level order and word mechanism of markov model," *Computers, Materials & Continua*, vol. 65, no. 2, pp. 1137–1156, 2020.

[11] S. A. AlAjlan and A. K. J. Saudagar, "Machine learning approach for threat detection on social media posts containing Arabic text," *Evolutionary Intelligence*, vol. 14, no. 2, pp. 811–822, 2021.

[12] S. Boukil, M. Biniz, F. E. Adnani, L. Cherrat and A. E. E. Moutaouakkil, "Arabic text classification using deep learning technics," *International Journal of Grid and Distributed Computing*, vol. 11, no. 9, pp. 103–114, 2018.

[13] A. M. Azmi, M. N. Almutery and H. A. Aboalsamh, "Real-word errors in Arabic texts: A better algorithm for detection and correction," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 8, pp. 1308–1320, 2019.

[14] S. Ahmed, S. Naz, M. Razzak and R. Yusof, "Arabic cursive text recognition from natural scene images," *Applied Sciences*, vol. 9, no. 2, pp. 236, 2019.

[15] H. Elfaik and E. H. Nfaoui, "Deep bidirectional LSTM network learning-based sentiment analysis for Arabic text," *Journal of Intelligent Systems*, vol. 30, no. 1, pp. 395–412, 2020.

[16] H. Butt, M. R. Raza, M. J. Ramzan, M. J. Ali and M. Haris, "Attention-based CNN-RNN Arabic text recognition from natural scene images," *Forecasting*, vol. 3, no. 3, pp. 520–540, 2021.

[17] A. Y. Muaad, M. A. Al-antari, S. Lee and H. J. Davanagere, "A novel deep learning arcar system for Arabic text recognition with character-level representation," *Computer Sciences & Mathematics Forum*, vol. 2, no. 1, pp. 14, 2021.

[18] S. I. Saleem, A. M. Abdulazeez and Z. Orman, "A new segmentation framework for Arabic handwritten text using machine learning techniques," *Computers, Materials & Continua*, vol. 68, no. 2, pp. 2727–2754, 2021.

[19] A. Y. Muaad, H. Jayappa, M. A. Al-antari and S. Lee, "ArCAR: A novel deep learning computer-aided recognition for character-level Arabic text representation and recognition," *Algorithms*, vol. 14, no. 7, pp. 216, 2021.

[20] I. Ahmad and G. A. Fink, "Handwritten Arabic text recognition using multi-stage sub-core-shape HMMs," *International Journal on Document Analysis and Recognition*, vol. 22, no. 3, pp. 329–349, 2019.

[21] R. Elbarougy, G. Behery and A. El Khatib, "Extractive Arabic text summarization using modified pagerank algorithm," *Egyptian Informatics Journal*, vol. 21, no. 2, pp. 73–81, 2020.

[22] L. Xu, S. Ding, X. Xu and N. Zhang, "Self-adaptive extreme learning machine optimized by rough set theory and affinity propagation clustering," *Cognitive Computation*, vol. 8, no. 4, pp. 720–728, 2016.

[23] S. Mirjalili, "Dragonfly algorithm: A new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems," *Neural Computing and Applications*, vol. 27, no. 4, pp. 1053–1073, 2016.

[24] Y. Meraihi, A. R. Cherif, D. Acheli and M. Mahseur, "Dragonfly algorithm: A comprehensive review and applications," *Neural Computing and Applications*, vol. 32, no. 21, pp. 16625–16646, 2020.