Computer Systems
Science and Engineering

Tech Science Press

DOI: 10.32604/csse.2023.033901
Article

Improved Ant Lion Optimizer with Deep Learning Driven Arabic Hate Speech Detection

Abdelwahed Motwakel[1,*], Badriyya B. Al-onazi[2], Jaber S. Alzahrani[3], Sana Alazwari[4], Mahmoud Othman[5], Abu Sarwar Zamani[1], Ishfaq Yaseen[1] and Amgad Atta Abdelmageed[1]

[1]Department of Computer and Self Development, Preparatory Year Deanship, Prince Sattam bin Abdulaziz University, AlKharj, Saudi Arabia
[2]Department of Language Preparation, Arabic Language Teaching Institute, Princess Nourah bint Abdulrahman University, P.O. Box 84428, Riyadh, 11671, Saudi Arabia
[3]Department of Industrial Engineering, College of Engineering at Alqunfudah, Umm Al-Qura University, Saudi Arabia
[4]Department of Information Technology, College of Computers and Information Technology, Taif University, Taif, P.O. Box 11099, Taif, 21944, Saudi Arabia
[5]Department of Computer Science, Faculty of Computers and Information Technology, Future University in Egypt, New Cairo, 11835, Egypt
*Corresponding Author: Abdelwahed Motwakel. Email: a.ismaeil@psau.edu.sa
Received: 01 July 2022; Accepted: 13 November 2022

Abstract: Arabic is the world's first language, categorized by its rich and complicated grammatical formats. Furthermore, the Arabic morphology can be perplexing because nearly 10,000 roots and 900 patterns were the basis for verbs and nouns. The Arabic language consists of distinct variations utilized in a community and particular situations. Social media sites are a medium for expressing opinions and social phenomena like racism, hatred, offensive language, and all kinds of verbal violence. Such conduct does not impact particular nations, communities, or groups only, extending beyond such areas into people's everyday lives. This study introduces an Improved Ant Lion Optimizer with Deep Learning Dirven Offensive and Hate Speech Detection (IALODL-OHSD) on Arabic Cross-Corpora. The presented IALODL-OHSD model mainly aims to detect and classify offensive/hate speech expressed on social media. In the IALODL-OHSD model, a three-stage process is performed, namely pre-processing, word embedding, and classification. Primarily, data pre-processing is performed to transform the Arabic social media text into a useful format. In addition, the word2vec word embedding process is utilized to produce word embeddings. The attention-based cascaded long short-term memory (ACLSTM) model is utilized for the classification process. Finally, the IALO algorithm is exploited as a hyperparameter optimizer to boost classifier results. To illustrate a brief result analysis of the IALODL-OHSD model, a detailed set of simulations were performed. The extensive comparison study portrayed the enhanced performance of the IALODL-OHSD model over other approaches.

Keywords: Hate speech; offensive speech; Arabic corpora; natural language processing; social networks

1 Introduction

Despite the prominence of the Arabic language, certain corpus-related semantic relation extraction researchers made attention towards the Arabic language. The reason behind this is the limited sources serving the linguistic and the shortage of well-annotated corpora [1]. Both the lack of tools and language constitute it hard to build an Arabic lexical corpus. The language becomes complicated in 3 aspects one is semantics, another one is morphology, and finally syntax [2]. The Arabic language contains many grammar rules that give surge to difficulties in creating the language in an official structure.

Moreover, the absence of diacritics in the written text constitutes ambiguity. Further, automatically differentiating between abbreviations, proper names, and acronyms becomes tough because capitalization was not utilized in Arabic [3]. The most difficult task in natural language processing (NLP) was deriving semantic relationships. The task teaches discovering instances of predefined relationships among entity pairs. Determining the semantic relationship between 2 words will greatly enhance the accuracy of NLP applications [4]. NLP applications, which were affected by semantic relationship, involves discourse processing, word sense disambiguation, and sentiment analysis. But existing Arabic lexical sources were inadequate for Arabic language process tasks because of their limited coverage [5,6]. For example, Arabic WordNet (AWN) will cover just general ideas, and wants are protracted for encompassing more explicit fields. Traditional manual semantic relationship extraction is time-consuming, labor-intensive, and expensive [7]. Few researchers have argued that an automated technique is useful in deriving semantic relationships and educational lexical sources, but automated techniques do not indulge in direct procedures [8].

The utility of similarity distribution and corpus statistical approaches was valuable in extracting the semantic relationship among pairs of words [9]. Detection of hate speech (HS) was a difficult task because there is the absence of a common understanding of the actual meaning of HS, and the absence of high-quality annotated data sets, particularly for other languages except for English [10,11]. Certain works on HS and offensive language detection (OFF) tasks, including the Arabic language. Many works allot labels to a provided input; the labels differ because of the absence of a universal explanation of offensive and HS [12]. Questionably, all HS, cyberbullying, toxic comments and aggressive subjects make different types of offensive and hate content absent or present in diverse corpora [13]. Moreover, considering all classifier tasks separately takes more resources [14].

Mossie et al. [15] devise an HS detection method for identifying hatred towards vulnerable minority groups in mass media. Initially, posts were mechanically gathered and pre-processed in Spark's distributed processing structure, and features were derived using word n-grams and word-embedded methods like Word2Vec. Then, deep learning (DL) methods for classifying a variety of recurrent neural networks (RNN), Gated Recurrent Unit (GRU), have been utilized for HS detection. Lastly, hate words were clustered with approaches like Word2Vec for predicting the possible target ethnic group for hatred. Aljarah et al. [16] aim to identify cyber-HS based on the Arabic context over the Twitter platform by implementing NLP and machine learning (ML) techniques. This study takes a set of tweets based on terrorism, racism, Islam, sports orientation, and journalism. Numerous feature types and feelings were derived and organized in 15 distinct data combinations. In [17], created a method where, taking profits of neural network (NN), classifies tweets written in 7 distinct languages (and also those above one language at once) to hate speech (HS) or non-HS. It utilized a convolutional neural network (CNN) and character-level representation.

Khalafat et al. [18] introduce the design and application for violence detection on mass media utilizing ML techniques. This system operates in the Jordanian Arabic dialect rather than Modern Standard Arabic (MSA). The data was gathered from two popular mass media websites (Twitter and Facebook) and utilized native speakers for annotating the data. Additionally, distinct pre-processing methods were utilized to show the effect on model accuracy. The Arabic lexicon can be employed to generate feature vectors and distinguish them into feature sets. In [19], it creates the first public Arabic data set of tweets annotated for religious HS identification. And also created 3 public Arabic lexicons of terms based on religion together with hate scores. Afterwards, present detailed scrutiny of the labelled dataset, reporting the most targeted religious groups and non-hateful and hateful tweets. The labelled dataset was utilized for training 7 classifier methods utilizing DL-based, lexicon-based, and n-gram-based approaches.

This study introduces an Improved Ant Lion Optimizer with Deep Learning Dirven Offensive and Hate Speech Detection (IALODL-OHSD) on Arabic Cross-Corpora. In the IALODL-OHSD model, a three-stage process is performed, namely pre-processing, word embedding, and classification. Primarily, data pre-processing is performed to transform the Arabic social media text into a useful format. In addition, the word2vec word embedding process is utilized to produce word embeddings. An attention-based cascaded long short-term memory (ACLSTM) model is utilized for the classification process. Finally, the IALO algorithm is exploited as a hyperparameter optimizer to boost classifier results. To illustrate a brief result analysis of the IALODL-OHSD model, a detailed set of simulations were performed.

2 The Proposed Model

This study devised a new IALODL-OHSD technique to detect and classify offensive/hate speech expressed on social media. In the IALODL-OHSD model, a three-stage process is performed, namely pre-processing, word embedding, and classification. Fig. 1 depicts the block diagram of the IALODL-OHSD approach.

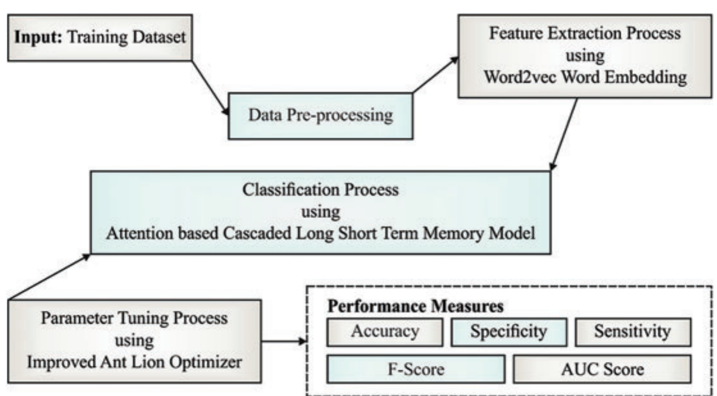

Figure 1: Block diagram of IALODL-OHSD approach

2.1 Pre-processing and Word Embedding

Data pre-processing is performed at the initial level to transform the Arabic social media text into a useful format. In addition, the word2vec word embedding process is utilized to produce word embedding. Word embedded refers to a set of language feature learning methods in NLP translating word tokens to machine-readable vectors [20]. Word2vec was a 2 layer neural network that translates text words into a vector. The input was a text corpus, and the output referred to a vector set. The benefit of word2vec is it could train largescale corpora to produce lower-dimension word vectors. Provided a sentence comprising of n words (x1, x2, x3, ..., xn−2, xn−1, xn), each word xi is translated into a real-value vector, e_i, characterized by

$$e_i = [w_1, w_2, w_3, \ldots, w_{n-2}, w_{n-1}, w_n] \in R^{n\times d} \tag{1}$$

Here w represents a word, and d denotes the size of the word embedding.

2.2 Offensive and Hate Speech Classification Model

For the classification process, the ACLSTM model is utilized in this study. A recurrent neural network (RNN) is a DL algorithm variant based on preceding and present input. Generally, it is applicable for the scenario whereby the dataset has a sequential correlation. When managing a long sequence of datasets, there is a gradient vanishing and exploiting problems [21]. To overcome these problems, an LSTM is applied that has an internal memory state that adds forget gate. The gate controls the effects of previous input and the time dependency. Bi-directional RNN (BiRNN) and Bi-directional LSTM (BiLSTM) are other variations which reflect previous input and consider the forthcoming input of a specific time frame. The study presents the cascaded uni-directional LSTM and Bi-LSTM RNN mechanisms. The technique encompasses the primary layer of BiRNN incorporated into the uni-directional RNN layer. The BiLSTM encompasses forward and backward tracks for learning patterns in two directions.

$$O_n^{f1},\ h_n^{f1},\ i_n^{f1} = L^{f1}\left(i_{n-1}^{f1},\ h_{n-1}^{f1},\ x_n : P^{f1}\right) \tag{2}$$

$$O_n^{b1},\ h_n^{b1},\ i_n^{b1} = L^{b1}\left(i_{n-1}^{b1},\ h_{n-1}^{b1},\ x_n : P^{b1}\right) \tag{3}$$

Eqs. (2) and (3) show the process of forwarding and backward tracks. Here, O_n^{f1}, h_n^{f1}, i_n^{f1} and O_n^{b1}, h_n^{b1}, i_n^{b1} indicate the output, hidden, and internal states of the present state for forward and backward LSTM tracks. x_n represents the consecutive input, P specifies the LSTM cell. The output from two tracks is incorporated as in Eq. (4) and dispatched to the following layers.

$$O_n^1 = O_n^{f1} + O_{N-n+1}^{b1}, \tag{4}$$

Uni-directional and Bi-RNN transform data to the abstract format and help to learn spatial dependency. The output from the uni-directional layer is accomplished as follows.

$$O_n^l, h_n^l, i_n^l = LSTM^l\ \left(i_{n-1}^l,\ h_{n-1}^l,\ O_n^{l-1}; P^l\right), \tag{5}$$

In Eq. (5), the output from the lower layer O_n^{l-1} is incorporated into the previously hidden state h_{n-1}^l, and internal state i_{n-1}^l for attaining output O_n^l of layer l, and P^l specifies a parameter of LSTM cell. The input data encompasses a sequence of samples $(x_1,\ x_2, \ldots,\ x_N)$, where each feature x_n is assumed at time $n\ \ (n = 1, 2, \ldots,\ N)$. The information is primarily categorized into the window of time segment N and given to the cascaded LSTM. Then, obtain predictive score vector for each time step $\left(O_1^L,\ O_2^L, \ldots,\ O_N^L\right)$ at the output. The whole predictive score is obtained by incorporating the prediction score for window N. At last, the predictive score is changed into a probability with a *softmax* layer over Y.

$$Y = \frac{1}{N}\sum\nolimits_{(n=1)}^{N} O_n^L. \tag{6}$$

Cascaded LSTM is used for simulating increment changes of n time steps, and each LSTM is applied to estimate the increment for a one-time step. In the study, the θ-increment learning mechanism learns increment of parameters through the cascading LSTM for high-frequency calculation, and θ signifies the target variable to be evaluated. In the case of an adequate dataset, the deep the structure, the improved the appropriate dataset. Once the models get deep, certain problems might arise. For example, a large variety of parameters in a single layer might result in the common amendment of the parameter in the subsequent layer. As a result, the training efficacy is considerably decreased. In addition, the output of the layer passes the activation function to the following layer, drastically exceeding the suitable extent of the activation function, which might result in an unsuccessful work of the neuron.

$$\mu_B = \frac{1}{m}\sum_{i=1}^{m} x_i$$
$$\sigma_B^2 = \frac{1}{m}\sum_{i=1}^{m} (x_i - \mu_B)^2, \tag{7}$$
$$\hat{x}_i = \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \varepsilon}}$$
$$y_i = \gamma^{\hat{x}_i} + \beta$$

In Eq. (7), x_i indicates the input value, y_i denotes the output afterwards BN, and m indicates the input number in the mini-batch. u_B shows the mean of input, σ_B^2 denotes the average alteration of the input and u_B in a mini-batch. $\hat{x}$ denotes the standardized x_i. r and β variables are learned through the BP mechanism. Afterwards, BN, the output is constrained to a fitting range, and the efficacy of the training procedure would be enhanced.

For imitating the presented method to increase the prediction capacity of the DL model, an attention model is employed in the domain of CV and NLP. The attention mechanism highlights the significance of different characteristics for predicting models by allocating weight to the feature. Afterwards, presenting the attention module, CLSTM was implemented effectively in a long time sequence. Here, the attention block is included among LSTM layers, and the output of attention is demonstrated in Eq. (8),

$$c_i = \sum_{j=1}^{L_\chi} \alpha_{ij} h_j \tag{8}$$

h_j characterizes the global feature, α_{ij} signifies the weight allocated to the feature through the attention module, *and* c_i shows the output of the attention block. The attention module is classified into hard and soft attention. Soft attention weight global input feature to emphasise particular regions, and the training process is distinguishable; therefore, the network straightforwardly adapts the end-to-end architecture. The hard attention reduces the trained cost by choosing the concerned region as the input. But, the input control technique is unsuitable for handling time-based problems. Furthermore, the hard attention making the network trained non-distinguishable, in addition to the gradient descending technique of RL, must be presented in the training model. The complication of training increases abruptly. Fig. 2 showcases the framework of BiLSTM.

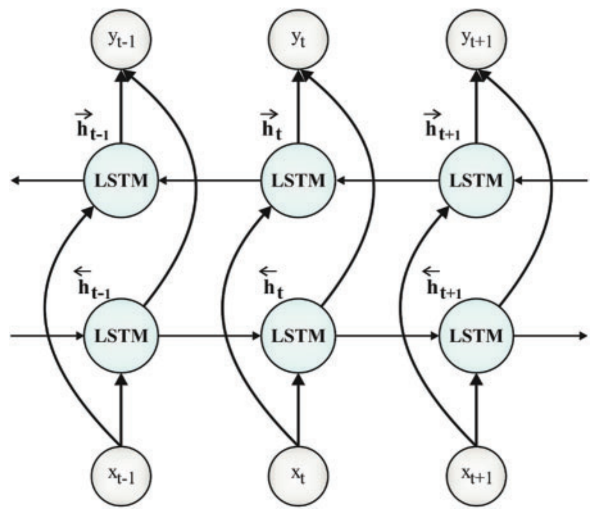

Figure 2: Structure of BiLSTM

2.3 Hyperparameter Tuning

To optimally modify the hyperparameter values of the ACLSTM model, the IALO algorithm is exploited in this study. The conventional ALO approach has better development and exploration abilities. The random walk of ants nearby the elite ant lion guarantees the convergence of the optimization technique [22]. The roulette selection technique increases the global search capability. On the other hand, still, the procedure has subsequent difficulties: (1) the predation-trapped border of antlion linearly reductions with the increasing iteration number, and the wandering border is comparatively distinct that is easier to lose the variety of the population, as well as the approach is easier to get trapped in local optimum. (2) The arbitrary walk of ants is effortlessly controlled by the elite antlion, resulting in the loss of global development and exploration capability.

To resolve the abovementioned challenges, the study presented a better ant lion optimization algorithm (IALO) to increase the global exploration capability and population diversity and improvement, prevent getting trapped into local optima, and increase the convergence accuracy.

Dynamic adaptive boundary amendment: here, the size of the trapped border linearly reduces with increasing iteration times. Though, this method doesn't vigorously reflect the recent efficiency and decreases the algorithm diversity. Considering a dynamic adaptive boundary adjustment is developed for enhancement, and it is shown in the following equation:

$$l = 10^{w} \frac{t}{T} * \left(1.5 - \cos\left(\frac{t\pi}{2T} * rand\right)\right) \tag{9}$$

In Eq. (9), rand denotes an arbitrary amount distributed uniformly among [0, 1]. The better formula adds a dynamic variable that makes the size of the trapped boundary show a non-linear reducing trend. By vigorously altering the range of trapped boundaries, the antlion's randomness and diversity are improved, and the approach's global exploration capability is enhanced.

Weighted elitism: it includes an elite weight variable δ that could alter the amount of ant lions arbitrarily roaming near elite ant lions and roulette designated through ants in different times, and it is shown below:

$$\delta = \delta_{\max} - t\frac{\delta_{\max} - \delta_{\min}}{T} + \xi * rand \tag{10}$$

In Eq. (10), δ_{max} and δ_{min} indicates the maximal and minimal inertial weight, δ denotes a constant:

$$Ant_i^t = \frac{(2-\delta) * R_A^t + \delta * R_E^t}{2} \tag{11}$$

The weighted elitism balances the wandering weights in dissimilar times and efficiently increases the development and exploration capability of the ALO technique.

3 Results and Discussion

The experimental validation of the IALODL-OHSD model is examined on two datasets, Dataset-1: OSACT-HS and Dataset-2: OSACT-OFF. The parameter settings are learning rate: 0.01, dropout: 0.5, batch size: 5, epoch count: 50, and activation: ReLU. The details linked to the datasets are given in Table 1.

Table 1: Dataset details

Dataset-1	No. of samples
Hate speech (HS)	506
Not hate speech (NHS)	9494
Total number of samples	10000
Dataset-2	No. of samples
Offensive (OFF)	1991
Not offensive (NOFF)	8009
Total number of samples	10000

Fig. 3 reports the confusion matrices produced by the IALODL-OHSD model on dataset-1. With run-1, the IALODL-OHSD model has recognized 445 samples into HS and 9450 samples into NHS. Along with the run-2, the IALODL-OHSD approach has simultaneously recognized 443 samples into HS and 9452 samples into NHS. With run-3, the IALODL-OHSD algorithm has recognized 408 samples into HS and 9458 samples into NHS. Also, in run-4, the IALODL-OHSD model recognized 380 samples into HS and 9460 samples into NHS.

Table 2 and Fig. 4 portray a detailed classifier outcome of the IALODL-OHSD model on dataset-1. The table values inferred that the IALODL-OHSD model had improved results under each run. For instance, on run-1, the IALODL-OHSD model has obtained average $accu_y$, $sens_y$, $spec_y$, F_{score}, and AUC_{score} of 98.95%, 93.74%, 93.74%, 94.45%, and 93.74% respectively. Next, For example, on run-2, the IALODL-OHSD technique has acquired average $accu_y$, $sens_y$, $spec_y$, F_{score}, and AUC_{score} of

98.95%, 93.55%, 93.55%, 94.43%, and 93.55% respectively. Concurrently, on run-3, the IALODL-OHSD method has attained average $accu_y$, $sens_y$, $spec_y$, F_{score}, and AUC_{score} of 98.66%, 90.13%, 90.13%, 92.60%, and 90.13% correspondingly, in the meantime, on run-4, the IALODL-OHSD model has reached average $accu_y$, $sens_y$, $spec_y$, F_{score}, and AUC_{score} of 98.40%, 87.37%, 87.37%, 90.89%, and 87.37% correspondingly.

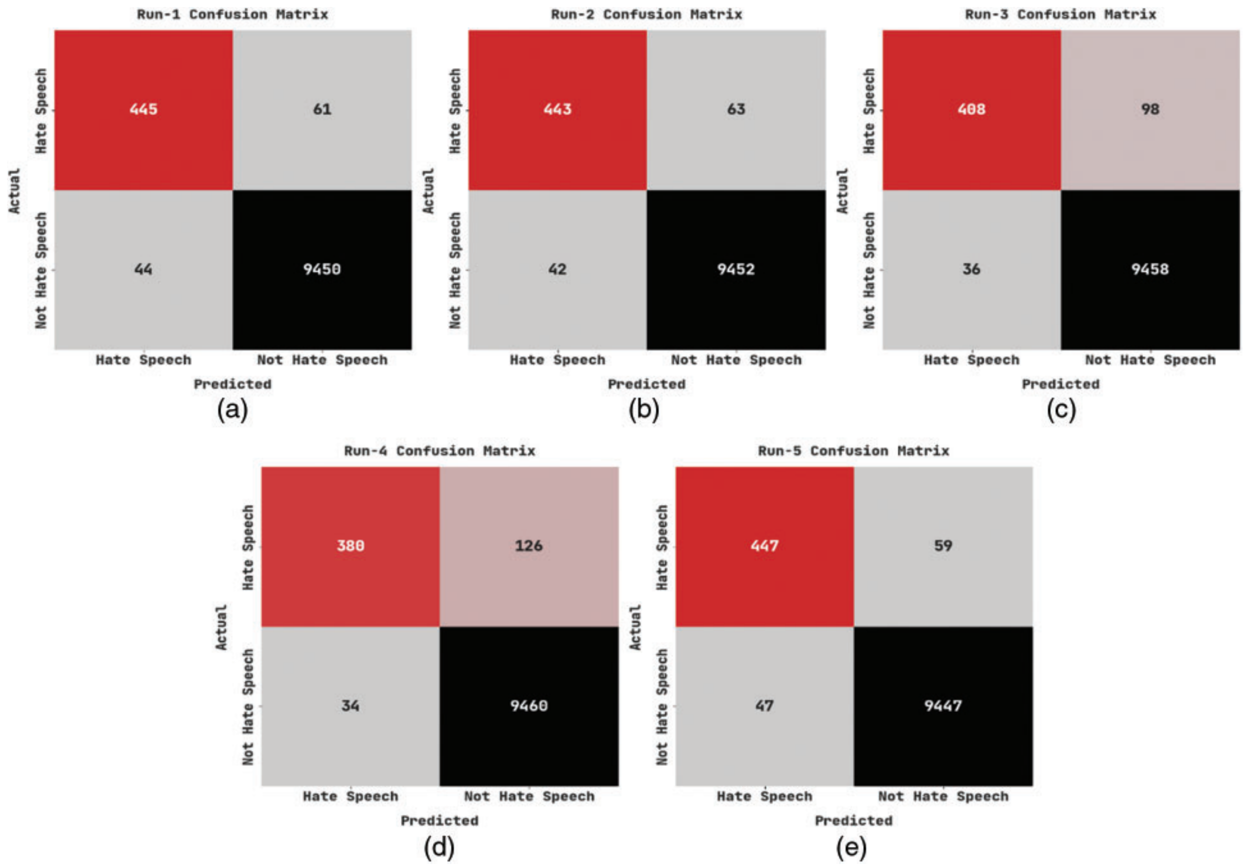

Figure 3: Confusion matrices of IALODL-OHSD approach under dataset-1 (a) run1, (b) run2, (c) run3, (d) run4, and (e) run5

Table 2: Result analysis of IALODL-OHSD approach with different measures and runs under dataset-1

Class labels	Accuracy	Sensitivity	Specificity	F-score	AUC score
Run-1					
Hate speech	98.95	87.94	99.54	89.45	93.74
Not hate speech	98.95	99.54	87.94	99.45	93.74
Average	98.95	93.74	93.74	94.45	93.74

(Continued)

Table 2: Continued

Class labels	Accuracy	Sensitivity	Specificity	F-score	AUC score
Run-2					
Hate speech	98.95	87.55	99.56	89.40	93.55
Not hate speech	98.95	99.56	87.55	99.45	93.55
Average	98.95	93.55	93.55	94.43	93.55
Run-3					
Hate speech	98.66	80.63	99.62	85.89	90.13
Not hate speech	98.66	99.62	80.63	99.30	90.13
Average	98.66	90.13	90.13	92.60	90.13
Run-4					
Hate speech	98.40	75.10	99.64	82.61	87.37
Not hate speech	98.40	99.64	75.10	99.16	87.37
Average	98.40	87.37	87.37	90.89	87.37
Run-5					
Hate speech	98.94	88.34	99.50	89.40	93.92
Not hate speech	98.94	99.50	88.34	99.44	93.92
Average	98.94	93.92	93.92	94.42	93.92

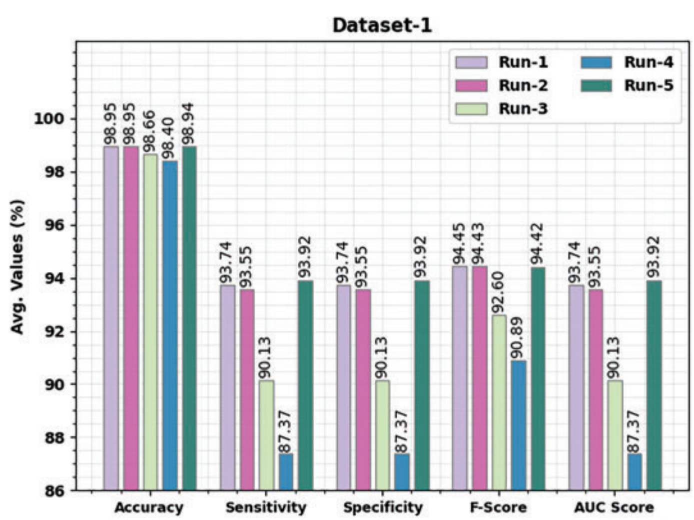

Figure 4: Average analysis of IALODL-OHSD approach with different runs under dataset-1

A clear precision-recall analysis of the IALODL-OHSD method on dataset-1 is displayed in Fig. 5. The figure is implicit that the IALODL-OHSD algorithm has resulted in enhanced precision-recall values under all classes.

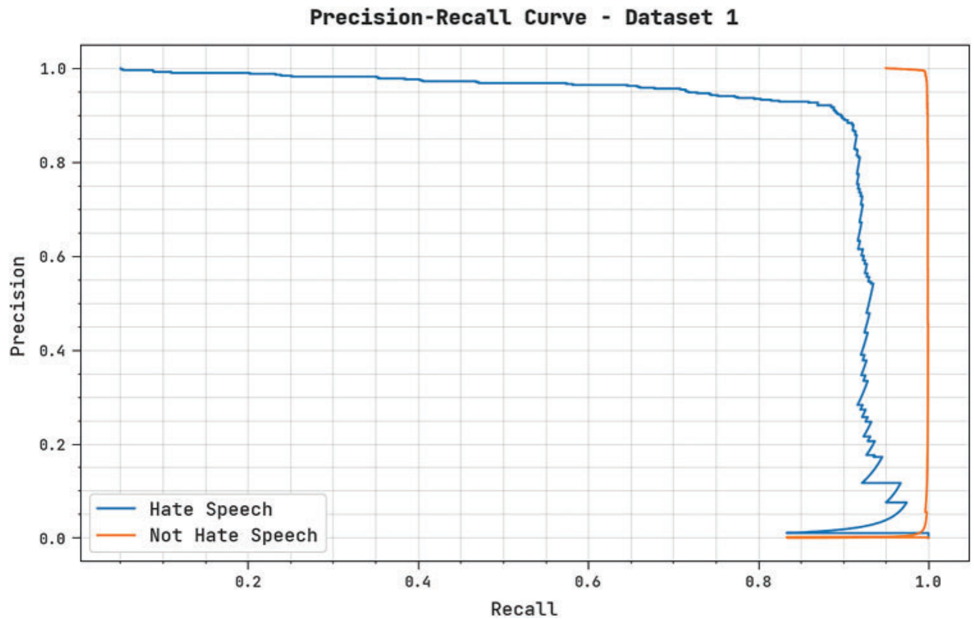

Figure 5: Precision-recall analysis of IALODL-OHSD approach under dataset-1

A brief ROC examination of the IALODL-OHSD technique on dataset-1 is shown in Fig. 6. The results represented the IALODL-OHSD approach has shown its ability in categorizing distinct classes on dataset-1.

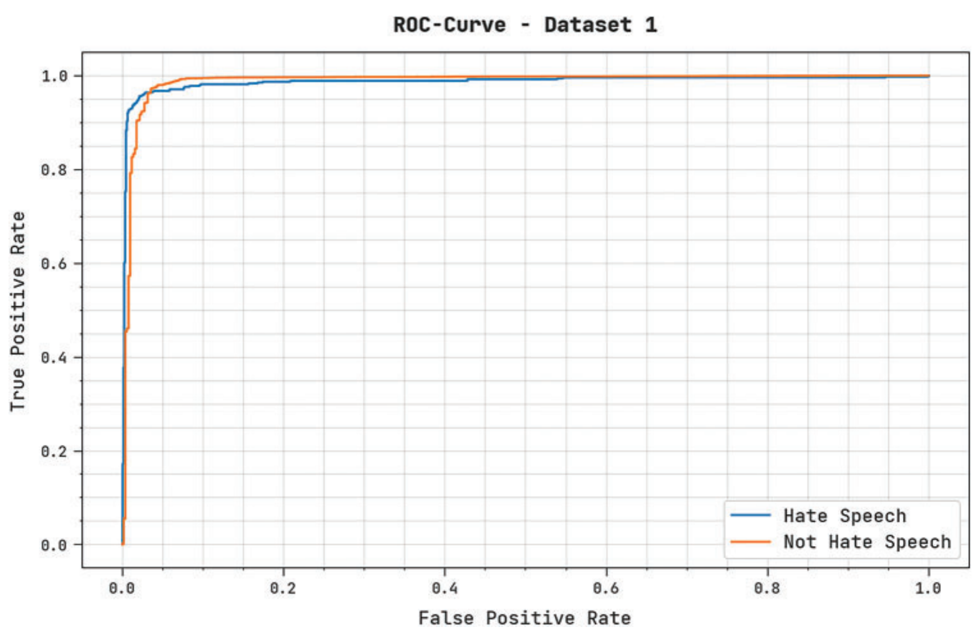

Figure 6: ROC analysis of IALODL-OHSD approach under dataset-1

Fig. 7 establishes the confusion matrices produced by the IALODL-OHSD model on dataset-2. With run-1, the IALODL-OHSD method has recognized 1863 samples into OFFSEN and 7907 samples into NOT OFFSSEN; with run-2, the IALODL-OHSD model has recognized 1801 samples into OFFSEN and 7911 samples into NOT OFFSSEN, Additionally With run-3, the IALODL-OHSD approach has recognized 1911 samples into OFFSEN and 7939 samples into NOT OFFSSEN, Meanwhile With run-4, the IALODL-OHSD technique has recognized 1866 samples into OFFSEN and 7956 samples into NOT OFFSSEN.

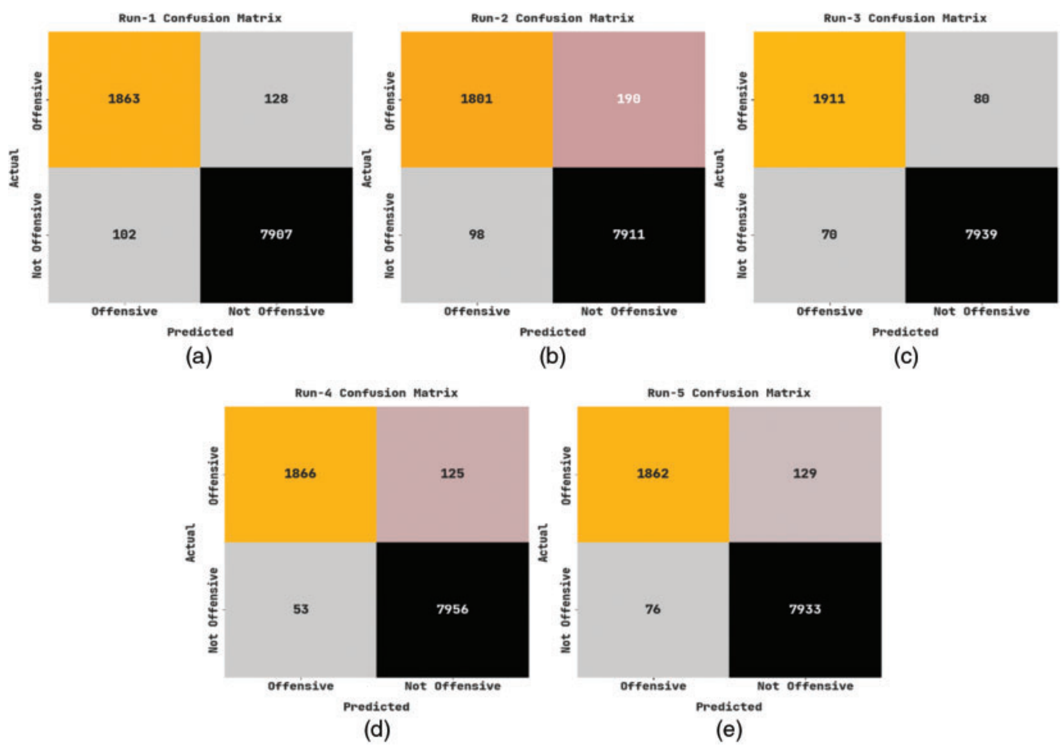

Figure 7: Confusion matrices of IALODL-OHSD approach under dataset-2 (a) run1, (b) run2, (c) run3, (d) run4, and (e) run5

Table 3 and Fig. 8 depict detailed classifier outcomes of the IALODL-OHSD algorithm on dataset-2. The table values denoted the IALODL-OHSD model has shown improved results under each run. For example, on run-1, the IALODL-OHSD model has obtained average $accu_y$, $sens_y$, $spec_y$, F_{score}, and AUC_{score} of 97.70%, 96.15%, 96.15%, 96.38%, and 96.15% respectively, Then, on run-2, the IALODL-OHSD model has acquired average $accu_y$, $sens_y$, $spec_y$, F_{score}, and AUC_{score} of 97.12%, 94.62%, 94.62%, 95.40%, and 94.62% correspondingly, in the meantime, on run-3, the IALODL-OHSD model has gained average $accu_y$, $sens_y$, $spec_y$, F_{score}, and AUC_{score} of 98.50%, 97.55%, 97.55%, 97.64%, and 97.55% correspondingly, Next, on run-4, the IALODL-OHSD model has obtained average $accu_y$, $sens_y$, $spec_y$, F_{score}, and AUC_{score} of 98.22%, 96.53%, 96.53%, 97.17%, and 96.53% correspondingly.

Table 3: Result analysis of IALODL-OHSD approach with different measures and runs under dataset-2

Class labels	Accuracy	Sensitivity	Specificity	F-score	AUC score
Run-1					
Hate speech	97.70	93.57	98.73	94.19	96.15
Not hate speech	97.70	98.73	93.57	98.57	96.15
Average	97.70	96.15	96.15	96.38	96.15
Run-2					
Hate speech	97.12	90.46	98.78	92.60	94.62
Not hate speech	97.12	98.78	90.46	98.21	94.62
Average	97.12	94.62	94.62	95.40	94.62
Run-3					
Hate speech	98.50	95.98	99.13	96.22	97.55
Not hate speech	98.50	99.13	95.98	99.06	97.55
Average	98.50	97.55	97.55	97.64	97.55
Run-4					
Hate speech	98.22	93.72	99.34	95.45	96.53
Not hate speech	98.22	99.34	93.72	98.89	96.53
Average	98.22	96.53	96.53	97.17	96.53
Run-5					
Hate speech	97.95	93.52	99.05	94.78	96.29
Not hate speech	97.95	99.05	93.52	98.72	96.29
Average	97.95	96.29	96.29	96.75	96.29

A clear precision-recall scrutiny of the IALODL-OHSD method on dataset-2 is shown in Fig. 9. The figure indicated that the IALODL-OHSD algorithm has resulted in enhanced precision-recall values under all classes.

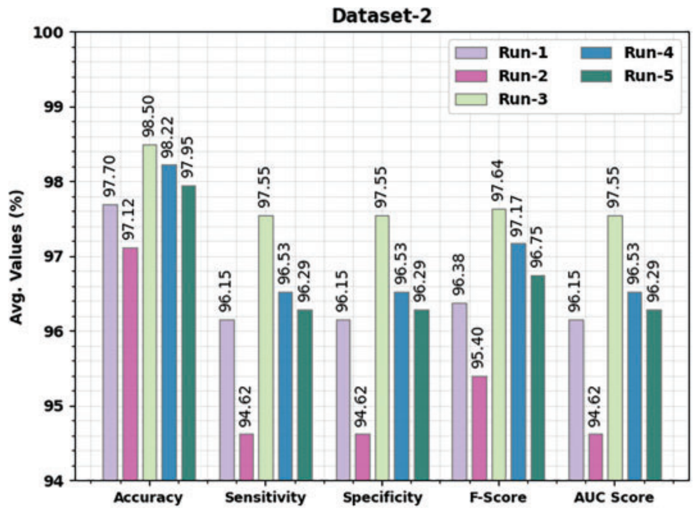

Figure 8: Average analysis of IALODL-OHSD approach with different runs under dataset-2

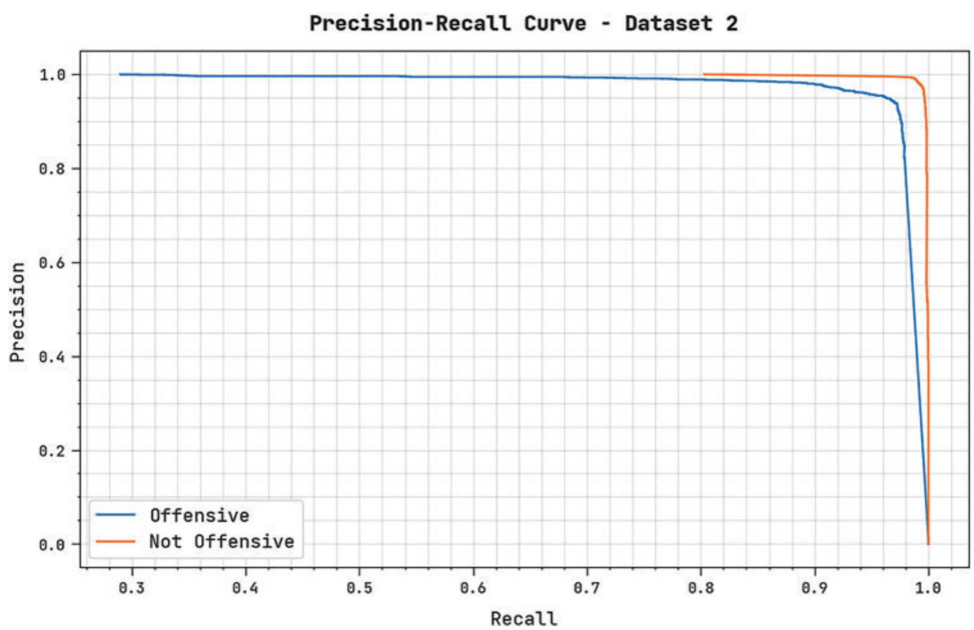

Figure 9: Precision-recall analysis of IALODL-OHSD approach under dataset-2

A brief ROC investigation of the IALODL-OHSD method on dataset-2 is presented in Fig. 10. The results denoted the IALODL-OHSD approach has shown its ability to categorize distinct classes on dataset-2.

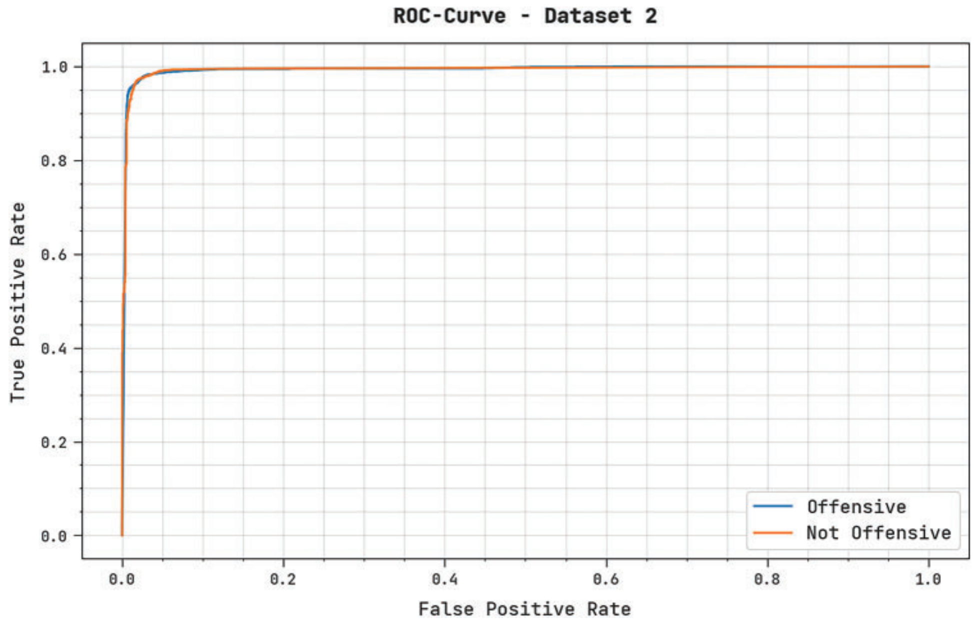

Figure 10: Precision-recall analysis of IALODL-OHSD approach under dataset-2

Table 4 and Fig. 11 highlight the comparison outcomes of the IALODL-OHSD model with other models on dataset-1 [1]. The results indicated that the IALODL-OHSD model had offered an increased $accu_y$ of 98.95%, whereas the AraBERT v02, MTL-A-L, MTL-M-L, MTL-A-T, MTL-M-T, MTL-AraBERT, and MTL-MarBERT models have attained reduced $accu_y$ of 94.61%, 94.71%, 95.05%, 93.90%, 94.82%, 94.63%, and 95.33% respectively. On the other hand, the results denoted the IALODL-OHSD model has rendered an increased $F1_{score}$ of 94.45%, whereas the AraBERT v02, MTL-A-L, MTL-M-L, MTL-A-T, MTL-M-T, MTL-AraBERT, and MTL-MarBERT methods have achieved reduced $F1_{score}$ of 90.47%, 90.52%, 92.55%, 90.96%, 92.37%, 91.50%, and 92.22% correspondingly.

Table 4: Comparative analysis of IALODL-OHSD approach with existing methodologies under dataset-1

Dataset-1		
Methods	Accuracy	F1-score
IALODL-OHSD	98.95	94.45
AraBERT v02 model	94.61	90.47
MTL-A-L model	94.71	90.52
MTL-M-L model	95.05	92.55
MTL-A-T model	93.90	90.96
MTL-M-T model	94.82	92.37
MTL-AraBERT model	94.63	91.50
MTL-MarBERT model	95.33	92.22

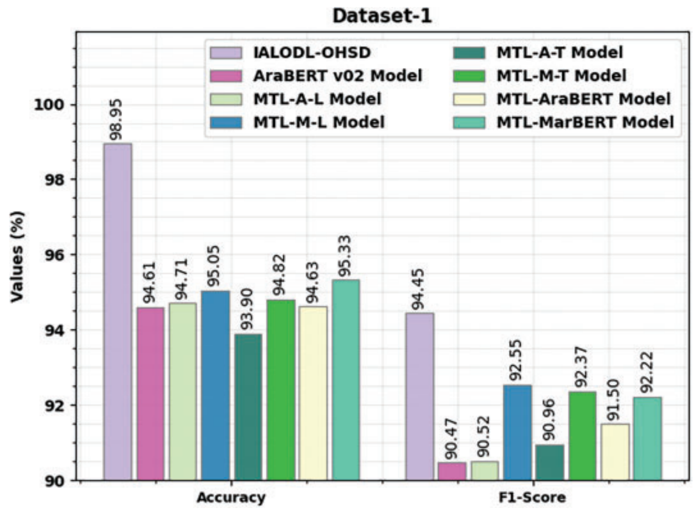

Figure 11: $Accu_y$ and $F1_{score}$ analysis of IALODL-OHSD approach with existing methodologies under dataset-1

Table 5 and Fig. 12 highlight the comparative outcomes of the IALODL-OHSD model with other models on dataset 2. The results indicated that the IALODL-OHSD technique had presented increased $accu_y$ of 98.50%. In contrast, the AraBERT v02, MTL-A-L, MTL-M-L, MTL-A-T, MTL-M-T, MTL-AraBERT, and MTL-MarBERT models have attained reduced $accu_y$ of 93.92%, 93.91%, 94.97%, 94.18%, 94.88%, 94.21%, and 95.08% individually. Besides, the results indicated that the IALODL-OHSD model had offered increased $F1_{score}$ of 97.64%, whereas the AraBERT v02, MTL-A-L, MTL-M-L, MTL-A-T, MTL-M-T, MTL-AraBERT, and MTL-MarBERT models have attained reduced $F1_{score}$ of 89.99%, 90.48%, 92.30%, 90.75%, 91.94%, 91.45%, and 92.33% correspondingly.

Table 5: Comparative analysis of IALODL-OHSD approach with existing methodologies under dataset-2

Dataset-2		
Methods	Accuracy	F1-score
IALODL-OHSD	98.50	97.64
AraBERT v02 model	93.92	89.99
MTL-A-L model	93.91	90.48
MTL-M-L model	94.97	92.30
MTL-A-T model	94.18	90.75
MTL-M-T model	94.88	91.94
MTL-AraBERT model	94.21	91.45
MTL-MarBERT model	95.08	92.33

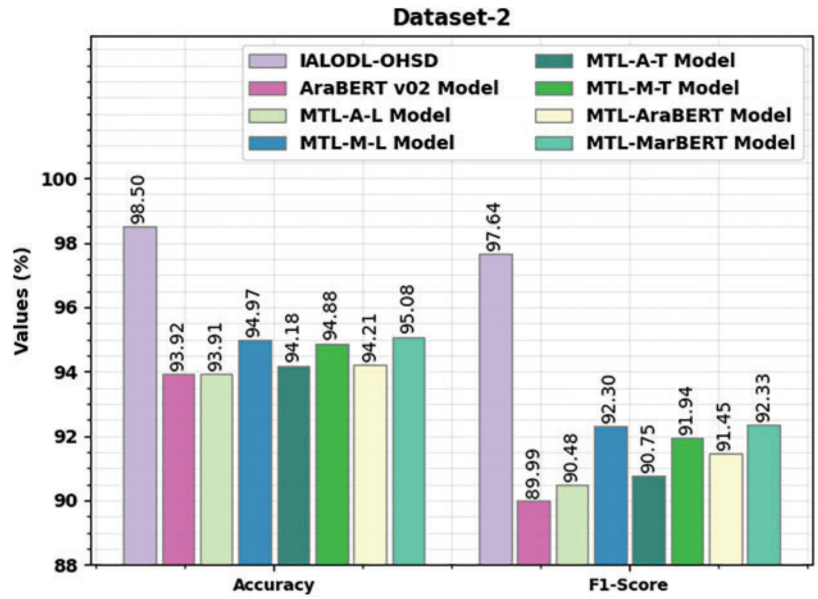

Figure 12: $Accu_y$ and $F1_{score}$ analysis of IALODL-OHSD approach with existing methodologies under datatset-2

4 Conclusion

This study devised a new IALODL-OHSD technique to detect and classify offensive/hate speech expressed on social media. In the IALODL-OHSD model, a three-stage process is performed namely pre-processing, word embedding, and classification. Primarily, data pre-processing is performed to transform the Arabic social media text into a useful format. In addition, the word2vec word embedding process is utilized to produce word embedding. For the classification process, the ACLSTM model is utilized. Finally, the IALO algorithm is exploited as a hyperparameter optimizer to boost classifier results. To illustrate a brief result analysis of the IALODL-OHSD model, a detailed set of simulations were performed. The extensive comparison study portrayed the enhanced performance of the IALODL-OHSD model over other approaches. As a future extension, the proposed model can be modified for emotion classification in microblogging platforms.

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: 22UQU4340237DSR43.

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

References

[1] W. Aldjanabi, A. Dahou, M. A. A. Al-qaness, M. A. Elaziz, A. M. Helmi *et al.,* "Arabic offensive and hate speech detection using a cross-corpora multi-task learning model," *Informatics*, vol. 8, no. 4, pp. 69, 2021.

[2] A. S. Alammary, "BERT models for Arabic text classification: A systematic review," *Applied Sciences*, vol. 12, no. 11, pp. 5720, 2022.

[3] R. Alshalan and H. Al-Khalifa, "A deep learning approach for automatic hate speech detection in the Saudi twittersphere," *Applied Sciences*, vol. 10, no. 23, pp. 8614, 2020.

[4] R. Duwairi, A. Hayajneh and M. Quwaider, "A deep learning framework for automatic detection of hate speech embedded in Arabic tweets," *Arabian Journal for Science and Engineering*, vol. 46, no. 4, pp. 4001–4014, 2021.

[5] 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.

[6] R. Alshalan, H. A. Khalifa, D. Alsaeed, H. Al-Baity and S. Alshalan, "Detection of hate speech in COVID-19–related tweets in the Arab region: Deep learning and topic modeling approach," *Journal of Medical Internet Research*, vol. 22, no. 12, pp. e22609, 2020.

[7] 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.

[8] F. Y. A. Anezi, "Arabic hate speech detection using deep recurrent neural networks," *Applied Sciences*, vol. 12, no. 12, pp. 6010, 2022.

[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. Poletto, V. Basile, M. Sanguinetti, C. Bosco and V. Patti, "Resources and benchmark corpora for hate speech detection: A systematic review," *Lang Resources & Evaluation*, vol. 55, no. 2, pp. 477–523, 2021.

[11] 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.

[12] S. Alsafari and S. Sadaoui, "Semi-supervised self-training of hate and offensive speech from social media," *Applied Artificial Intelligence*, vol. 35, no. 15, pp. 1621–1645, 2021.

[13] E. Pronoza, P. Panicheva, O. Koltsova and P. Rosso, "Detecting ethnicity-targeted hate speech in Russian social media texts," *Information Processing & Management*, vol. 58, no. 6, pp. 102674, 2021.

[14] R. Ali, U. Farooq, U. Arshad, W. Shahzad and M. Beg, "Hate speech detection on Twitter using transfer learning," *Computer Speech & Language*, vol. 74, pp. 101365, 2022.

[15] Z. Mossie and J. H. Wang, "Vulnerable community identification using hate speech detection on social media," *Information Processing & Management*, vol. 57, no. 3, pp. 102087, 2020.

[16] I. Aljarah, M. Habib, N. Hijazi, H. Faris, R. Qaddoura *et al.,* "Intelligent detection of hate speech in Arabic social network: A machine learning approach," *Journal of Information Science*, vol. 47, no. 4, pp. 483–501, 2021.

[17] A. Elouali, Z. Elberrichi and N. Elouali, "Hate speech detection on multilingual twitter using convolutional neural networks," *Revue d'Intelligence Artificielle*, vol. 34, no. 1, pp. 81–88, 2020.

[18] M. Khalafat, J. S. Alqatawna, R. M. H. Al-Sayyed, M. Eshtay and T. Kobbaey, "Violence detection over online social networks: An arabic sentiment analysis approach," *International Journal of Interactive Mobile Technologies*, vol. 15, no. 14, pp. 90, 2021.

[19] N. Albadi, M. Kurdi and S. Mishra, "Investigating the effect of combining GRU neural networks with handcrafted features for religious hatred detection on Arabic Twitter space," *Social Network Analysis and Mining*, vol. 9, no. 1, pp. 1–19, 2019.

[20] I. Chalkidis and D. Kampas, "Deep learning in law: Early adaptation and legal word embeddings trained on large corpora," *Artificial Intelligence and Law*, vol. 27, no. 2, pp. 171–198, 2019.

[21] W. Qi, X. Zhang, N. Wang, M. Zhang and Y. Cen, "A spectral-spatial cascaded 3d convolutional neural network with a convolutional long short-term memory network for hyperspectral image classification," *Remote Sensing*, vol. 11, no. 20, pp. 2363, 2019.

[22] M. M. Mafarja and S. Mirjalili, "Hybrid binary ant lion optimizer with rough set and approximate entropy reducts for feature selection," *Soft Computing*, vol. 23, no. 15, pp. 6249–6265, 2019.