



ARTICLE

Multi-Versus Optimization with Deep Reinforcement Learning Enabled Affect Analysis on Arabic Corpus

Mesfer Al Duhayyim^{1,*}, Badriyya B. Al-onazi², Jaber S. Alzahrani³, Hussain Alshahrani⁴, Mohamed Ahmed Elfaki⁴, Abdullah Mohamed⁵, Ishfaq Yaseen⁶, Gouse Pasha Mohammed⁶, Mohammed Rizwanullah⁶ and Abu Sarwar Zamani⁶

¹Department of Computer Science, College of Sciences and Humanities-Aflaj, Prince Sattam bin Abdulaziz University, Aflaj, 16273, Saudi Arabia

²Department of Language Preparation, Arabic Language Teaching Institute, Princess Nourah Bint Abdulrahman University, Riyadh, 11671, Saudi Arabia

³Department of Industrial Engineering, College of Engineering at Alqunfudah, Umm Al-Qura University, Makkah, 24211, Saudi Arabia

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

⁵Research Centre, Future University in Egypt, New Cairo, 11845, Egypt

⁶Department of Computer and Self Development, Preparatory Year Deanship, Prince Sattam bin Abdulaziz University, AlKharj, Saudi Arabia

*Corresponding Author: Mesfer Al Duhayyim. Email: m.alduhayyim@psau.edu.sa

Received: 29 June 2022 Accepted: 04 November 2022 Published: 09 November 2023

ABSTRACT

Sentiment analysis (SA) of the Arabic language becomes important despite scarce annotated corpora and confined sources. Arabic affect Analysis has become an active research zone nowadays. But still, the Arabic language lags behind adequate language sources for enabling the SA tasks. Thus, Arabic still faces challenges in natural language processing (NLP) tasks because of its structure complexities, history, and distinct cultures. It has gained lesser effort than the other languages. This paper developed a Multi-versus Optimization with Deep Reinforcement Learning Enabled Affect Analysis (MVODRL-AA) on Arabic Corpus. The presented MVODRL-AA model majorly concentrates on identifying and classifying effects or emotions that occurred in the Arabic corpus. Firstly, the MVODRL-AA model follows data pre-processing and word embedding. Next, an n-gram model is utilized to generate word embeddings. A deep Q-learning network (DQLN) model is then exploited to identify and classify the effect on the Arabic corpus. At last, the MVO algorithm is used as a hyperparameter tuning approach to adjust the hyperparameters related to the DQLN model, showing the novelty of the work. A series of simulations were carried out to exhibit the promising performance of the MVODRL-AA model. The simulation outcomes illustrate the betterment of the MVODRL-AA method over the other approaches with an accuracy of 99.27%.

KEYWORDS

Arabic language; Arabic corpus; natural language processing; affect analysis; deep learning



1 Introduction

Arabic is a rapidly growing language on Twitter and the official language for twenty-two countries. But not much work has been done on analysing the Arabic language [1]. This can be because of various dialects and complications in the Arabic language that make it difficult to build a single mechanism for detecting emotions and sentiments in the Arabic language [2]. It can be divided into 2 major categories: Dialectal Arabic (DA) and Standard Arabic. Standard Arabic has 2 types they are Modern Standard Arabic (MSA) and Classical Arabic (CA) [3]. Even though MSA became the principal language of Arab countries and was printed in books and educated in schools [4], DA was spoken as a native language in individuals' informal daily interaction and contains a robust existence in text and comments in emails or on microblogging networks. There exists only one MSA language for every Arabic speaker, but numerous dialects do not have formal written formats [5]. These results in the absence of lexicon sources for such dialects, and official grammar rulebooks do not work as proficiently with DA and MSA.

Twitter is a rich data bank full of opinions, sentiments, and emotions. Sentiment analysis (SA) is to classify a subjective text as negative, positive, or neutral [6]; emotion detection detects feelings types via the expression of texts, like sadness, anger, fear, and joy. Automated emotion recognition or affect detection focuses on identifying human affective states like sadness, joy, and love from several modalities, which include video, text, image, and audio [7]. Since a particular natural language processing (NLP) task, emotion identification from textual data has been a promising research matter for the past few years. Substantial attempts have been made to build a perfect automatic system that detects real human emotions from the text. A multilabel classifier challenge was assumed; more than one emotion was taken in a study [8]. Therefore, it provides more difficulties beyond multiclass or binary classifier issues. Automated text emotion recognition helps analyze user feelings, attitudes, and sentiments from online text like product reviews, tweets, comments, Facebook status, blogs, and news and maybe implied to several domains [9], i.e., customer services, chatbots, mental health monitoring, and e-learning mechanisms. Deep learning (DL) is described as a sub-field of machine learning (ML) which is made up of multiple hidden layers which were devised for feature extraction and complicated modelling [10]. DL has resulted in breakthroughs in several NLP applications, like affect analysis and Arabic sentiment.

This paper presents a Multi-versus Optimization with Deep Reinforcement Learning Enabled Affect Analysis (MVODRL-AA) on Arabic Corpus. The presented MVODRL-AA model majorly concentrates on identifying and classifying effects or emotions that occurred in the Arabic corpus. Firstly, the MVODRL-AA model follows data pre-processing and word embedding. Next, an n-gram model is utilized to generate word embeddings. A deep Q-learning network (DQLN) model is then exploited to identify and classify the effects on the Arabic corpus. At last, the MVO algorithm is used as a hyperparameter tuning approach to adjust the hyperparameters related to the DQLN model. A series of simulations were carried out to showcase the promising performance of the MVODRL-AA model.

The rest of the paper is organized as follows. [Section 2](#) offers the related works, and [Section 3](#) introduces the proposed model. Next, [Section 4](#) gives an experimental validation, and [Section 5](#) concludes the work.

2 Literature Review

AlGhamdi et al. [11] provided an intellectual mechanism for analysing Arabic tweets to detect suspicious messages. And needed Arabic tweet datasets from the microblog social network site Twitter through Twitter Streaming Application Programming Interface and saves in a desirable file format. The mechanism tokenizes and pre-processes tweet datasets. Manual labelling was conducted on the

tweet dataset for not-suspicious (label 0) and suspicious (label 1) classes. The labelled tweet datasets can be utilized for training a classifier by using supervised ML systems to identify suspicious actions. At the testing stage, the system processes the unlabeled tweet datasets and identifies whether they belong to the not-suspicious or suspicious class. Alsayat et al. [12] introduced a complete strategy proposal for Arabic Language Sentiment Analysis (ALSA). The ALSA structure examines the feelings and opinions at every language level and the significance of framing an annotated corpus that helps understand an Arabic sentence from the phonetics level to the metonymy and rhetorical levels.

Alharbi et al. [13] suggested a SA with DL (SA-DL) related method for predicting the polarity of sentiments and opinions. Two kinds of recurrent neural networks (RNNs) were utilized for learning higher-level representations. Afterwards, mitigating the data dependency challenges and raising the forcefulness of the method, 3 distinct classifier techniques have been used for producing the concluding output. Ombabi et al. [14] suggested a new DL method for ALSA based on a single-layer convolutional neural network (CNN) structure for local feature extraction and 2 layers of long short-term memory (LSTM) for maintaining long-run dependences. The feature maps studied by LSTM and CNN were passed to the support vector machine (SVM) classifier for generating the concluding categorization. This method can be maintained by the FastText word embedding method. In [15], an efficient Bidirectional LSTM Network (BiLSTM) can be examined for enhancing ALSM by applying Forward-Backwards encapsulating context information from the Arabic feature series.

The authors in [16] introduced an optimized heterogeneous stacking ensemble technique to enhance the SA on Arabic text using three distinct pre-trained models. The proposed model enhances the SA and prediction outcomes. The authors in [17] developed an ML model for Arabic tweet analysis. Here, the Word2Vec approach is applied for word embedding with two pretrained continuous bag-of-words (CBOW) approaches examined. Finally, Naïve Bayes (NB) model is employed as a baseline classification model. The authors in [18] presented various DL models for Arabic SA. These models have been validated on the hybrid dataset, and the Arabic book reviews dataset (BRAD). The presented model investigated deep networks' capability to detect discriminating features from data depicted at the character level.

3 The Proposed Model

In this paper, an MVODRL-AA technique was introduced to identify and classify effects or emotions that occurred in the Arabic corpus. Firstly, the MVODRL-AA model follows data pre-processing and word embedding. Next, the n-gram model is utilized to generate word embeddings. Then, the MVO with the DQLN model is exploited to identify and classify the effect on the Arabic corpus. Fig. 1 depicts the overall block diagram of the MVODRL-AA approach.

3.1 Data Pre-Processing and Word Embedding

At the beginning level, the MVODRL-AA model follows data pre-processing and word embedding. Tweet pre-processing is the primary level in the suggested technique, transforming Arabic tweets to a form suitable and appropriate for the multilabel emotion classifier mechanism. Such pre-processing tasks encompass eliminating diacritics, punctuation, digits, stop words, and Latin characters and inspecting the process of tokenization, light stemming, and normalization. Moreover, improved the tweets by copying the embedded emoji in their respective Arabic words. Such language was used to reduce the tweets' noisiness and ambiguity to increase the proposal's effectiveness and accuracy.

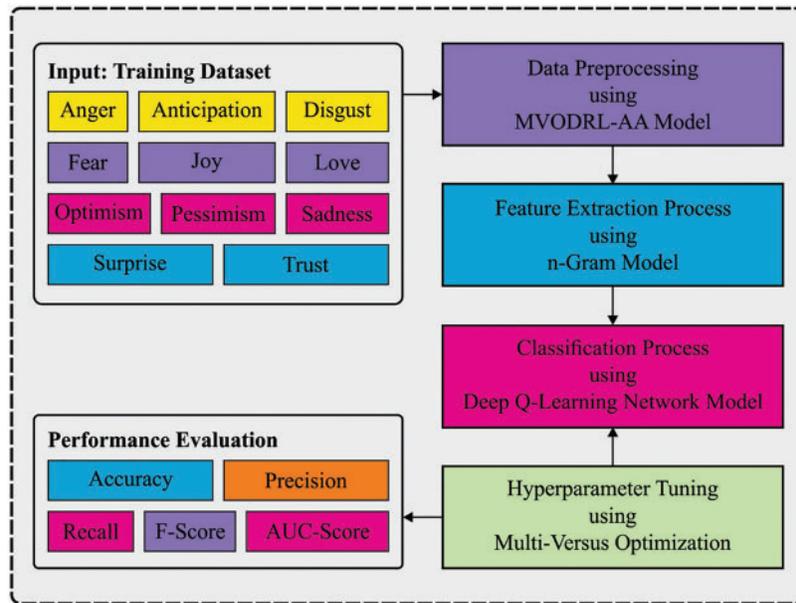


Figure 1: Overall block diagram of MVODRL-AA approach

Lexical n-gram methods were broadly utilized in NLP for syntax feature mapping & statistical analysis [19]. An advanced n-gram method for analyzing the generated corpus comprises tokenized words to find the word's popularity or set of nearest terms. The probability of the existence of a series is computed by utilizing the probability chain rule:

$$P(x_1, x_2, x_3, x_n) = P(x_1) P(x_2|x_1) P(x_3|x_1, x_2) \dots P(x_n|x_1, x_2, x_3, \dots, x_{(n-1)}) \quad (1)$$

$$= \prod_{i=1}^n P(x_i|x_1^{(i-1)}) \quad (2)$$

For instance, it could assume a sentence as “Still COVID19 wave is running”. As per the probability chain rule, $P(\text{“Still Covid-19 wave is running”}) = P(\text{“Still”}) \times P(\text{“Covid-19”} | \text{“Still”}) \times P(\text{“wave”} | \text{“Still Covid-19”}) \times P(\text{“is”} | \text{“Still Covid-19 wave”}) \times P(\text{“running”} | \text{“Still Covid-19 wave is”})$.

The words probabilities in every sentence after applications of the probability chain rule:

$$P(W_1, W_2, W_3, W_n) = \prod_j P(W_j|W_1, W_2, W_3, W_{(j-1)}) \quad (3)$$

$$= \prod_{j=1}^n P(W_j|W_1^{(j-1)}) \quad (4)$$

The bigram method predicts the word probability a by utilizing the conditional probability $P(W_i|W_{i-1})$ of one previous term on the specified condition of each preceding word $P(W_i|W_{i-1})$.

$$P(W_1, W_2) = \prod_{i=2} P(W_2|W_1) \quad (5)$$

The expression for the probability can be:

$$P(W_k|W_{(k-1)}) = \frac{\text{count}(W_{(k-1)} W_k)}{\text{count}(W_{(k-1)})} \quad (6)$$

The famous trigrams, unigrams, and bigrams are found in our corpus using the n-gram method.

3.2 DQLN-Based Affect Classification

At this stage, the model is exploited to identify and classify the effect on the Arabic corpus. Deep reinforcement learning (DRL) integrates three mechanisms such as reward, state, and action. The DRL agent aims to learn the mapping function from the state to action spaces. Next, the DRL agent obtains an award. The objective is to maximize the overall rewards. The classification method of the DQLN method is a function that receives the sample and returns the label probability [20].

$$\pi(a|s) = P(a_t = a | s_t = s) \quad (7)$$

The classifier agent's goal is to recognize an instance in the trainable dataset correctly. Meanwhile, the classifier agent obtains a positive reward once it properly recognizes the instance; it accomplishes the objective by maximizing g_t cumulative rewards:

$$g_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k} \quad (8)$$

In the RL technique, there exists a function that calculates the quality of state and action incorporation called as Q function:

$$Q^\pi(s, a) = E_\pi [g_t | s_t = s, a_t = a] \quad (9)$$

According to the Bellman equation, the Q function is expressed by:

$$Q^\pi(s, a) = E_\pi [r_t + \gamma Q^\pi(s_{t+1}, a_{t+1}) | s_t = s, a_t = a] \quad (10)$$

The classification agent maximizes the cumulative reward by solving the optimal Q^* function, and the greedy approach in an optimal Q^* function is the best classification method π^* for DQLN.

$$\pi^*(a|s) = \begin{cases} 1, & \text{if } a = \operatorname{argmax}_a Q^*(s, a) \\ 0, & \text{else} \end{cases} \quad (11)$$

$$Q^*(s, a) = EA [r_t + \gamma \max_{a'} Q^*(s_{t+1}, a_{t+1}) | s_t = s, a_t = a] \quad (12)$$

The table records the Q function in the lower dimension finite state space. However, in the higher dimension constant state space, the Q function is not resolved, yet a deep Q -learning method is proposed that fits the Q function using a deep neural network (DNN). In the deep Q -learning method, the transmission data (s, a, r, s') accomplished from (12) are stored in the replay memory M . The agent arbitrarily instances a mini-batch of transition B from M and implements the gradient descent stage on the Deep Q network according to the loss function:

$$L(\theta_k) = \sum_{(s,a,r,s') \in B} (y - Q(s, a; \theta_k))^2 \quad (13)$$

In Eq. (13), y indicates the evaluated target of the Q function as follows:

$$y = r + (1 - t) \gamma \max_{a'} Q(s', a'; \theta_{k-1}) \quad (14)$$

In Eq. (14), s' denotes the following state of s , a' signifies the act performed by the agent in state s' , $t = 1$ if *terminal* = *True*; then $t = 0$, and it is given below:

$$\frac{\nabla L(\theta_k)}{\nabla \theta_k} = -2 \sum_{(s,a,r,s') \in B} (y - Q(s, a; \theta_k)) \frac{\nabla Q(s, a; \theta_k)}{\nabla \theta_k} \quad (15)$$

Now, the optimal Q^* function is attained by minimalizing the loss function (8), and the greedy policy in the optimal Q^* function gets the highest cumulative reward. As a result, the optimal classification method $\pi^* : \mathcal{S} \rightarrow \mathcal{A}$ for DQLN is attained. Fig. 2 illustrates the structure of Deep Q-Networks.

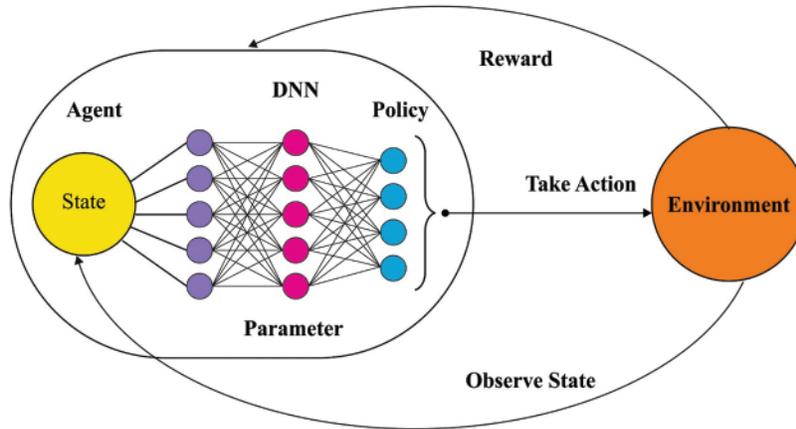


Figure 2: Structure of deep Q-networks

3.3 MVO-Based Hyperparameter Tuning

Finally, the MVO algorithm is used as a hyperparameter tuning approach to adjust the hyperparameters related to the DQLN model. MVO employs the black-and-white hole conception to explore the search space, where it employs a wormhole to exploit the search space [21]. Similar to each evolutionary algorithm, it initiates the optimization technique by initializing a population of solutions and trying to improve the solution through a predetermined iteration count. In the study, the individual development in every population is implemented according to the theory regarding the presence of a multi-universe. In this theory, every object in the universe can be regarded as a variable, and every solution to an optimizing issue is considered a universe. In the optimization technique, MVO follows the subsequent stages:

- The high inflation rate is proportionate to the low probability of black holes.
- The high inflation rate is proportionate to the high probability of white holes.
- Object in the universe with low inflation rate tends to obtain further objects via black holes.
- Object in the universe with high inflation rate moves from white to black holes.
- Objects in each universe might face random motion toward the better universe nevertheless of the inflation rate.

In MVO, a wormhole tunnel is often determined among a universe, and the better universe formed solar.

$$Y_i^j = \begin{cases} \begin{cases} Y_j + \text{TDR} \times ((ubp_j - lbp_j) \times R_4 + lbp_j) & R_3 < 0.5 \\ Y_j - \text{TDR} \times ((ubp_j - lbp_j) \times R_4 + lbp_j) & R_3 \geq 0.5 \end{cases} & R_2 < WEP \\ Y_i^j & R_2 \geq WEP \end{cases} \quad (16)$$

In Eq. (16), Y_j characterizes the j -th objects of the better universe, u_i^j signifies the j -th variables of the i -th universe, lbp_j specifies the lower limit at j th iterations. In contrast, ubp_j denotes the upper

limit at j th iterations and R_2 , R_3 , and R_4 represent arbitrary parameter ranges from zero and one, travelling distance rate (TDR) and wormhole existence probability (WEP) are constant variables, in which WEP is increased linearly via the iteration whereas TDR is minimized over the iteration. The adaptive equation for the coefficients is shown in the following:

$$WEP = WEP_{\min} + iter \times \left(\frac{WEP_{\max} - WEP_{\min}}{\text{Max}_{iter}} \right) \quad (17)$$

In Eq. (17), WEP_{\min} denotes the minimal values of WEP, WEP_{\max} indicates the maximal values of WEP, iter represents the current iteration, and max_{iter} shows the maximal iteration count.

$$TDR = 1 - \frac{iter^{AC}}{\text{Max}_{iter}^{AC}} \quad (18)$$

In Eq. (18), AC is separated by exploitation accuracy through iteration. The high value of AC is a very precise exploitation. Initially, the process initiates with initializing a collection of arbitrary universes (solution in the searching region) of the optimization issue. In every iteration, an object with a higher inflation rate in the universe moves towards the universe with a lower inflation rate.

4 Performance Validation

The proposed model is simulated using Python 3.6.5 tool. In this study, the effect analysis of the MVODRL-AA model on the Arabic language is examined on the SemEval2018-Ar Dataset comprising 10107 Arabic tweets under 11 class labels, as displayed in Table 1.

Fig. 3 illustrates the confusion matrix created by the MVODRL-AA model on the entire dataset. The figure implied that the MVODRL-AA model has properly identified 1689 samples under class 1, 392 samples under class 2, 815 samples under class 3, 754 samples under class 4, 1150 samples under class 5, 1060 samples under class 6, 1044 samples under class 7, 956 samples under class 8, 1594 samples under class 9, 56 samples under class 10, and 198 samples under class 11.

Table 1: Dataset details

Label	Class	No. of tweets
1	Anger	1723
2	Anticipation	424
3	Disgust	855
4	Fear	780
5	Joy	1177
6	Love	1104
7	Optimism	1074
8	Pessimism	1001
9	Sadness	1638
10	Surprise	98
11	Trust	233
Total number of tweets		10107

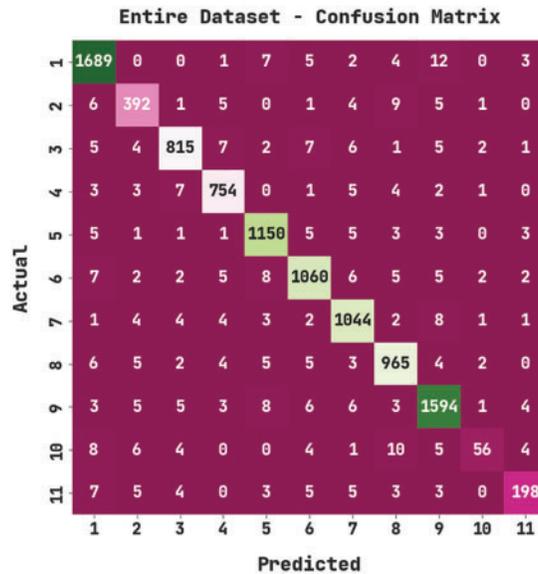


Figure 3: Confusion matrix of MVODRL-AA approach under the entire dataset

Table 2 and Fig. 4 indicate the affect analysis outcomes of the MVODRL-AA model on the entire dataset. The MVODRL-AA model has recognized Arabic tweets under class 1 with $accu_y$, $prec_n$, $reca_l$, F_{score} , and AUC_{score} of 99.16%, 97.07%, 98.03%, 97.55%, and 98.71%, respectively. Simultaneously, the MVODRL-AA model has identified Arabic tweets under class 5 with $accu_y$, $prec_n$, $reca_l$, F_{score} , and AUC_{score} of 99.38%, 96.96%, 97.71%, 97.33%, and 98.65% correspondingly. Concurrently, the MVODRL-AA technique has recognized Arabic tweets under class 9 with $accu_y$, $prec_n$, $reca_l$, F_{score} , and AUC_{score} of 99.05%, 96.84%, 97.31%, 97.08%, and 98.35% correspondingly. At last, the MVODRL-AA approach has recognized Arabic tweets under class 11 with $accu_y$, $prec_n$, $reca_l$, F_{score} , and AUC_{score} of 99.48%, 91.67%, 84.98%, 88.20%, and 92.40% correspondingly.

Table 2: Result analysis of MVODRL-AA approach with distinct class labels under the entire dataset

Entire dataset					
Labels	Accuracy	Precision	Recall	F-score	AUC score
1	99.16	97.07	98.03	97.55	98.71
2	99.34	91.80	92.45	92.13	96.05
3	99.31	96.45	95.32	95.88	97.50
4	99.45	96.17	96.67	96.42	98.17
5	99.38	96.96	97.71	97.33	98.65
6	99.16	96.28	96.01	96.15	97.78
7	99.28	96.04	97.21	96.62	98.37
8	99.21	95.64	96.40	96.02	97.96
9	99.05	96.84	97.31	97.08	98.35
10	99.49	84.85	57.14	68.29	78.52

(Continued)

Table 2 (continued)

Entire dataset					
Labels	Accuracy	Precision	Recall	F-score	AUC score
11	99.48	91.67	84.98	88.20	92.40
Average	99.30	94.53	91.75	92.88	95.68

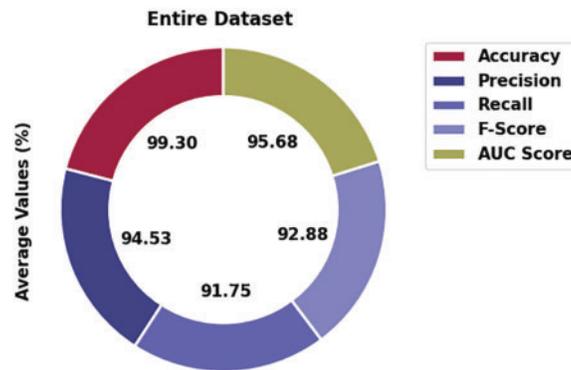


Figure 4: Average analysis of MVODRL-AA approach under the entire dataset

Fig. 5 demonstrates the confusion matrix created by the MVODRL-AA model on 70% of training (TR) data. The figure implied the MVODRL-AA technique has properly identified 1204 samples under class 1, 281 samples under class 2, 589 samples under class 3, 520 samples under class 4, 840 samples under class 5, 725 samples under class 6, 686 samples under class 7, 677 samples under class 8, 1118 samples under class 9, 39 samples under class 10, and 127 samples under class 11.

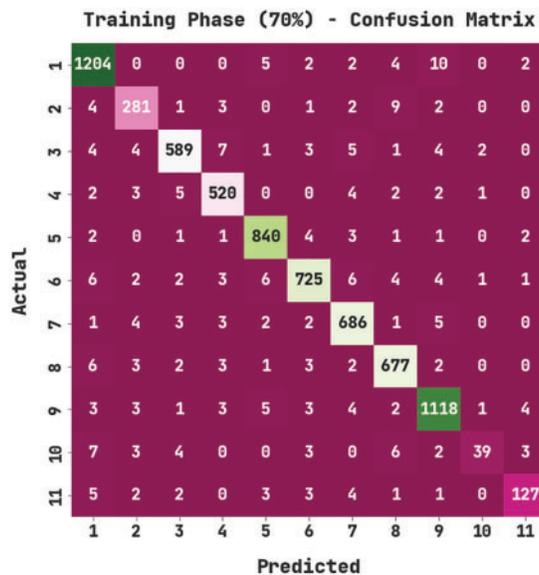


Figure 5: Confusion matrix of MVODRL-AA approach under 70% of TR data

Table 3 and Fig. 6 depict the affect analysis outcomes of the MVODRL-AA method on 70% of TR data. The MVODRL-AA model has recognized Arabic tweets under class 1 with $accu_y$, $prec_n$, $reca_l$, F_{score} , and AUC_{score} of 99.08%, 96.78%, 97.97%, 97.37%, and 98.64%, respectively. Concurrently, the MVODRL-AA model has recognized Arabic tweets under class 5 with $accu_y$, $prec_n$, $reca_l$, F_{score} , and AUC_{score} of 99.46%, 97.33%, 98.25%, 97.79%, and 98.94%, respectively. Simultaneously, the MVODRL-AA model has recognized Arabic tweets under class 9 with $accu_y$, $prec_n$, $reca_l$, F_{score} , and AUC_{score} of 99.12%, 97.13%, 97.47%, 97.30%, and 98.46%, respectively. Finally, the MVODRL-AA approach has recognized Arabic tweets under class 11 with $accu_y$, $prec_n$, $reca_l$, F_{score} , and AUC_{score} of 99.53%, 91.37%, 85.81%, 88.50%, and 92.82%, correspondingly.

Table 3: Result analysis of MVODRL-AA approach with distinct class labels under 70% of TR data

Labels	Training phase (70%)				
	Accuracy	Precision	Recall	F-score	AUC score
1	99.08	96.78	97.97	97.37	98.64
2	99.35	92.13	92.74	92.43	96.19
3	99.26	96.56	95.00	95.77	97.34
4	99.41	95.76	96.47	96.12	98.06
5	99.46	97.33	98.25	97.79	98.94
6	99.17	96.80	95.39	96.09	97.51
7	99.25	95.54	97.03	96.28	98.26
8	99.25	95.62	96.85	96.23	98.18
9	99.12	97.13	97.47	97.30	98.46
10	99.53	88.64	58.21	70.27	79.07
11	99.53	91.37	85.81	88.50	92.82
Average	99.31	94.88	91.93	93.11	95.77

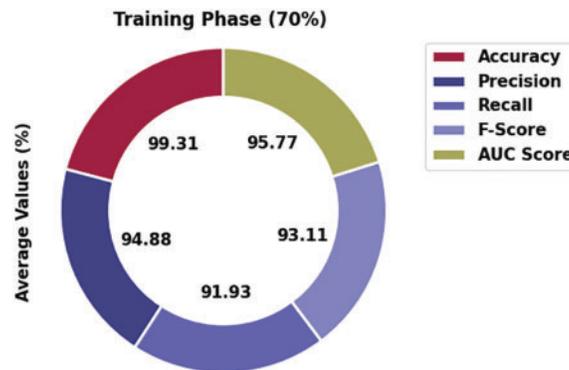


Figure 6: Average analysis of MVODRL-AA approach under 70% of TR data

Fig. 7 portrays the confusion matrix created by the MVODRL-AA method on 30% of testing (TS) data. The figure denoted the MVODRL-AA model has properly identified 485 samples under class 1, 111 samples under class 2, 226 samples under class 3, 234 samples under class 4, 310 samples under

class 5, 335 samples under class 6, 358 samples under class 7, 288 samples under class 8, 476 samples under class 9, 17 samples under class 10, and 71 samples under class 11.

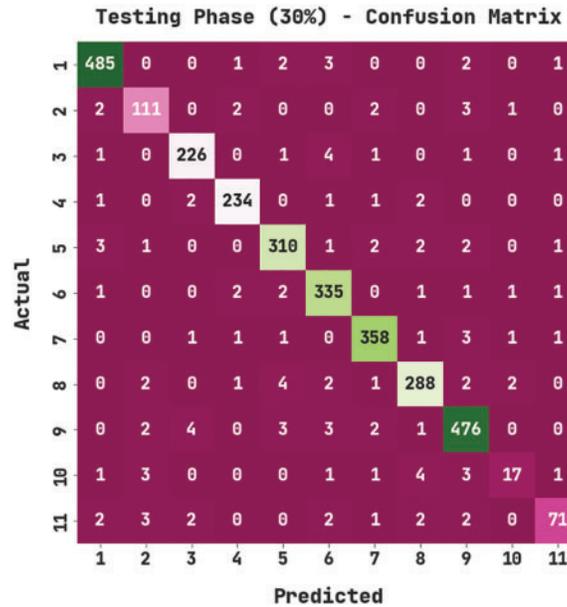


Figure 7: Confusion matrix of MVODRL-AA approach under 30% of TS data

Table 4 and Fig. 8 represent the affect analysis outcomes of the MVODRL-AA model on 30% of TS data. The MVODRL-AA model has recognized Arabic tweets under class 1 with $accu_y$, $prec_n$, $reca_l$, F_{score} , and AUC_{score} of 99.34%, 97.78%, 98.18%, 97.98%, and 98.87%, respectively. Simultaneously, the MVODRL-AA model has recognized Arabic tweets under class 5 with $accu_y$, $prec_n$, $reca_l$, F_{score} , and AUC_{score} of 99.18%, 95.98%, 96.27%, 96.12%, and 97.90% correspondingly. Concurrently, the MVODRL-AA method has recognized Arabic tweets under class 9 with $accu_y$, $prec_n$, $reca_l$, F_{score} , and AUC_{score} of 98.88%, 96.16%, 96.95%, 96.55%, and 98.10%, respectively. At last, the MVODRL-AA model has recognized Arabic tweets under class 11 with $accu_y$, $prec_n$, $reca_l$, F_{score} , and AUC_{score} of 99.34%, 92.21%, 83.53%, 87.65%, and 91.66% correspondingly.

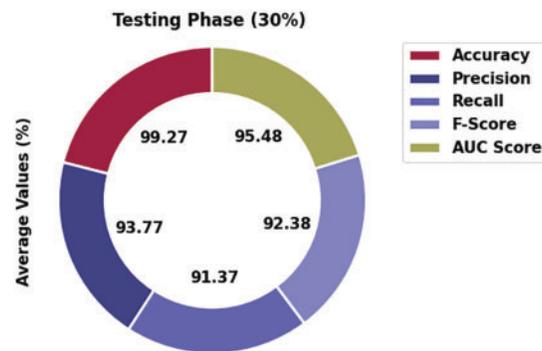
Table 4: Result analysis of MVODRL-AA approach with distinct class labels under 30% of TS data

Labels	Testing phase (30%)				
	Accuracy	Precision	Recall	F-score	AUC score
1	99.34	97.78	98.18	97.98	98.87
2	99.31	90.98	91.74	91.36	95.68
3	99.41	96.17	96.17	96.17	97.92
4	99.54	97.10	97.10	97.10	98.42
5	99.18	95.98	96.27	96.12	97.90
6	99.14	95.17	97.38	96.26	98.38
7	99.34	97.02	97.55	97.28	98.57
8	99.11	95.68	95.36	95.52	97.44
9	98.88	96.16	96.95	96.55	98.10

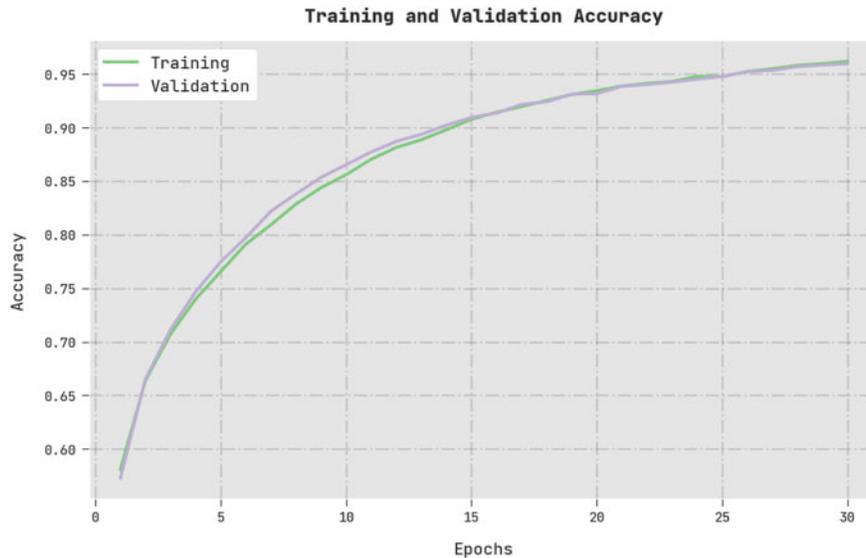
(Continued)

Table 4 (continued)

Labels	Testing phase (30%)				
	Accuracy	Precision	Recall	F-score	AUC score
10	99.37	77.27	54.84	64.15	77.34
11	99.34	92.21	83.53	87.65	91.66
Average	99.27	93.77	91.37	92.38	95.48

**Figure 8:** Average analysis of MVODRL-AA approach under 30% of TS data

The training accuracy (TA) and validation accuracy (VA) gained by the MVODRL-AA method on the test dataset is shown in Fig. 9. The experimental outcome denoted the MVODRL-AA technique has achieved maximal values of TA and VA. Particularly the VA is greater than TA.

**Figure 9:** TA and VA analysis of MVODRL-AA methodology

The training loss (TL) and validation loss (VL) attained by the MVODRL-AA approach on the test dataset are displayed in Fig. 10. The experimental outcome implied the MVODRL-AA algorithm had exhibited the least values of TL and VL. In specific, the VL is lesser than TL.

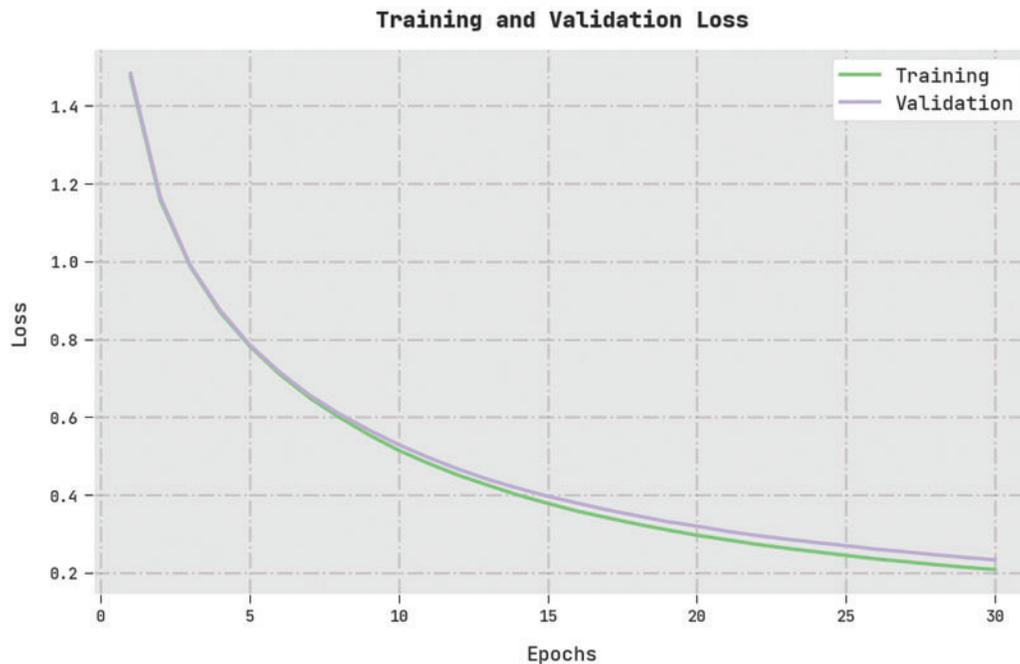


Figure 10: TL and VL analysis of MVODRL-AA methodology

A clear precision-recall analysis of the MVODRL-AA method on the test dataset is displayed in Fig. 11. The figure indicates the MVODRL-AA method has enhanced precision-recall values under all classes.

A brief ROC analysis of the MVODRL-AA method on the test dataset is illustrated in Fig. 12. The results indicated the MVODRL-AA approach had shown its ability to categorize distinct classes on the test dataset.

To ensure the improvements of the MVODRL-AA model over other models, a comparative study is given in Table 5 and Fig. 13 [22]. The outcomes implied the supremacy of the MVODRL-AA model over other models. Based on $accu_y$, the MVODRL-AA model has reached an increased $accu_y$ value of 99.27%. In contrast, the Bidirectional Encoder Representations from Transformers (BERT), Arabic BERT (AraBERT) v01-base, AraBERT v02-base, AraBERT v02-large, BiLSTM, and LeNet models have obtained reduced $accu_y$ of 93.15%, 92.84%, 86.68%, 93.42%, 90.73%, and 86.58% respectively. In addition, based on $prec_n$, the MVODRL-AA model has attained an increased $prec_n$ value of 93.77%. In contrast, the BERT, AraBERT v01-base, AraBERT v02-base, AraBERT v02-large, BiLSTM, and LeNet models have reduced $prec_n$ of 87.72%, 85.01%, 91.01%, 90.82%, 85.31%, and 87.07% correspondingly. Next to that, based on F_{score} , the MVODRL-AA model has acquired an increased F_{score} value of 92.38%, whereas the BERT, AraBERT v01-base, AraBERT v02-base, AraBERT v02-large, BiLSTM, and LeNet models have reached reduced F_{score} of 86.84%, 87.15%, 87.77%, 86.04%, 86.85%, and 87.36% correspondingly. The detailed results and discussion assume that the MVODRL-AA model has shown enhanced results over other models.

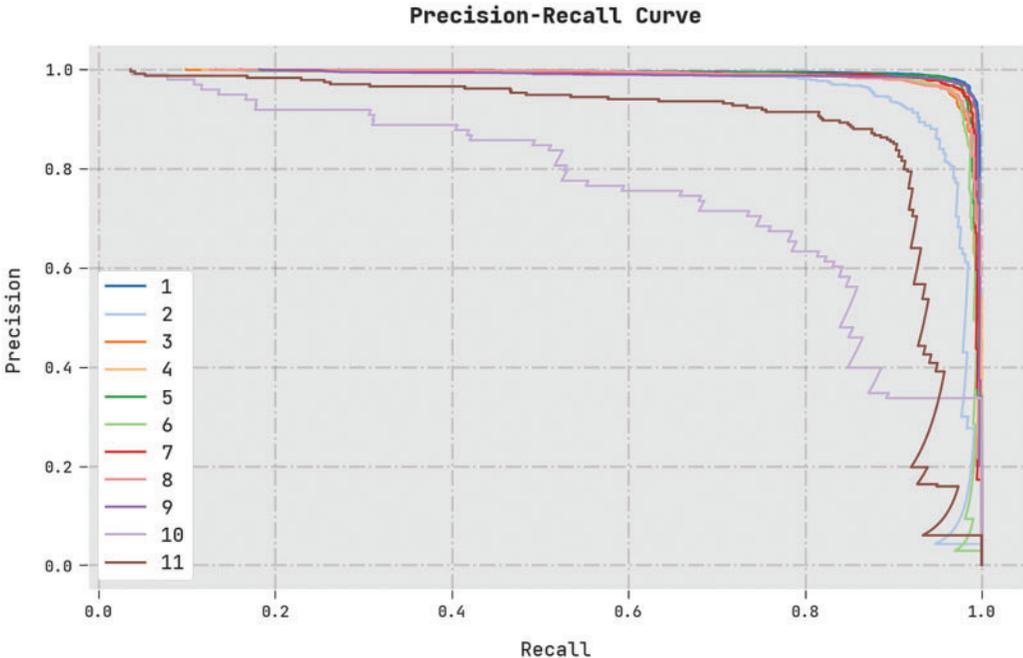


Figure 11: Precision-recall curve analysis of MVODRL-AA methodology

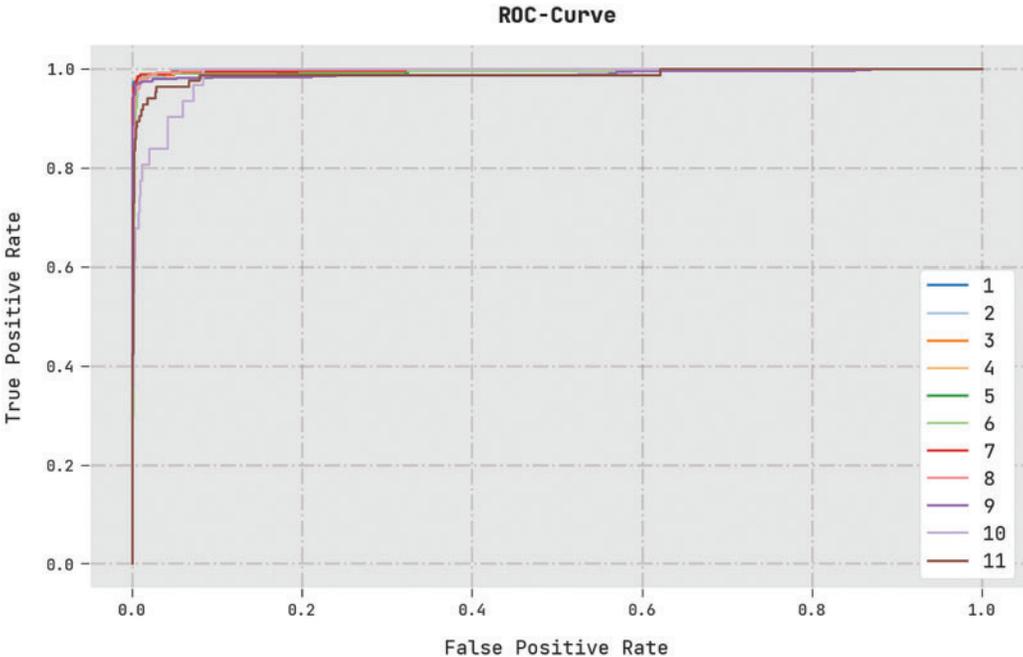


Figure 12: ROC curve analysis of MVODRL-AA methodology

Table 5: Comparative analysis of MVODRL-AA approach with existing methodologies

Methods	Accuracy	Precision	Recall	F-score
MVODRL-AA	99.27	93.77	91.37	92.38
BERT model	93.15	87.72	87.98	86.84
AraBERT v01-base	92.84	85.01	87.08	87.15
AraBERT v02-base	86.68	91.01	88.73	87.77
AraBERT v02-large	93.42	90.82	88.11	86.04
BiLSTM model	90.73	85.31	88.60	86.85
LeNet model	86.58	87.07	87.13	87.36

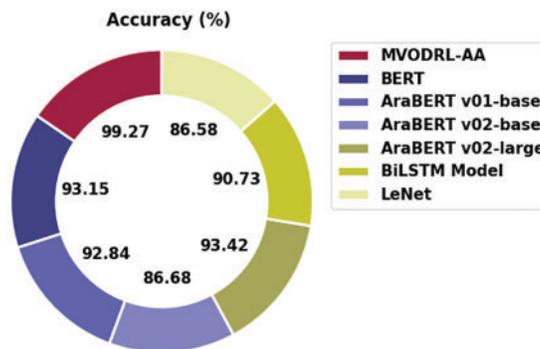


Figure 13: Comparative analysis of MVODRL-AA approach with existing methodologies

5 Conclusion

This paper introduces the MVODRL-AA technique to identify and classify effects or emotions in the Arabic corpus. Firstly, the MVODRL-AA model followed data pre-processing and the word embedding process. Next, the n-gram model is utilized to generate word embedding. The DQLN model is then exploited to identify and classify the effect on the Arabic corpus. At last, the MVO algorithm is used as a hyperparameter tuning approach to adjust the hyperparameters related to the DQLN model. A series of simulations were carried out to showcase the promising performance of the MVODRL-AA model. The simulation outcomes signify the betterment of the MVODRL-AA method over other approaches. In the future, the outcomes of the MVODRL-AA model can be extended to the use of hybrid metaheuristics. Besides, the proposed model can be employed in the fake news detection process.

Acknowledgement: None.

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: 22UQU4340237DSR38.

Author Contributions: The manuscript was written through contributions of all authors. All authors have given approval to the final version of the manuscript. All authors have read and agreed to the final version of the manuscript.

Availability of Data and Materials: Data sharing not applicable to this article as no datasets were generated during the current study.

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

References

- [1] 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.
- [2] E. A. H. Khalil, E. M. F. E. Houbay and H. K. Mohamed, "Deep learning for emotion analysis in Arabic tweets," *Journal of Big Data*, vol. 8, no. 1, pp. 136, 2021.
- [3] N. Al-Twairish, "The evolution of language models applied to emotion analysis of Arabic tweets," *Information*, vol. 12, no. 2, pp. 84, 2021.
- [4] M. Hegazi, Y. Al-Dossari, A. Al-Yahy, A. Al-Sumari and A. Hilal, "Preprocessing arabic text on social media," *Heliyon*, vol. 7, no. 2, pp. e06191, 2021.
- [5] A. Alharbi, M. Taileb and M. Kalkatawi, "Deep learning in arabic sentiment analysis: An overview," *Journal of Information Science*, vol. 47, no. 1, pp. 129–140, 2021.
- [6] A. Al-Laith and M. Alenezi, "Monitoring people's emotions and symptoms from arabic tweets during the COVID-19 pandemic," *Information*, vol. 12, no. 2, pp. 86, 2021.
- [7] 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.
- [8] M. M. Fouad, A. Mahany, N. Aljohani, R. A. Abbasi and S. U. Hassan, "Arwordvec: Efficient word embedding models for Arabic tweets," *Soft Computing*, vol. 24, no. 11, pp. 8061–8068, 2020.
- [9] A. Yadav and D. K. Vishwakarma, "Sentiment analysis using deep learning architectures: A review," *Artificial Intelligence Review*, vol. 53, no. 6, pp. 4335–4385, 2020.
- [10] 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.
- [11] M. A. AlGhamdi and M. A. Khan, "Intelligent analysis of Arabic tweets for detection of suspicious messages," *Arabian Journal for Science and Engineering*, vol. 45, no. 8, pp. 6021–6032, 2020.
- [12] A. Alsayat and N. Elmitwally, "A comprehensive study for arabic sentiment analysis (challenges and applications)," *Egyptian Informatics Journal*, vol. 21, no. 1, pp. 7–12, 2020.
- [13] A. Alharbi, M. Kalkatawi and M. Taileb, "Arabic sentiment analysis using deep learning and ensemble methods," *Arabian Journal for Science and Engineering*, vol. 46, no. 9, pp. 8913–8923, 2021.
- [14] A. H. Ombabi, W. Ouarda and A. M. Alimi, "Deep learning CNN–LSTM framework for Arabic sentiment analysis using textual information shared in social networks," *Social Network Analysis and Mining*, vol. 10, no. 1, pp. 53, 2020.
- [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. Saleh, S. Mostafa, A. Alharbi, S. El-Sappagh and T. Alkhalifah, "Heterogeneous ensemble deep learning model for enhanced arabic sentiment analysis," *Sensors*, vol. 22, no. 10, pp. 3707, 2022.
- [17] A. Al-Hashedi, B. Al-Fuhaidi, A. M. Mohsen, Y. Ali, H. A. Gamal Al-Kaf *et al.* "Ensemble classifiers for arabic sentiment analysis of social network (Twitter data) towards COVID-19-related conspiracy theories," *Applied Computational Intelligence and Soft Computing*, vol. 2022, pp. 1–10, 2022.

- [18] E. Omara, M. Mousa and N. Ismail, "Character gated recurrent neural networks for arabic sentiment analysis," *Scientific Reports*, vol. 12, no. 1, pp. 9779, 2022.
- [19] M. Ghiassi, J. Skinner and D. Zimbra, "Twitter brand sentiment analysis: A hybrid system using n-gram analysis and dynamic artificial neural network," *Expert Systems with Applications*, vol. 40, no. 16, pp. 6266–6282, 2013.
- [20] X. Wang, Y. Wang, D. Shi, J. Wang and Z. Wang, "Two-stage wecc composite load modeling: A double deep Q-learning networks approach," *IEEE Transactions on Smart Grid*, vol. 11, no. 5, pp. 4331–4344, 2020.
- [21] M. H. Lokman, I. Musirin, S. I. Suliman, H. Suyono, R. N. Hasanah *et al.*, "Multi-verse optimization based evolutionary programming technique for power scheduling in loss minimization scheme," *IAES International Journal of Artificial Intelligence*, vol. 8, no. 3, pp. 292, 2019.
- [22] H. Elfaik and E. H. Nfaoui, "Combining context-aware embeddings and an attentional deep learning model for arabic affect analysis on twitter," *IEEE Access*, vol. 9, pp. 111214–111230, 2021.