Computers, Materials & Continua
DOI: 10.32604/cmc.2023.033564
Article

Tech Science Press

Automated Arabic Text Classification Using Hyperparameter Tuned Hybrid Deep Learning Model

Badriyya B. Al-onazi[1], Saud S. Alotaib[2], Saeed Masoud Alshahrani[3,*], Najm Alotaibi[4], Mrim M. Alnfiai[5], Ahmed S. Salama[6] and Manar Ahmed Hamza[7]

[1]Department of Language Preparation, Arabic Language Teaching Institute, Princess Nourah bint Abdulrahman University, P.O. Box 84428, Riyadh, 11671, Saudi Arabia
[2]Department of Information Systems, College of Computing and Information System, Umm Al-Qura University, Saudi Arabia
[3]Department of Computer Science, College of Computing and Information Technology, Shaqra University, Shaqra, Saudi Arabia
[4]Prince Saud AlFaisal Institute for Diplomatic Studies, Saudi Arabia
[5]Department of Information Technology, College of Computers and Information Technology, Taif University, Taif P.O. Box 11099, Taif, 21944, Saudi Arabia
[6]Department of Electrical Engineering, Faculty of Engineering & Technology, Future University in Egypt, New Cairo, 11845, Egypt
[7]Department of Computer and Self Development, Preparatory Year Deanship, Prince Sattam bin Abdulaziz University, AlKharj, Saudi Arabia
*Corresponding Author: Saeed Masoud Alshahrani. Email: salshahrani@su.edu.sa

Received: 20 June 2022; Accepted: 09 September 2022

Abstract: The text classification process has been extensively investigated in various languages, especially English. Text classification models are vital in several Natural Language Processing (NLP) applications. The Arabic language has a lot of significance. For instance, it is the fourth mostly-used language on the internet and the sixth official language of the United Nations. However, there are few studies on the text classification process in Arabic. A few text classification studies have been published earlier in the Arabic language. In general, researchers face two challenges in the Arabic text classification process: low accuracy and high dimensionality of the features. In this study, an Automated Arabic Text Classification using Hyperparameter Tuned Hybrid Deep Learning (AATC-HTHDL) model is proposed. The major goal of the proposed AATC-HTHDL method is to identify different class labels for the Arabic text. The first step in the proposed model is to pre-process the input data to transform it into a useful format. The Term Frequency-Inverse Document Frequency (TF-IDF) model is applied to extract the feature vectors. Next, the Convolutional Neural Network with Recurrent Neural Network (CRNN) model is utilized to classify the Arabic text. In the final stage, the Crow Search Algorithm (CSA) is applied to fine-tune the CRNN model's hyperparameters, showing the work's novelty. The proposed AATC-HTHDL model was experimentally validated under different parameters and the outcomes established the supremacy of the proposed AATC-HTHDL model over other approaches.

Keywords: Hybrid deep learning; natural language processing; arabic language; text classification; parameter tuning

1 Introduction

Arabic is one of the six prevalently-spoken global languages and its speakers are spread across the nations, especially in the Middle East region. Further, a certain number of Islamic nations to use Arabic as their main language since it is the language of the Qur'an [1,2]. The Arabic language has a total of 28 letters including elmodod and hamzas. Every note contains some notations that denote the position of a letter in a word, i.e., in the beginning, middle or at the end of a word [3,4]. There are no upper-or lower cases in the Arabic language, which is a peculiar characteristic compared to the other languages [5]. The geomorphology of the Arabic language is a complex phenomenon, while the complexity of the language is attributed to the fact that it contains radical letters, suffixes and prefixes.

Further, the letters in the Arabic language do not have consecutive procedures as in other languages. This is because the term assemblies of the letters are dissimilar and hinge on location in a sentence to convey a meaning [6,7]. As mentioned above, the categorization of the texts is extensive in the Arabic language. In contrast, various research works have been conducted earlier to categorize the texts in other languages, such as English language. However, the studies concerning the categorization of the texts in Arabic are limited [8]. Several methods and systems have been developed for classifying English texts which provide brilliant outcomes with high precision, thanks to the nature of the words, language and the letters in the English language [9].

Arabic Natural Language Processing (NLP) is a challenging process due to the challenging morphology and structure of the language. Therefore, the number of studies on Arabic NLP is less than the English NLP [10]. In general, several methods are utilized to improve the applications of NLP, like statistical methods, word embedding, graph representation and Deep Learning (DL) [11]. Text classification is a categorization or labelling procedure focusing on tagging the writing processes and distinguishing its kind or category [12]. Text is classified either as a single-label text or a multi-label text under which many difficult and different categories of labels exist. Several challenges exist in the classification of Arabic text and it cannot be handled using the methods generally used for English NLP. This is because Arabic is an amusing yet difficult language with delicate geomorphology [13]. Various studies indicate that the data is an important complexity in the Arabic text classification process. If a text contains additional structures, it increases the runtime too. So, the preferred solution to overcome this problem is to reduce the number of features. Therefore, feature selection techniques are proposed to select the appropriate features representing the language [14].

1.1 Prior Works on Arabic Text Classification

Boukil et al. [15] introduced a state-of-the-art technique for classifying Arabic texts. In this study, an Arabic stemming approach was introduced for the extraction, selection and reduction of the required characteristics. Next, the Term Frequency-Inverse Document Frequency (TF-IDF) approach was incorporated to provide weightage for the features. In the categorization stage, the DL algorithm was utilized, which is powerful in other fields such as pattern recognition and image processing. Though it is rarely used in text mining approaches, this study used Convolutional Neural Network. Aljedani et al. [16] examined a hierarchical multi-level categorization model with respect to Arabic. In this study, a Hierarchical Multi-level Arabic Text Categorization (HMATC) mechanism was developed based on Machine Learning (ML) method. Also, the study assessed the influence of the feature set dimensions and Feature Selection (FS) methods on the classification accuracy of the method. The Hierarchy of Multi-label categorization (HOMER) approach was improved by examining distinct sets of multi-label categories, clustering algorithms and dissimilar cluster counts to enhance the outcomes of the hierarchical classification.

Elnagar et al. [17] presented a novel unbiased, and rich dataset for single-label (SANAD) and multi-label (NADiA) Arabic text classification process. These datasets were made easily accessible to the researchers in Arabic computation linguistics. Furthermore, the study presented a wide range of DL analytical approaches for the Arabic text classification and estimated the model's efficacy. The proposed model had a distinctive feature compared to the existing models, i.e., no pre-processing stage was required in the proposed model, and it was completely based on the DL method. In literature [18], a better methodology was proposed for the Arabic text classification process in which the Chi-square FS method was applied to improve the accuracy of the classification process. In addition, the study compared the chi-square values with three other conventional FS metrics as Chi-square, mutual data and information gain. Sainte et al. [19] could not execute the Arabic text classification process comprehensively due to the difficulty of the Arabic language. A new firefly algorithm-related FS technique was developed in this study. This technique was effectively employed in dissimilar combinatorial challenges. But the model did not include the FS model to perform Arabic Text categorization.

1.2 Paper Contribution

The current study has developed an Automated Arabic Text Classification using Hyperparameter-Tuned Hybrid Deep Learning (AATC-HTHDL) model. The major goal of the developed AATC-HTHDL algorithm is to identify different class labels for the Arabic text. It includes distinct levels of operations. At first, the data is pre-processed to transform the input data into a useful format. Secondly, the TF-IDF model is applied to extract the feature vectors. Next, the Convolutional Neural Network with Recurrent Neural Network (CRNN) method is utilized for the classification of the Arabic text. Finally, the Crow Search Algorithm (CSA) is used for fine tuning the hyperparameters involved in the CRNN model. The results from the experimental analysis conducted upon the proposed AATC-HTHDL model were investigated under distinct aspects.

2 Materials and Methods

The proposed AATC-HTHDL model effectively identifies different class labels for the Arabic text. It encompasses different stages of operations. At first, data pre-processing is performed to transform the input data into a useful format. Secondly, the TF-IDF model is applied to extract the feature vectors. Next, the CRNN model is utilized for Arabic text classification. Finally, the CSA is used to fine-tune the hyperparameters involved in the CRNN model. Fig. 1 depicts the block diagram of the AATC-HTHDL approach.

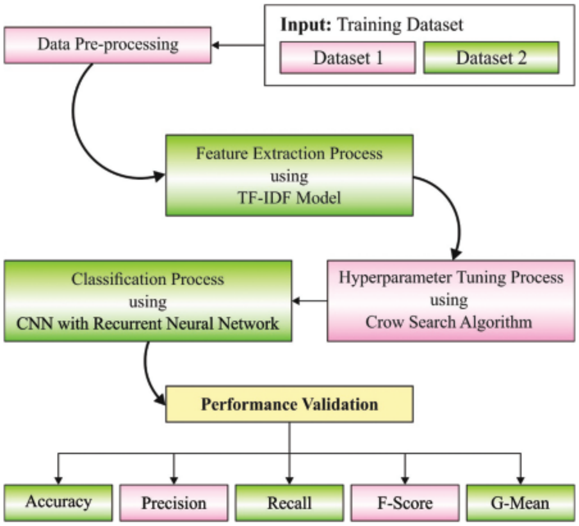


Figure 1: Block diagram of the AATC-HTHDL approach

2.1 Stage I: Data Pre-Processing

Primarily, the data is pre-processed to transform the input data into a useful format. In general, the tokenization procedure considers the dataset and divides it into distinct tokens. The tokens of single- or 2-characters length, numerical values and the non-Arabic characters are avoided from the dataset so that it does not affect the performance of the classifier [20]. Regular expressions help perform the tokenization process, whereas stop words are nothing but general functional words, for instance, prepositions, conjunctions and so on. Mostly, these words appear in a text with less impact on the classification procedure. A list of Arabic stop-words is collected, and such words are removed from the text. The NLP application developers eliminate such stop-words from the search engine indices so that the size of the indices gets extensively reduced. This phenomenon enhances recall and the precision results. Then, stemming, a mapping procedure is performed on the derivative words into a base format, i.e., stem that has been shared. Stemming employs the morphological heuristics to remove the affixes from the words before indexing it. For instance, the Arabic words "كتاب", "كاتب " and "مكتبة "share the same root "كتب , .".

2.2 Stage II: TF-IDF Model

In this stage, the TF-IDF model is applied to extract the feature vectors. Generally, the TF-IDF model operates by defining a similar frequency of the words in a document, in comparison with inverse ratio of the words over an entire corpus [21]. Indeed, this computation defines the significance of a word, provided in a specific document. A conventional word, in a few documents or a single document, is appropriate to have a high TF-IDF number than the conventional words like prepositions and articles. Assume a group of documents D and a word w, while deD denotes an individual document. Here, w_d i.e., the weight of the word w is calculated by employing the Eqs. (1) and (2).

$$w_d = TF^{*}IDF \tag{1}$$

$$w_d = f_{w,d*} log\left(|D| / f_{w,D}\right) \tag{2}$$

In this expression, $f_{w,d}$ indicates the number of times, the word w emerges in the d document, $|D|$ indicates the size of the corpus and $f_{w,D}$ indicates the document count in which w emerges in D.

Given that $|D| \sim f_{w,D}$, viz., the size of the corpus tentatively corresponds to the frequency of w.

$$1 < log\left(|D| / fw,\ D\right) < cte \tag{3}$$

The above equation is satisfied by a small constant like cte. Here, w_d remains small in comparison with $f_{w,d}$, which is still positive. This recommends that w is roughly communal over an entire corpus and simultaneously holds a certain set of consequences all over the element, D. For example, at some circumstances, the extreme elementary words like prepositions, articles and pronouns do not possess any applicable significance. Consequently, these words secure low TF-IDF scores that make it fundamentally unrelated during the search.

At last, visualize that $f_{w,d}$ is larger and $f_{w,D}$ is smaller. Here, the $log\left(|D| / f_{w,D}\right)$ is larger and w_d is smaller. These circumstances heavily create the interest, because the words with maximum w_d recommend that a w word is significant in the d document, but is not communal in the D corpus. This word w claims to be a significant discriminative power.

2.3 Stage III: CRNN Based Classification

For effectual classification of the Arabic text, the CRNN model is exploited in this study. The Convolutional Neural Network (CNN) model yields better performance in the development of features from spatial datasets like images. However, the CNN model cannot handle consecutive datasets. On the contrary, the Recurrent Neural Network (RNN)-based model possesses stronger abilities in terms of modelling the consecutive datasets like text [22]. Based on the simulation from features, the authors propose to combine RNN and CNN as a novel mechanism and term it as CRNN. When grouping RNN and CNN, the CNN method is placed in front of the RNN method which results in the creation of the CRNN method.

$y = F(X, \theta)$ of CRNN is denoted by:

$$y = Soft\max\ \left(FC\left(LSTM\left(Pool\left(Conv\left(X,\ \theta_{C0NV}\right), \theta_{Pool}\right),\ \theta_{LSTM}\right),\ \theta_{FC}\right)\right) \tag{4}$$

Here, the additional grouping is performed to place the RNN in front of the CNN and it results in the formation of the RCNN mechanism. $Y = F(X,\ \theta)$ of RCNN is denoted as follows.

$$y = Soft\max\ \left(FC\left(Pool\left(Conv\left(LSTM\left(X, \theta_{LSTM}\right), \theta_{C0NV}\right), \theta_{Pool}\right), \theta_{FC}\right)\right) \tag{5}$$

Here, the input dataset x denotes the short and sequential sentences. A novel and a better CNN-RNN mechanism is developed in this study. In the presented method, the CNN approach is placed in front of the RNN approach. A strong pooling layer moves towards the back of the Long Short Term Memory (LSTM)-RNN layer. In other terms, the LSTM layer processes the textual features that are attained by the convolutional layer and extracts the time-based features of the input dataset. Then, with the help of the maximal pooling layer, the text and its temporal characteristics are grouped, highlighted and compressed. The principle of the presented mechanism is that the problem is consecutive yet shorter. Hence, a better CRNN method places the pooling layer with prominent features behind the LSTM layer. It also highlights the textual features as well as the temporal features.

$y = F(X, \theta)$ of CRNN is denoted by:

$$y = Softmax\left(FC\left(Pool\left(LSTM\left(Conv\left(\mathrm{X},\ \theta_{C0NV}\right), \theta_{LSTM}\right), \theta_{Pool}\right), \theta_{FC}\right)\right) \tag{6}$$

2.4 Stage IV: CSA Based Parameter Tuning

In this final stage, the CSA method is used to fine-tune the hyperparameters of the CRNN method. CSA is a population-based stochastic searching technique [23]. The presented method is a recently-established optimization approach that is used to resolve complex engineering-based optimization problems. It is inspired from the smart performance of the crows. The principles of the CSA model are listed herewith.

a. Crow lives in a flock.

b. Crow remembers the location of the hiding position

c. Crow follows one another to commit thievery.

d. Crow protects its caches from getting stolen through possibility.

In line with the abovementioned assumptions, the central method of the CSA has three fundamental stages such as initialization, generation of a novel location and updating the crow's memory. In the beginning, an initialized population of the crows, characterized by n dimension, is arbitrarily produced. During the t iteration, the location of the crow is quantified as $\mathrm{x}^{\mathrm{i,t}} = \left[\mathrm{x}_1^{\mathrm{i,t}},\ \mathrm{x}_2^{\mathrm{i,t}}, \ldots\ldots, \mathrm{x}_{\mathrm{n}}^{\mathrm{i,t}}\right]$ and is considered that these crows possess better knowledge and memory, $\mathrm{m}^{\mathrm{i,t}} = \left[\mathrm{m}_{\mathrm{m1}}^{\mathrm{i,t}},\ \mathrm{m}_2^{\mathrm{iJt}},\ \cdots\cdots{}'\ \mathrm{m}_{\mathrm{n}}^{\mathrm{iJt}}\right]$. To produce a novel location, the ith crow arbitrarily selects a j-*th* crow from the population and tries to chase it so as to find the location of its hiding position (mj). In such cases, two states may occur based on a variable called Awareness Possibility (AP).

a. State 1: The crow j does not know that the crow i is chasing it i.e., i^{t}. Consequently, the crow i determines the hidden location of the crow j.

b. State 2: The crow j knows that it is being chased upon. Therefore, to defend the cache from being pilfered, the crow j fools the crow i by flying to other locations within the search space.

Based on the states mentioned above, the location of the crows is upgraded as follows.

$$\mathrm{x}^{\mathrm{i,iter+1=}}\begin{cases}\mathrm{x}^{\mathrm{i,iter+1}+r_i} \times \mathrm{fl}^{\mathrm{i,\ iter}} \times \left(\mathrm{m}^{i,\ \mathrm{iter}} - \mathrm{x}^{i,\mathrm{iter}}\right) \\ \mathrm{r_j} \geq \mathrm{AP}^{\mathrm{j,iter}}\text{A random positon of search space otherwise}\end{cases} \tag{7}$$

In Eq. (7), rj represents a uniform distribution fuzzy value within [0, 1] and $\mathrm{AP}^{j,\ \mathrm{iter}}$ signifies the awareness possibility of the crow j at *t* iteration. At last, the crow upgrades its memory with the help of the following expression.

$$\mathrm{m}^{\mathrm{i,iter+1=}}\begin{cases}\mathrm{x}^{\mathrm{i,iter+1}}, \text{if } \mathrm{f}\left(\mathrm{x}^{i,\ \mathrm{iter}}\right) \text{ is better than } \mathrm{f}\left(\mathrm{m}^{\mathrm{i,iter}}\right) \\ \\ \mathrm{m}^{\mathrm{i,iter}}, \text{otherwise}\end{cases} \tag{8}$$

In Eq. (8), f (−) represents the objective function. Once the fitness function of the novel location of a crow is superior to the fitness function of the existing location, the crow upgrades its memory about the novel location. The abovementioned procedure is reiterated until a satisfactory condition i.e., (iter-max) is achieved. At last, the optimal solution of the memory is attained as the best possible solution. Fig. 2 demonstrates the flowchart of the CSA method.

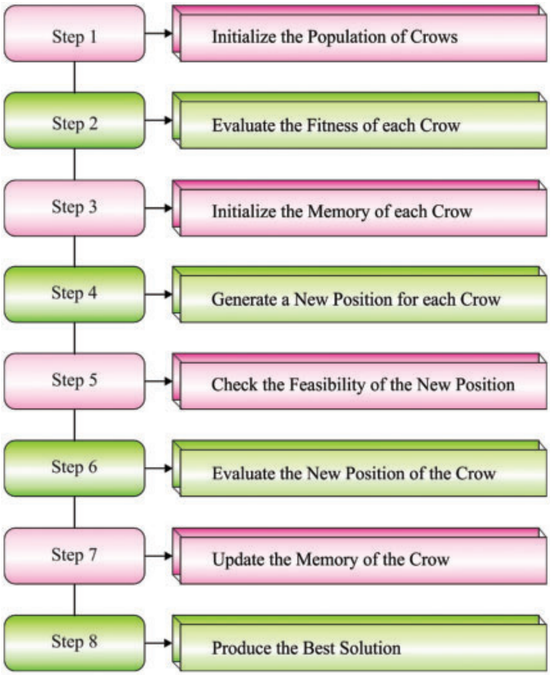

Step 1
Initialize the Population of Crows
Step 2
Evaluate the Fitness of each Crow
Step 3
Initialize the Memory of each Crow
Step 4
Generate a New Position for each Crow
Step 5
Check the Feasibility of the New Position
Step 6
Evaluate the New Position of the Crow
Step 7
Update the Memory of the Crow
Step 8
Produce the Best Solution

Figure 2: Flowchart of the CSA method

The CSA method derives a Fitness Function (FF) to achieve the enhanced classification outcomes. It uses a positive numeral to indicate the superior outcomes of the candidate solution. In this work, the reduced classification error rate is considered as the fitness function and is provided in Eq. (9). The finest solution contains the least error rate whereas the poor solution possesses the highest error rate.

$$fitness\,(x_i) == \frac{number\ of\ misclassified\ samples}{Total\ number\ of\ samples} * 100 \tag{9}$$

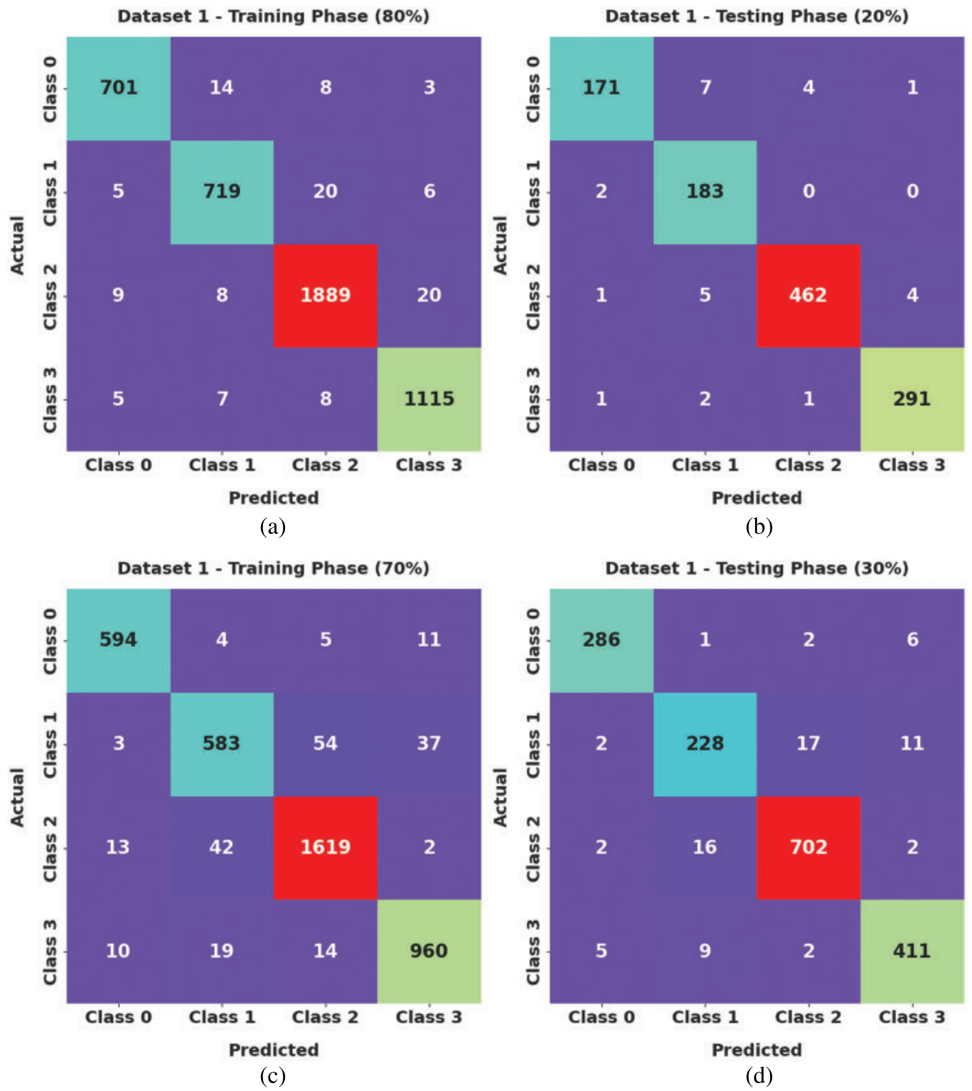
3 Experimental Validation

In this section, a detailed analysis was conducted upon the proposed AATC-HTHDL model using two datasets and the results are discussed herewith. The details related to the dataset-1 are shown in Table 1. The dataset holds 5,690 samples under four class labels.

Table 1: Details on dataset-1

Dataset-1		
Label	Category	No. of documents
Class 0	Economy	909
Class 1	International news	935
Class 2	Local news	2398
Class 3	Sport	1430
Total number of documents		5690

Fig. 3 depicts the confusion matrices formed by the proposed AATC-HTHDL model on the applied dataset-1. The figure reports that the proposed AATC-HTHDL model proficiently recognized the samples under four distinct class labels.



(a) (b) (c) (d)

Figure 3: Confusion matrices of the AATC-HTHDL approach under dataset-1 (a) 80% of TR data, (b) 20% of TS data, (c) 70% of TR data, and (d) 30% of TS data

Table 2 shows the detailed classification output achieved by the proposed AATC-HTHDL model on 80% of Training (TR) data and 20% of Testing (TS) datasets. The obtained values denote that the proposed AATC-HTHDL model produced an improved performance. With 80% of TR data, the presented AATC-HTHDL model achieved an average $accu_y$ of 98.75%, $prec_n$ of 97.27%, $reca_l$ of 97.18%, F_{score} of 97.23% and a G_{mean} of 98.15%. Also, with 20% of TS data, the proposed AATC-HTHDL method accomplished an average $accu_y$ of 98.77%, $prec_n$ of 96.69%, $reca_l$ of 97.22%, F_{score} of 97.06% and a G_{mean} of 98.19%.

Table 2: Analytical results of the AATC-HTHDL technique on dataset-1 under different measures

Dataset-1					
Labels	Accuracy	Precision	Recall	F-Score	G-Mean
Training phase (80%)					
Class 0	99.03	97.36	96.56	96.96	98.02
Class 1	98.68	96.12	95.87	95.99	97.54
Class 2	98.39	98.13	98.08	98.10	98.35
Class 3	98.92	97.47	98.24	97.85	98.69
Average	98.75	97.27	97.18	97.23	98.15
Testing phase (20%)					
Class 0	98.59	97.71	93.44	95.53	96.46
Class 1	98.59	92.89	98.92	95.81	98.72
Class 2	98.68	98.93	97.88	98.40	98.56
Class 3	99.21	98.31	98.64	98.48	99.02
Average	98.77	96.96	97.22	97.06	98.19

Table 3 shows the detailed classification outcomes achieved by the proposed AATC-HTHDL model on 70% of TR data and 30% of TS datasets. The obtained values indicate that the proposed AATC-HTHDL model achieved an improved performance. With 70% of TR data, the proposed AATC-HTHDL model provided an average $accu_y$ of 97.30%, $prec_n$ of 94.13%, $reca_l$ of 93.79%, F_{score} of 93.95% and a G_{mean} of 95.89%. Also, with 30% of TS data, the AATC-HTHDL technique offered an average $accu_y$ of 97.80%, $prec_n$ of 94.85%, $reca_l$ of 94.70%, F_{score} of 94.77% and a G_{mean} of 96.56%.

Table 3: Analytical results of the AATC-HTHDL methodology on dataset-1 under dissimilar measures

Dataset-1					
Labels	Accuracy	Precision	Recall	F-Score	G-Mean
Training phase (70%)					
Class 0	98.84	95.81	96.74	96.27	97.98
Class 1	95.99	89.97	86.12	88.00	91.88
Class 2	96.73	95.69	96.60	96.14	96.71
Class 3	97.66	95.05	95.71	95.38	97.01
Average	97.30	94.13	93.79	93.95	95.89
Testing phase (30%)					
Class 0	98.94	96.95	96.95	96.95	98.15
Class 1	96.71	89.76	88.37	89.06	93.16
Class 2	97.59	97.10	97.23	97.16	97.54
Class 3	97.94	95.58	96.25	95.92	97.37
Average	97.80	94.85	94.70	94.77	96.56

Both Training Accuracy (TA) and Validation Accuracy (VA) values, achieved by the proposed AATC-HTHDL methodology on dataset-1, are established in Fig. 4. The experimental results imply that the proposed AATC-HTHDL algorithm accomplished the highest TA and VA values whereas the VA values were higher than the TA values.

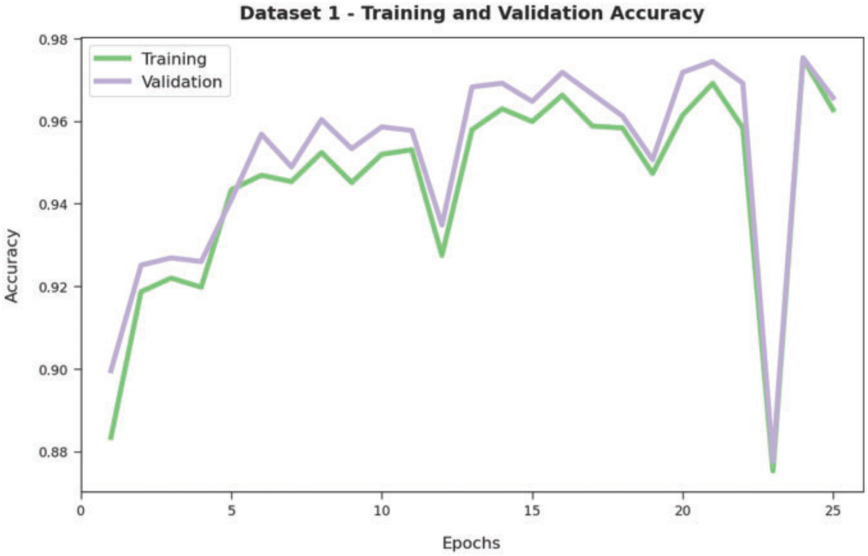


Figure 4: TA and VA analyses results of the AATC-HTHDL approach on dataset-1

Both Training Loss (TL) and Validation Loss (VL) values, obtained by the proposed AATC-HTHDL approach on dataset-1, are illustrated in Fig. 5. The experimental results infer that the proposed AATC-HTHDL technique achieved the minimal TL and VL values whereas the VL values were lesser compared to TL values.

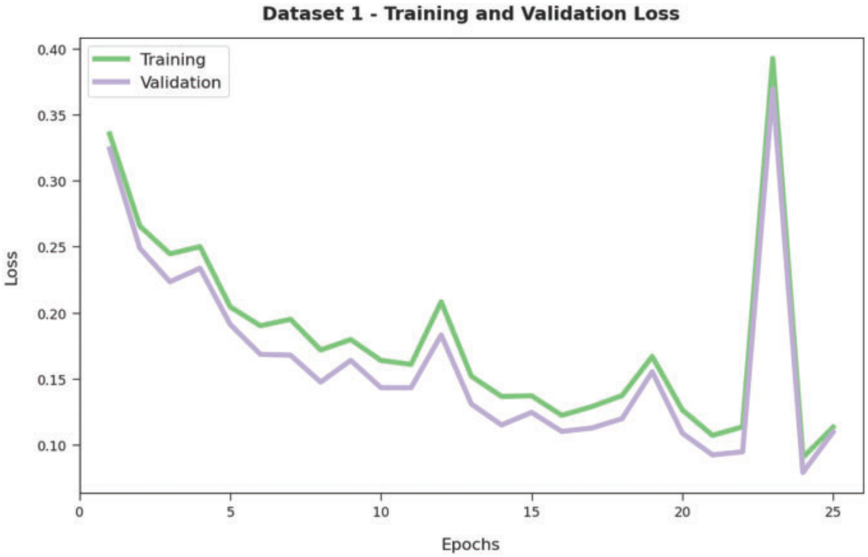


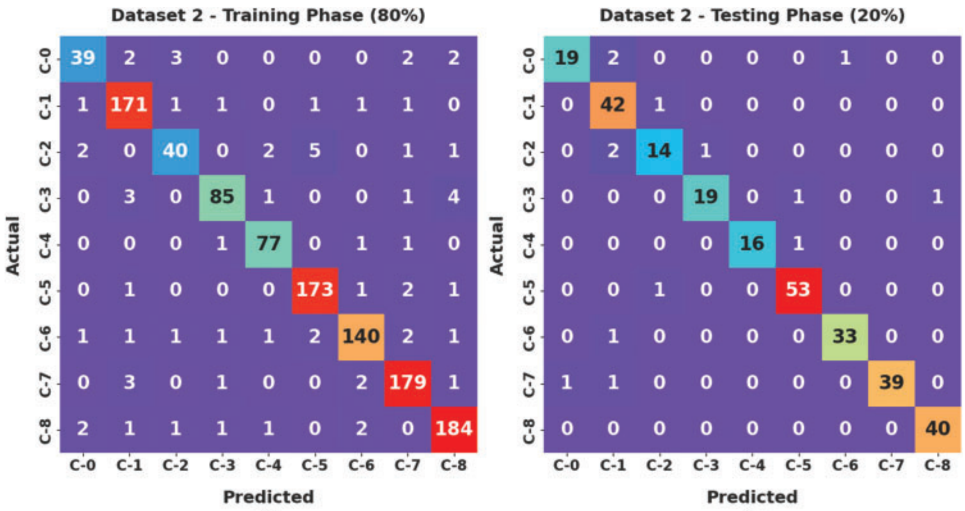
Figure 5: TL and VL analyses results of the AATC-HTHDL approach on dataset-1

The details related to the dataset-2 are given in Table 4. The dataset has 1,445 samples under eight class labels.

Table 4: Details on dataset-2

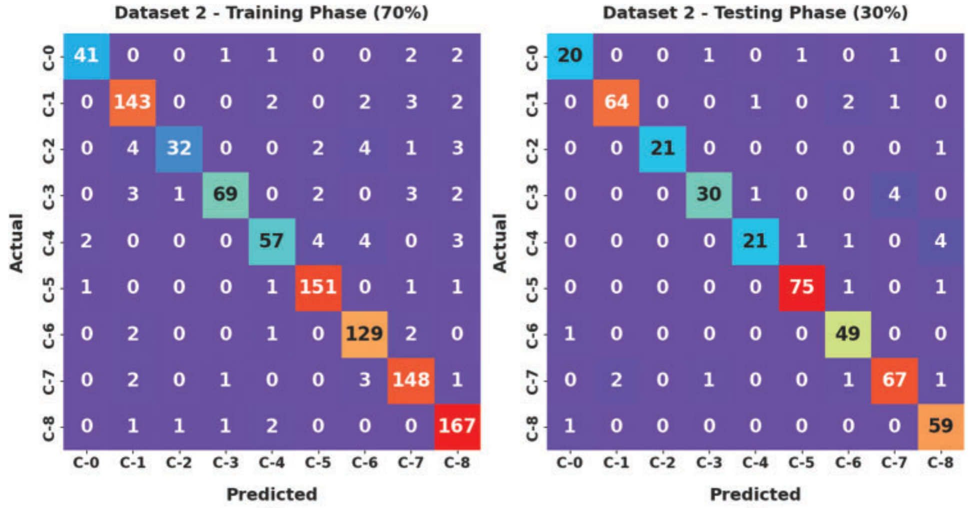
Dataset-2		
Label	Category	No. of documents
Class 0	Computer	70
Class 1	Economics	220
Class 2	Education	68
Class 3	Engineering	115
Class 4	Law	97
Class 5	Medicine	232
Class 6	Politics	184
Class 7	Religion	227
Class 8	Sport	232
Total number of documents		1445

Fig. 6 exhibits the confusion matrices generated by the proposed AATC-HTHDL approach on the applied dataset i.e., dataset-2. The obtained values denote that the proposed AATC-HTHDL technique proficiently recognized the samples under eight dissimilar class labels.



(a) (b)

Figure 6: (Continued)



(c) (d)

Figure 6: Confusion matrices of the AATC-HTHDL approach under dataset-2 (a) 80% of TR data, (b) 20% of TS data, (c) 70% of TR data, and (d) 30% of TS data

Table 5 depicts the detailed classification outcomes of the proposed AATC-HTHDL method on 80% of TR data and 20% of TS datasets. The figure reports that the presented AATC-HTHDL method achieved a better performance. With 80% of TR data, the AATC-HTHDL approach achieved an average $accu_y$ of 98.69%, $prec_n$ of 92.92%, $reca_l$ of 91.73%, F_{score} of 92.28% and a G_{mean} of 95.35%. Also, with 20% of TS dataset, the proposed AATC-HTHDL technique accomplished an average $accu_y$ of 98.92%, $prec_n$ of 95.11%, $reca_l$ of 93.48%, F_{score} of 94.21% and a G_{mean} of 96.34%.

Table 5: Analytical results of the AATC-HTHDL methodology on dataset-2 under different measures

Dataset-2					
Labels	Accuracy	Precision	Recall	F-Score	G-Mean
Training phase (80%)					
Class 0	98.70	86.67	81.25	83.87	89.89
Class 1	98.53	93.96	96.61	95.26	97.74
Class 2	98.53	86.96	78.43	82.47	88.32
Class 3	98.79	94.44	90.43	92.39	94.87
Class 4	99.31	93.90	96.25	95.06	97.88
Class 5	98.88	95.58	97.19	96.38	98.18
Class 6	98.53	95.24	93.33	94.28	96.27
Class 7	98.53	94.71	96.24	95.47	97.59
Class 8	98.44	94.85	95.83	95.34	97.39
Average	98.69	92.92	91.73	92.28	95.35

(Continued)

Table 5: Continued

Dataset-2					
Labels	Accuracy	Precision	Recall	F-Score	G-Mean
Testing phase (20%)					
Class 0	98.62	95.00	86.36	90.48	92.76
Class 1	97.58	87.50	97.67	92.31	97.62
Class 2	98.27	87.50	82.35	84.85	90.41
Class 3	98.96	95.00	90.48	92.68	94.94
Class 4	99.65	100.00	94.12	96.97	97.01
Class 5	98.96	96.36	98.15	97.25	98.65
Class 6	99.31	97.06	97.06	97.06	98.33
Class 7	99.31	100.00	95.12	97.50	97.53
Class 8	99.65	97.56	100.00	98.77	99.80
Average	98.92	95.11	93.48	94.21	96.34

Table 6 shows the detailed classification outcomes of the proposed AATC-HTHDL method on 70% of TR dataset and 30% of TS dataset. The values obtained denote that the proposed AATC-HTHDL approach achieved a better performance. With 70% of TR data, the proposed AATC-HTHDL technique accomplished an average $accu_y$ of 98.37%, $prec_n$ of 92.78%, $reca_l$ of 89.42%, F_{score} of 90.82% and a G_{mean} of 93.99%. Also, with 30% of TS data, the proposed AATC-HTHDL method yielded an average $accu_y$ of 98.57%, $prec_n$ of 93.58%, $reca_l$ of 91.87%, F_{score} of 92.59% and a G_{mean} of 95.38%.

Table 6: Analytical results of the AATC-HTHDL technique on dataset-2 under different measures

Dataset-2					
Labels	Accuracy	Precision	Recall	F-Score	G-Mean
Training phase (70%)					
Class 0	99.11	93.18	87.23	90.11	93.25
Class 1	97.92	92.26	94.08	93.16	96.31
Class 2	98.42	94.12	69.57	80.00	83.32
Class 3	98.62	95.83	86.25	90.79	92.72
Class 4	98.02	89.06	81.43	85.07	89.90
Class 5	98.81	94.97	97.42	96.18	98.24
Class 6	98.22	90.85	96.27	93.48	97.39
Class 7	98.12	92.50	95.48	93.97	97.03
Class 8	98.12	92.27	97.09	94.62	97.71
Average	98.37	92.78	89.42	90.82	93.99

(Continued)

Table 6: Continued

Dataset-2					
Labels	Accuracy	Precision	Recall	F-Score	G-Mean
Testing phase (30%)					
Class 0	98.85	90.91	86.96	88.89	93.02
Class 1	98.62	96.97	94.12	95.52	96.75
Class 2	99.77	100.00	95.45	97.67	97.70
Class 3	98.39	93.75	85.71	89.55	92.35
Class 4	98.16	91.30	77.78	84.00	87.97
Class 5	99.08	97.40	97.40	97.40	98.42
Class 6	98.62	90.74	98.00	94.23	98.35
Class 7	97.47	91.78	93.06	92.41	95.66
Class 8	98.16	89.39	98.33	93.65	98.23
Average	98.57	93.58	91.87	92.59	95.38

Both TA and VA values, attained by the proposed AATC-HTHDL approach on dataset-2, are established in Fig. 7. The experimental results imply that the proposed AATC-HTHDL algorithm accomplished the highest TA and VA values whereas the VA values were higher than the TA values.

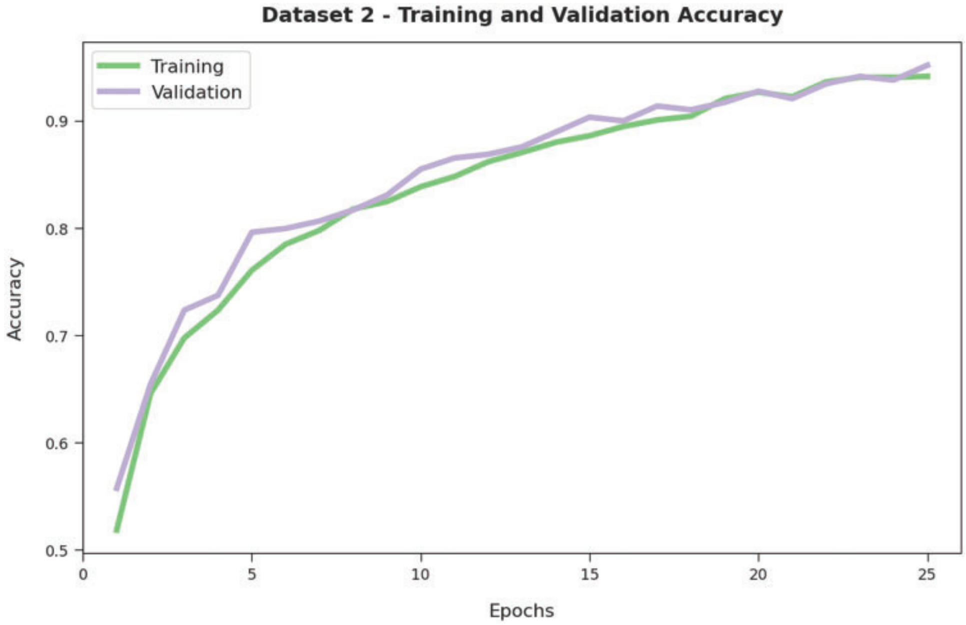


Figure 7: TA and VA analyses results of the AATC-HTHDL method on dataset-2

Both TL and VL values, achieved by the proposed AATC-HTHDL approach on dataset-2, are illustrated in Fig. 8. The experimental results infer that the proposed AATC-HTHDL technique achieved the minimal TL and VL values while the VL values were lesser than the TL values.

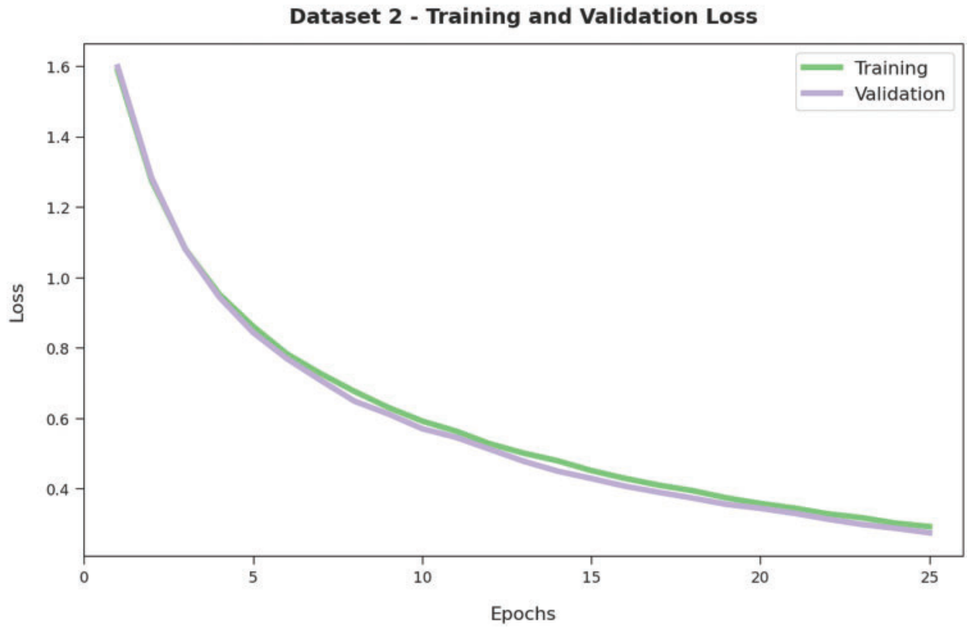


Figure 8: TL and VL analyses results of the AATC-HTHDL approach on dataset-2

In order to validate the improved performance of the proposed AATC-HTHDL model, a comparative analysis was conducted between the proposed model and the rest of the existing models such as Logistic Regression (LR), Support Vector Machine (SVM), Generative Adversarial Network (GAN), Artificial Neural Network (ANN) and Ant Colony Optimization (ACO) on dataset-1 and the results are portrayed in Table 7 and Fig. 9 [24]. The results demonstrate that the proposed AATC-HTHDL model achieved an excellent performance. For instance, in terms of $accu_y$, the proposed AATC-HTHDL model achieved a maximum $accu_y$ of 98.77%, whereas the LR, SVM, GAN, ANN and the ACO models accomplished the least $accu_y$ values such as 97.53%, 94.29%, 96.22%, 92.26% and 91.79% respectively. Next, in terms of F_{score}, the proposed AATC-HTHDL method yielded a maximum F_{score} of 97.06%, whereas the LR, SVM, GAN, ANN and the ACO approaches provided the least F_{score} values such as 90.58%, 92.55%, 96.17%, 92.36% and 95.74% correspondingly.

Table 7: Comparative analysis results of the AATC-HTHDL approach and other recent methodologies under dataset-1

Dataset-1		
Methods	Accuracy	F-Score
AATC-HTHDL	98.77	97.06
LR model	97.53	90.58
SVM model	94.29	92.55
GAN model	96.22	96.17
ANN model	92.26	92.36
ACO model	91.79	95.74

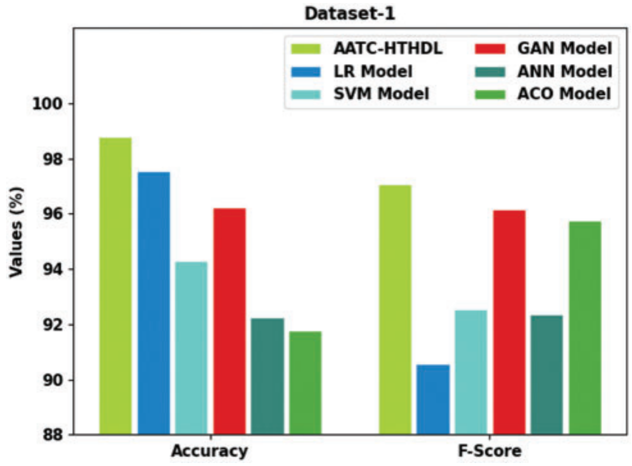


Figure 9: Comparative analysis results of the AATC-HTHDL approach upon dataset-1

In order to validate the enhanced outcomes of the proposed AATC-HTHDL approach, a comparative analysis was conducted between the proposed model and the existing models on dataset-2 and the results are demonstrated in Table 8 and Fig. 10. The results illustrate that the presented AATC-HTHDL approach achieved an excellent performance. For instance, in terms of $accu_y$, the AATC-HTHDL system achieved a maximum $accu_y$ of 98.92%, while the LR, SVM, GAN, ANN and the ACO techniques scored the least $accu_y$ values such as 96.46%, 94.99%, 97.62%, 95.30% and 94.78% correspondingly. Next, in terms of F_{score}, the presented AATC-HTHDL methodology reported a maximum F_{score} of 94.21%, whereas other models such as LR, SVM, GAN, ANN and ACO accomplished less F_{score} values such as 89.37%, 91.33%, 92.86%, 92.05% and 93.26% correspondingly.

Table 8: Comparative analysis results of the AATC-HTHDL approach and other recent methodologies under dataset-1

Dataset-2		
Methods	Accuracy	F-Score
AATC-HTHDL	98.92	94.21
LR model	96.46	89.37
SVM model	94.99	91.33
GAN model	97.62	92.86
ANN model	95.30	92.05
ACO model	94.78	93.26

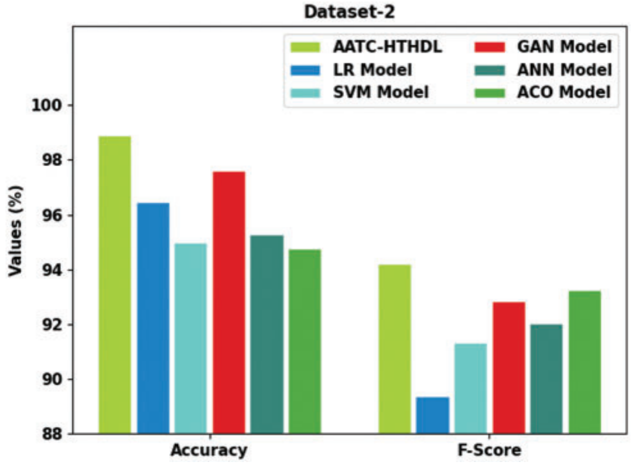


Figure 10: Comparative analysis results of the AATC-HTHDL approach on dataset-2

Based on the analytical results and the discussions made above, it can be inferred that the proposed AATC-HTHDL model achieved a better performance compared to other models.

4 Conclusion

In the current study, the AATC-HTHDL model has been developed to identify different class labels for the Arabic text. It encompasses different stages of operations. At first, the data is pre-processed to transform the input data into a useful format. Secondly, the TF-IDF model is applied to extract the feature vectors. Next, the CRNN model is utilized for the Arabic text classification process. Finally, the CSA is used for fine-tuning the hyperparameters involved in the CRNN model. The proposed AATC-HTHDL model was experimentally validated and the outcomes were investigated under distinct aspects. The comparison study outcomes highlight the promising performance of the presented AATC-HTHDL model over other approaches. In the future, feature selection and feature reduction methodologies can be applied to improve the classification performance of the projected technique.

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: (22UQU4210118DSR31). The author would like to thank the Deanship of Scientific Research at Shaqra University for supporting this work.

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

References

[1] A. M. F. A. Sbou, "A survey of arabic text classification models," *International Journal of Electrical and Computer Engineering*, vol. 8, no. 6, pp. 4352, 2018.

[2] A. Wahdan, S. A. L. Hantoobi, S. A. Salloum and K. Shaalan, "A systematic review of text classification research based on deep learning models in arabic language," *International Journal of Electrical and Computer Engineering*, vol. 10, no. 6, pp. 6629, 2020.

[3] 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.

[4] H. Chantar, M. Mafarja, H. Alsawalqah, A. A. Heidari, I. Aljarah *et al.,* "Feature selection using binary grey wolf optimizer with elite-based crossover for arabic text classification," *Neural Computing and Applications*, vol. 32, no. 16, pp. 12201–12220, 2020.

[5] 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.

[6] A. E. Kah and I. Zeroual, "The effects of pre-processing techniques on arabic text classification," *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 10, no. 1, pp. 41–48, 2021.

[7] 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.

[8] 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.

[9] 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.

[10] F. S. Al-Anzi and D. AbuZeina, "Beyond vector space model for hierarchical arabic text classification: A markov chain approach," *Information Processing & Management*, vol. 54, no. 1, pp. 105–115, 2018.

[11] X. Luo, "Efficient English text classification using selected machine learning techniques," *Alexandria Engineering Journal*, vol. 60, no. 3, pp. 3401–3409, 2021.

[12] A. Adel, N. Omar, M. Albared and A. Al-Shabi, "Feature selection method based on statistics of compound words for arabic text classification," *The International Arab Journal of Information Technology*, vol. 16, no. 2, pp. 178–185, 2019.

[13] H. N. Alshaer, M. A. Otair, L. Abualigah, M. Alshinwan and A. M. Khasawneh, "Feature selection method using improved CHI square on arabic text classifiers: Analysis and application," *Multimedia Tools and Applications*, vol. 80, no. 7, pp. 10373–10390, 2021.

[14] R. Janani and S. Vijayarani, "Automatic text classification using machine learning and optimization algorithms," *Soft Computing*, vol. 25, no. 2, pp. 1129–1145, 2020.

[15] 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.

[16] N. Aljedani, R. Alotaibi and M. Taileb, "HMATC: Hierarchical multi-label arabic text classification model using machine learning," *Egyptian Informatics Journal*, vol. 22, no. 3, pp. 225–237, 2021.

[17] A. Elnagar, R. A. Debsi and O. Einea, "Arabic text classification using deep learning models," *Information Processing & Management*, vol. 57, no. 1, pp. 102121, 2020.

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

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

[20] E. Al-Thwaib, B. H. Hammo and S. Yagi, "An academic arabic corpus for plagiarism detection: Design, construction and experimentation," *International Journal of Educational Technology in Higher Education*, vol. 17, no. 1, pp. 1, 2020.

[21] S. Douzi, F. AlShahwan, M. Lemoudden and B. Ouahidi, "Hybrid email spam detection model using artificial intelligence," *International Journal of Machine Learning and Computing*, vol. 10, no. 2, pp. 316–322, 2020.

[22] S. Yusuf, A. Alshdadi, M. Alassafi, R. AlGhamdi and A. Samad, "Predicting catastrophic temperature changes based on past events via a CNN-LSTM regression mechanism," *Neural Computing and Applications*, vol. 33, no. 15, pp. 9775–9790, 2021.

[23] T. Gadekallu, M. Alazab, R. Kaluri, P. K. R. Maddikunta, S. Bhattacharya *et al.,* "Hand gesture classification using a novel CNN-crow search algorithm," *Complex & Intelligent Systems*, vol. 7, pp. 1855–1868, 2021.

[24] K. Sundus, F. Al-Haj and B. Hammo, "A deep learning approach for arabic text classification," in *2019 2nd Int. Conf. on new Trends in Computing Sciences (ICTCS)*, Amman, Jordan, pp. 1–7, 2019.