Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2023.033531
Article

Quantum Particle Swarm Optimization with Deep Learning-Based Arabic Tweets Sentiment Analysis

Badriyya B. Al-onazi[1], Abdulkhaleq Q. A. Hassan[2], Mohamed K. Nour[3], Mesfer Al Duhayyim[4,*], Abdullah Mohamed[5], Amgad Atta Abdelmageed[6], Ishfaq Yaseen[6] and Gouse Pasha Mohammed[6]

[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 English, College of Science and Arts at Mahayil, King Khalid University, Abha, 62217, Saudi Arabia
[3]Department of Computer Sciences, College of Computing and Information System, Umm Al-Qura University, Makkah, 24231, Saudi Arabia
[4]Department of Computer Science, College of Sciences and Humanities-Aflaj, Prince Sattam bin Abdulaziz University, Al-Aflaj, 16828, Saudi Arabia
[5]Research Centre, Future University in Egypt, New Cairo, 11845, Egypt
[6]Department of Computer and Self Development, Preparatory Year Deanship, Prince Sattam bin Abdulaziz University, AlKharj, 16436, Saudi Arabia
*Corresponding Author: Mesfer Al Duhayyim. Email: m.alduhayyim@psau.edu.sa
Received: 19 June 2022; Accepted: 13 October 2022

Abstract: Sentiment Analysis (SA), a Machine Learning (ML) technique, is often applied in the literature. The SA technique is specifically applied to the data collected from social media sites. The research studies conducted earlier upon the SA of the tweets were mostly aimed at automating the feature extraction process. In this background, the current study introduces a novel method called Quantum Particle Swarm Optimization with Deep Learning-Based Sentiment Analysis on Arabic Tweets (QPSODL-SAAT). The presented QPSODL-SAAT model determines and classifies the sentiments of the tweets written in Arabic. Initially, the data pre-processing is performed to convert the raw tweets into a useful format. Then, the word2vec model is applied to generate the feature vectors. The Bidirectional Gated Recurrent Unit (BiGRU) classifier is utilized to identify and classify the sentiments. Finally, the QPSO algorithm is exploited for the optimal fine-tuning of the hyperparameters involved in the BiGRU model. The proposed QPSODL-SAAT model was experimentally validated using the standard datasets. An extensive comparative analysis was conducted, and the proposed model achieved a maximum accuracy of 98.35%. The outcomes confirmed the supremacy of the proposed QPSODL-SAAT model over the rest of the approaches, such as the Surface Features (SF), Generic Embeddings (GE), Arabic Sentiment Embeddings constructed using the Hybrid (ASEH) model and the Bidirectional Encoder Representations from Transformers (BERT) model.

Keywords: Sentiment analysis; Arabic tweets; quantum particle swarm optimization; deep learning; word embedding

1 Introduction

Arabic is the mother tongue of 300 million people across 22 nations. The language is predominantly spoken by 1.4 Billion global Muslims [1]. The Arabic language has 28 alphabets without upper or lower cases. The writing orientation of the language is from right to left. The letters are written in distinct shapes that vary slightly from place to place. Despite the predominant usage of Arabic on the Internet, Arabic language-focused Sentiment Analysis (SA) studies are rare. Various reasons are attributed to the absence of reliable research in this domain. Amongst these, the study conducted by Assiri stated two chief factors as the limited research fund and the complicated morphology of the Arabic language than the geomorphology of other languages [2,3]. The intricacy and the diversity of Arabic vernaculars demand progressive pre-processing and lexicon building processes. The wealth of user-generated content on social networking sites is yet to be reaped since the internet contains huge volumes of shapeless data. Such data includes valuable opinions and sentiments for both organizations and individuals [4,5]. SA has been defined as "the domain of study which examines emotions, opinions, attitudes, sentiments and appraisals of people against objects and its qualities uttered in the written text". The primary objective of SA is to distinguish the text as either neutral or positive/negative. However, the SA of the tweets made in Arabic is a complicated task, owing to the rich geomorphology of the language and the casual nature of the linguistics in Twitter [6]. The SA methods include supervised learning approaches, Machine Learning (ML) techniques with unsupervised learning and feature engineering methods that exploit the rule-based techniques and sentiment lexicons [7,8]. The leading approaches that utilize ML methods depend on the physical abstraction of the structures utilized in the classification process. However, the physical abstraction of the structures is not only labour-intensive but also takes much time to accomplish. Such manually-abstracted features are called 'surface features' [9,10].

Several SA approaches have been developed for the English language. However, the efforts taken to conduct SA on the Arabic language texts are less due to multiple challenges such as the rich language, need to produce results that should not get affected due to dialects and the demand for high accuracy [11]. In this background, a dynamic device should be developed so as to empower the Arabic language speakers in terms of mass media analysis and visualization of the overall sentiments regarding hot themes. Since the Arabic language contains numerous vernaculars, it is assumed most of the time. Further, every dialect sense of the words is completely dissimilar. Arabic is a morphologically-ironic language that possibly increases the issues for any automatic text analysis tool [12]. The current study considered the data from the individuals who utilize the social media platform, i.e., Twitter, to interact with others, comment on issues, etc. Twitter is a relaxed channel in which bloggers are free to use informal language for tweeting [13]. Every tweet is restricted to a maximum length of 140 characters, which makes it difficult to detect the sentiments from both informal and short comments.

In literature [14], a corpus-based system was developed for SA of the tweets made in the Arabic language to classify the data as either negative or positive. This technique was developed on the basis of the Discriminatory Multinomial Naïve Bayes (DMNB) system using the N-grams tokenizer, stemming, and the Term Frequency-Inverse Document Frequency (TF-IDF) systems. The study conducted earlier [15] presented a novel technique for detecting the Influenza-utilizing ML approaches in Arabic tweets from Arab countries. This study conducted a primary analysis of the tweets made in the Arabic language about epidemic diseases. In this work, the tweets were gathered, labelled and filtered to find and classify the influenza-related tweets made in the Arabic language. Various classification techniques were utilized for comparison; namely, the Naïve Bayes (NB) method, Support Vector Machine (SVM), Decision Tree (DT) and the k-Nearest Neighbor (KNN), to establish the performance of the proposed technique and the model's supreme quality.

Elshakankery et al. [16] proposed a semi-automatic learning model for SA in which the lexicon is updated with language changes. A hybrid technique i.e., the Hybrid Incremental Learning Approach for Arabic Tweets SA (HILATSA), was proposed in this study by integrating the lexicon-based techniques and the ML techniques to identify the sentiment polarity of the tweets. In the study conducted earlier [17], an opinion target extraction model was presented for Arabic tweets. During the pre-processing stage, many feature procedures were extracted from the tweets and studied in detail. The purpose of this task i.e., the feature extraction process, was to evaluate its influence on the accuracy of the outcomes. Next, two classifiers such as SVM and NB, were trained. According to the researchers [18], the research on Arabic SA has increased tremendously in recent years across the globe, unlike English language-related SA studies [18]. The impact of this work is two-fold; primarily, it can establish a corpus of 40,000 labelled Arabic tweets spanning different topics. The secondary impact is the proposal of three Deep Learning (DL) approaches for Arabic SA. With the help of the word embedding process, the performance of all three approaches can be validated on the presented corpus.

The current study introduces a new technique named Quantum Particle Swarm Optimization with Deep Learning-Based Sentiment Analysis on Arabic Tweets (QPSODL-SAAT). The presented QPSODL-SAAT model detects and classifies the sentiments expressed in tweets made in the Arabic language. Initially, the data pre-processing is performed to convert the raw tweets into a useful format. Then, the word2vec model is applied to generate the feature vectors. The Bidirectional Gated Recurrent Unit (BiGRU) classifier is utilized to identify and classify the sentiments. Finally, the QPSO algorithm is exploited for optimal fine-tuning of the hyperparameters involved in the BiGRU model. The proposed QPSODL-SAAT model was experimentally validated using a standard dataset.

2 The Proposed Model

In this study, a new QPSODL-SAAT model is devised to recognise and classify the sentiments found in the Arabic language tweets. Initially, the data pre-processing is performed to convert the raw tweets into a useful format. Then, the word2vec model is applied to generate the feature vectors. To identify and classify the sentiments, the BiGRU classifier is utilized. Finally, the QPSO algorithm is exploited for optimal fine-tuning of the hyperparameters involved in the BiGRU model. Fig. 1 shows the working process of the proposed QPSODL-SAAT technique.

2.1 Data Pre-Processing

Initially, the data is pre-processed and cleaned to generate the input Twitter invaluable form. Twitter collects text from human beings in the form of an unstructured language that contains orthographic mistakes, abbreviations and slang words. It is essential to transform this data through a structured architecture by implementing pre-processing approaches. This step is crucial so that the ML methodology can analyse the text and achieve consistent outcomes with maximum performance. The sentiment analysis of the tweets made in the Arabic language brings additional challenges than the sentiment analysis of the tweets made in other languages. Arabic natural language has a shortfall of powerful resources and tools that assist in extracting Arabic sentiments from the text [19]. The succeeding steps detail the pre-processing data phases while it is implemented for two data sets in the current study.

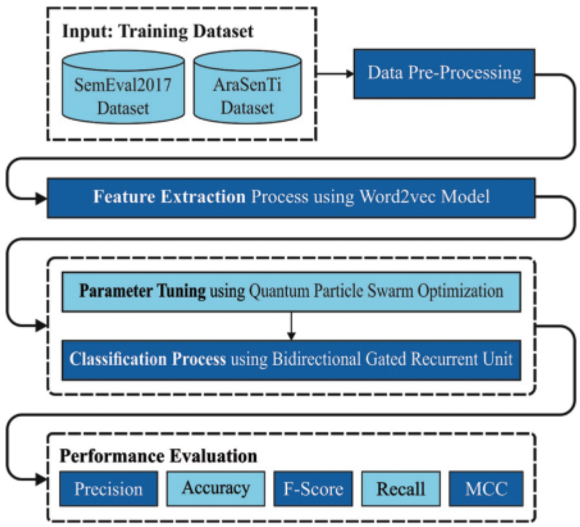


Figure 1: Overall process of the QPSODL-SAAT approach

At first, the unrelated tweets that contain ads and are not compatible with distance-learning subjects from the Kingdom of Saudi Arabia are eliminated manually. This action reduces the number of tweets in the primary data set to 5,096. In the subsequent dataset, the number of tweets further gets reduced to 9,160 tweets. The following steps were followed in this study during the pre-processing stage.

- Elimination of the non-Arabic letters.
- Elimination of the symbols that demonstrate the emotions, emoticons, hashtag signs, numbers and symbols.
- Elimination of the user mentions and the Uniform Resource Locators (URL).
- Elimination of Tashkeel employs '-' symbol to increase the length of certain characters.
- Elimination of the punctuation marks.
- Elimination of the repetitive characters.
- Elimination of the stop-words: The extracted stop-words were a part of a representative group of Arabic stop-words in Python's Natural Language Toolkit (NLTK) library.
- Execution of the Arabic language standardization.
- Execution of the word stemming process by exploiting the Information Science Research Institute's (ISRI) stemmer to decrease the Arabic language words for the word stemming process.
- Execution of the tokenization process that separates the texts for less token or piece.

2.2 Word Embeddings

After the data is pre-processed, the word2vec model is applied to generate the feature vectors. Word embedding is a collection of language feature learning mechanisms in the Natural Language Processing (NLP) methods. It translates the word 'tokens' into machine-readable vectors. *Word2vec* is a two-layer neural network that transforms a text word into a vector. The output is a set of vectors, and the input is a text corpus. *word2vec* has a valuable characteristic, i.e., it can train a large-scale corpus to generate the low-dimension word vectors. Assume a sentence that is composed of n words $(x,\ x_2, x_3,\ \ldots,\ x_{n-2},\ x_{n-1},\ x_n)$ while the word x_i can be transformed to a real value vector, e_i as demonstrated herewith.

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

In Eq. (1), d represents the size of the embedded word, whereas w indicates a word.

2.3 BiGRU-Based Sentiment Classification

In this study, the BiGRU model is exploited to detect and classify the sentiments. To compensate the inadequacy of the Convolutional Neural Network (CNN) in apprehending the time-based data of the signals, the Bi-directional GRU is exploited for explicitly modelling the temporal dependencies. Gated Recurrent Unit (GRU) [20] is a variant of the Recurrent Neural Network (RNN) model that works on parameter length series $[x_1,\ x_2, \ldots,\ x_T]$ to learn about the likelihood of the series. During every t time step, it obtains x_t and the preceding layer h_{t-1} as inputs to produce a hidden layer, h_t.

$$h_t = f\,(h_{t-1},\ x_t) \tag{2}$$

$$r_t = \sigma\,(W_r\,[h_{t-1},\ x_t]) \tag{3}$$

$$z_t = \sigma\left(W_z\left[h_{t-1,x_t}\right]\right) \tag{4}$$

$$\widetilde{h}_t = tanh\,(W_{\tilde{\mathrm{h}}}\,[r_t \odot h_{t-1}, x_t]) \tag{5}$$

$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot \widetilde{h}_t \tag{6}$$

In this expression, f indicates a nonlinear activation that is executed by r_t and z_t i.e., reset and update gates, respectively. W_r, W_z and $W_{\tilde{\mathrm{h}}}$ are the variables for respective gates, whereas σ indicates a sigmoid function. $\odot$ represents a Hadamard product. At every t time, r_t and z_t are calculated by the input x_t and the preceding layer, h_{t-1}. Both update and the reset gates are sensitive to long- and short-term dependencies correspondingly in such a way that the GRU method is capable of capturing the dependency over distinct scales. Fig. 2 depicts the infrastructure of the GRU model.

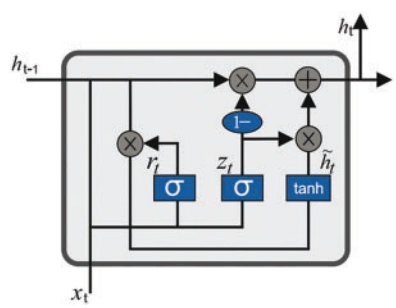


Figure 2: Structure of the GRU model

In the HybridNet approach, the features generated through the extractor are transmitted to the Bi-directional GRU to capture the temporal dependency. Every time, the Bi-GRU model generates two mechanisms in which the forward element is computed from the backward layer, whereas the front-to-end components are in conflict. Every input to the Bi-GRU model has a similar receptive field which might be correspondingly significant. Afterwards, the GRU calculates the component accesses to large fields that are highly instructive. Both backward and forward mechanisms pass over an unshared classification i.e., two Fully-Connected (FC) layers, to predict the outcomes. Next, two outcomes are added together, and the class probability is computed with the help of a *softmax* function.

2.4 QPSO Based Hyperparameter Tuning

At last, the QPSO algorithm is exploited for optimal fine-tuning of the hyperparameters involved in the BiGRU model. Based on the trajectory scrutiny of the Particle Swarm Optimization (PSO) approach, Sun et al. proposed QPSO as a variant of the PSO model. This QPSO approach outperforms the traditional PSO in terms of search ability [21]. QPSO sets a targeted point for every particle and is represented by $G_i = \left(g_{i1},\ g_{i2},\ \ldots,\ g_{iq}\right)$ as the targeted point, whereas the coordinate is as follows.

$$g_{ij} = (\varnothing_{ij} p_{ij} + \left(1 - \varphi_{ij}\right) p_{gj} \tag{7}$$

In this expression, β_{ij} denotes an arbitrary number that lies in the range of 0 and 1. The trajectory analysis outcomes demonstrate that G_i is the local attractor of the particles; In other terms, in the PSO approach, a particle i converges towards the center. The position of the *i-th* particle is upgraded using the formula given below.

$$x_{ij}^{k+1} = g_{ij}{}^{k} \pm \frac{L_{ij}^{k}}{2} \ln \left(\frac{1}{u}\right) \tag{8}$$

$$L_{ij}^{k} = 2\alpha \left|c_j - x_{ij}^{k}\right|,$$

In Eq. (8), u denotes an arbitrary number that lies in between 0 and 1. $C = \left[c_1,\ c_1,\ \ldots,\ c_q\right]$ indicates the mean optimum position, which is otherwise described as an average of the personal optimum position of every particle. So, the following equation is applied.

$$c_j = \frac{1}{n}\sum\nolimits_{t=1}^{n} p_{tj},\ j = 1, 2,\ \ldots, q \tag{9}$$

The α parameter corresponds to a Contraction-Expansion Coefficient and is fine-tuned to control the convergence rate. Since the iteration of the QPSO technique is different from the iteration of the PSO technique, the Binary PSO (BPSO) process cannot be applied. Further, the same technique cannot be employed by QPSO too. In the BQPSO technique, $X_i = \left(x_{i1},\ x_{i2},\ \ldots,\ x_{iq}\right)$ indicates the position of the *i-th* particle. However, it is important to highlight that X_i indicates a binary string rather than a vector. But, x_{ij} denotes the substring of X_i, not the *j-th* bit in the binary strings. Given that the length of every substring is l, the length of X_i can be expressed as lq. The target point G_i for the *i-th* particle is generated by a crossover operation i.e., BQPSO. It exerts a crossover function on P_i personal optimum position and the P_g global optimum position to produce two offspring binary strings. Here, G_i refers to a value that is arbitrarily selected between them.

$$p_m = \alpha * d_H\left(c_j,\ x_{ij}^{k}\right) * \ \ln \left(\frac{1}{u}\right),\ u \sim U[0, 1] \tag{10}$$

In Eq. (10), k denotes the iteration count, and $d_H\left(c_j,\ x_{ij}^{k}\right)$ characterizes the Hamming distance between c_j and x_{ij}^{k}. In contrast to two-bit strings, the Hamming distance denotes the quantity of a bit variance in two-strings. c_j indicates the *j-th* substring of the mean optimum position, and c_j is determined by the state of the *d-th* bit of optimum location of every particle. If more particles take on 1 at the *d-th* bit, then the *d-th* bit of c_j is one; otherwise, the bit is zero. For each bit of g_{ij}, after

$p_m > rand$ implements the operation, if the state of bit becomes 1, the state is set at 0; or else the state is set to 0.

Algorithm 1: Pseudocode of the PSO algorithm

For every particle
Initialization of particle
END
DO
For every particle
Compute fitness value
If the fitness value is superior to the fitness value (pBest) in history, then fix existing value as the new pBest
End
Select the particle with fitness value of all the particles as gBest for every particle
Compute particle velocity
Upgrade particle position
End
Whereas maximal iterations or minimal error conditions are not achieved

3 Results and Discussion

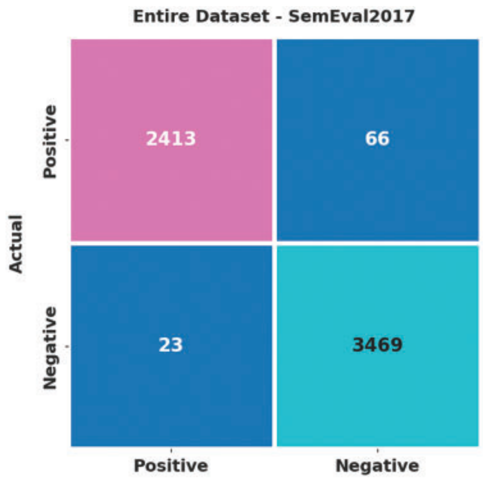
This section provides the results achieved from the SA process conducted on Arabic tweets sourced from SemEval2017 [22] and AraSenTi [23] datasets. Table 1 shows the details about both datasets used in this study.

Table 1: Dataset details

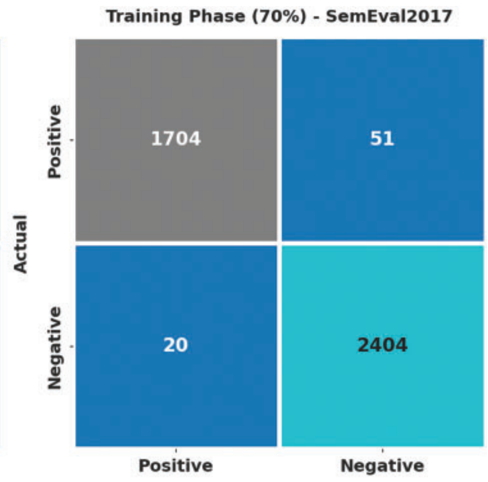
Class	Dataset	
	SemEval2017	AraSenTi
Positive	2479	4957
Negative	3492	6155
Total no. of samples	5971	11112

Fig. 3 shows the confusion matrices generated by the proposed QPSODL-SAAT model on the SemEval2017 dataset. On the entire dataset, the proposed QPSODL-SAAT model classified 2,413 samples under positive class and 3,469 samples under negative class. Moreover, on 70% of the Training (TR) dataset, the presented QPSODL-SAAT method recognized 1,704 samples as positive class and 2,404 samples as negative class. Meanwhile, on 30% of the Testing (TS) data, the proposed QPSODL-SAAT approach categorized 709 samples under positive class and 1,065 samples under negative class.

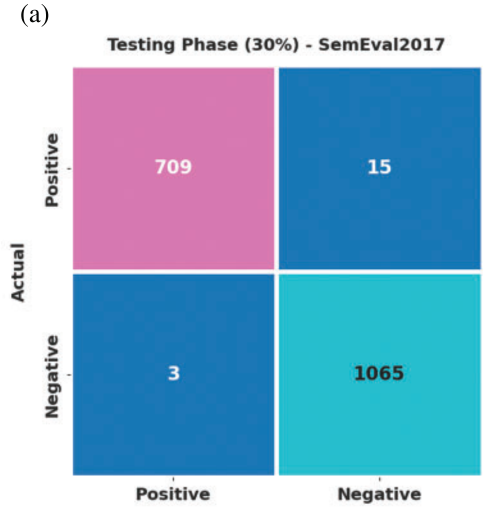
Table 2 and Fig. 4 report the overall SA outcomes accomplished by the proposed QPSODL-SAAT model on the SemEval2017 dataset. On the entire dataset, the presented QPSODL-SAAT model reached an average $accu_y$ of 98.51%, $prec_n$ of 98.59%, $reca_l$ of 98.34%, F_{score} of 98.46% and an MCC of 96.93%. Also, with 70% of the TR data, the proposed QPSODL-SAAT method attained an average $accu_y$ of 98.30%, $prec_n$ of 98.38%, $reca_l$ of 98.13%, F_{score} of 98.25% and an MCC of 96.52%. In addition to these, with 30% of the TS data, the presented QPSODL-SAAT approach reached an average $accu_y$ of 99%, $prec_n$ of 99.09%, $reca_l$ of 98.82%, F_{score} of 98.95% and an MCC of 97.92%.



(a)



(b)



(c)

Figure 3: Confusion matrices of the QPSODL-SAAT approach using SemEval2017 dataset (a) Entire dataset, (b) 70% of the TR data and (c) 30% of the TS data

Table 2: Analytical results of the QPSODL-SAAT approach on SemEval2017 dataset

SemEval2017 dataset					
Labels	Accuracy	Precision	Recall	F-score	MCC
Entire dataset					
Positive	98.51	99.06	97.34	98.19	96.93

(Continued)

Table 2: Continued

SemEval2017 dataset					
Labels	Accuracy	Precision	Recall	F-score	MCC
Negative	98.51	98.13	99.34	98.73	96.93
Average	98.51	98.59	98.34	98.46	96.93
Training phase (70%)					
Positive	98.30	98.84	97.09	97.96	96.52
Negative	98.30	97.92	99.17	98.54	96.52
Average	98.30	98.38	98.13	98.25	96.52
Testing phase (30%)					
Positive	99.00	99.58	97.93	98.75	97.92
Negative	99.00	98.61	99.72	99.16	97.92
Average	99.00	99.09	98.82	98.95	97.92

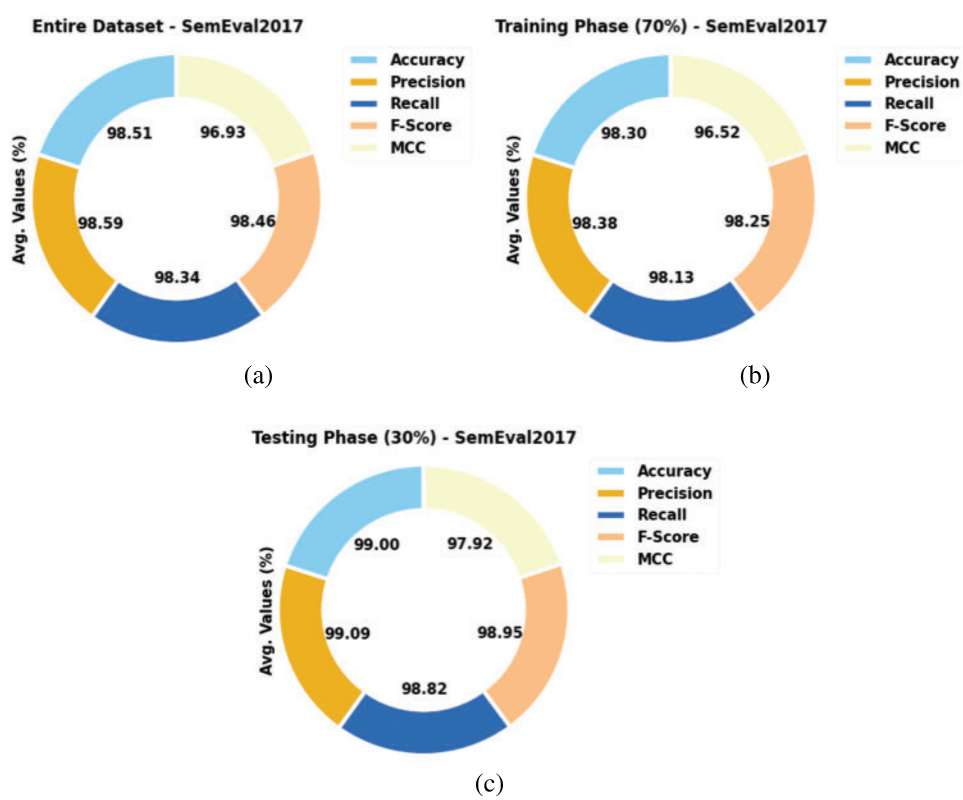


Figure 4: Average analysis results of the QPSODL-SAAT approach using SemEval2017 dataset (a) Entire dataset, (b) 70% of the TR data and (c) 30% of the TS data

Both Training Accuracy (TA) and Validation Accuracy (VA) values, obtained using the proposed QPSODL-SAAT system on SemEval2017 data, are depicted in Fig. 5. The experimental results infer that the proposed QPSODL-SAAT method accomplished the maximal TA and VA values whereas the VA values were higher than the TA values.

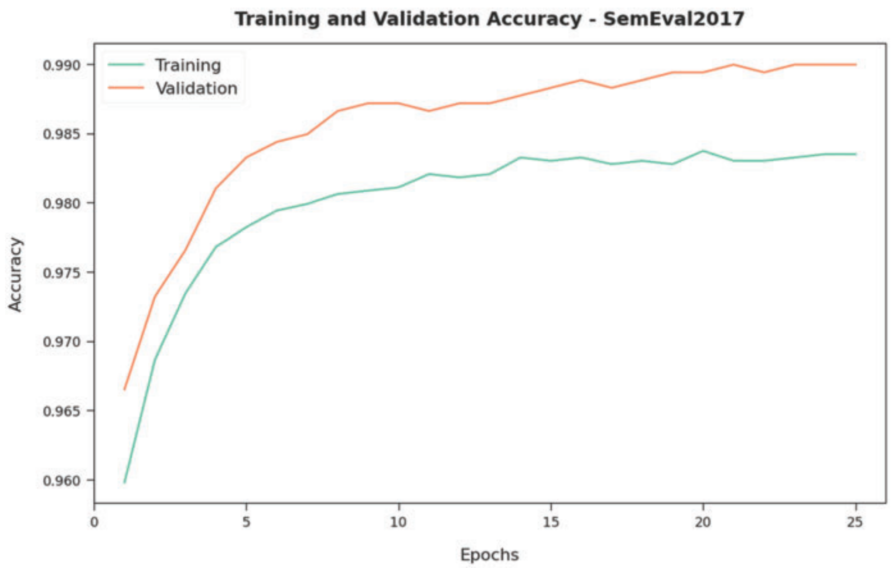


Figure 5: TA and VA analyses results of the QPSODL-SAAT approach on SemEval2017 dataset

Both Training Loss (TL) and Validation Loss (VL) values, achieved by the proposed QPSODL-SAAT approach on SemEval2017 data, are illustrated in Fig. 6. The experimental results illustrate that the presented QPSODL-SAAT approach obtained the minimal TL and VL values whereas the VL values were lesser than the TL values.

Figure 6: TL and VL analyses results of the QPSODL-SAAT approach on SemEval2017 dataset

Table 3 and Fig. 7 exhibit the comparison study results achieved by the proposed QPSODL-SAAT method and other existing models on SemEval2017 dataset. The obtained results establish that the QPSODL-SAAT model outperformed other models. With respect to $accu_y$, the proposed QPSODL-SAAT model achieved a maximum $accu_y$ of 99%, whereas the SF, GE, ASEH and BERT models attained low $accu_y$ values such as 90.74%, 92.56%, 94.07% and 92.46% respectively. Meanwhile, with respect to $reca_l$, the proposed QPSODL-SAAT method achieved a maximum $reca_l$ of 98.82%, while the rest of the techniques, such as the SF, GE, ASEH and the BERT, accomplished low $reca_l$ values such as 88.45%, 92.38%, 94.63% and 93.03% correspondingly. With regard to F_{score}, the proposed QPSODL-SAAT method accomplished a maximum F_{score} of 98.95%, whereas the SF, GE, ASEH and the BERT approaches attained low F_{score} values such as 88.57%, 93.09%, 94.26% and 96.66% correspondingly.

Table 3: Comparative analysis results of the QPSODL-SAAT approach and other existing methodologies on SemEval2017 dataset

SemEval2017 dataset				
Methods	Accuracy	Precision	Recall	F-score
QPSODL-SAAT	99.00	99.09	98.82	98.95
SF model	90.74	90.65	88.45	88.57
GE model	92.56	90.30	92.38	93.09
ASEH model	94.07	94.27	94.63	94.26
BERT model	92.46	95.27	93.03	96.66

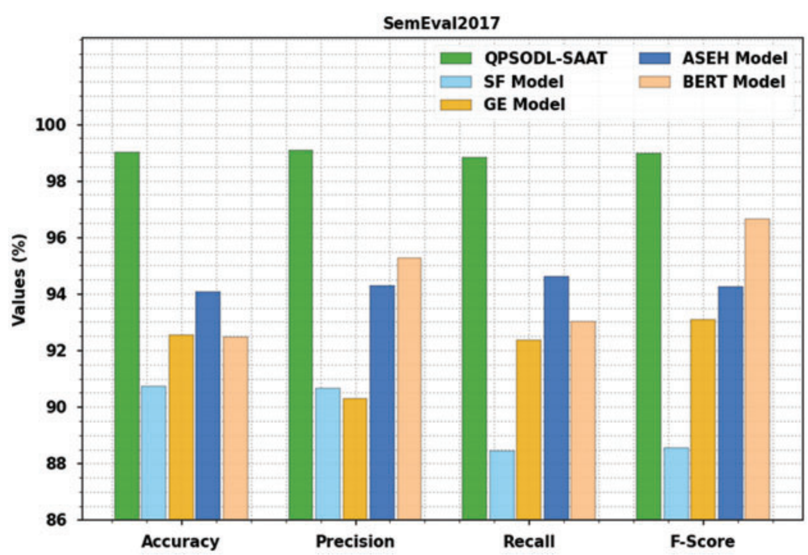


Figure 7: Comparative analysis outcomes of the QPSODL-SAAT approach on the SemEval2017 dataset

Fig. 8 shows the confusion matrices generated by the proposed QPSODL-SAAT method on AraSenTi dataset. On whole dataset, the proposed QPSODL-SAAT approach recognized 4,863 samples as positive class and 6,073 samples as negative class. Furthermore, on 70% of the TR data, the proposed QPSODL-SAAT method classified 3,400 samples under positive class and 4,256 samples under negative class. Meanwhile, on 30% of the TS data, the presented QPSODL-SAAT system recognized 1,463 samples as positive class and 1,816 samples as negative class.

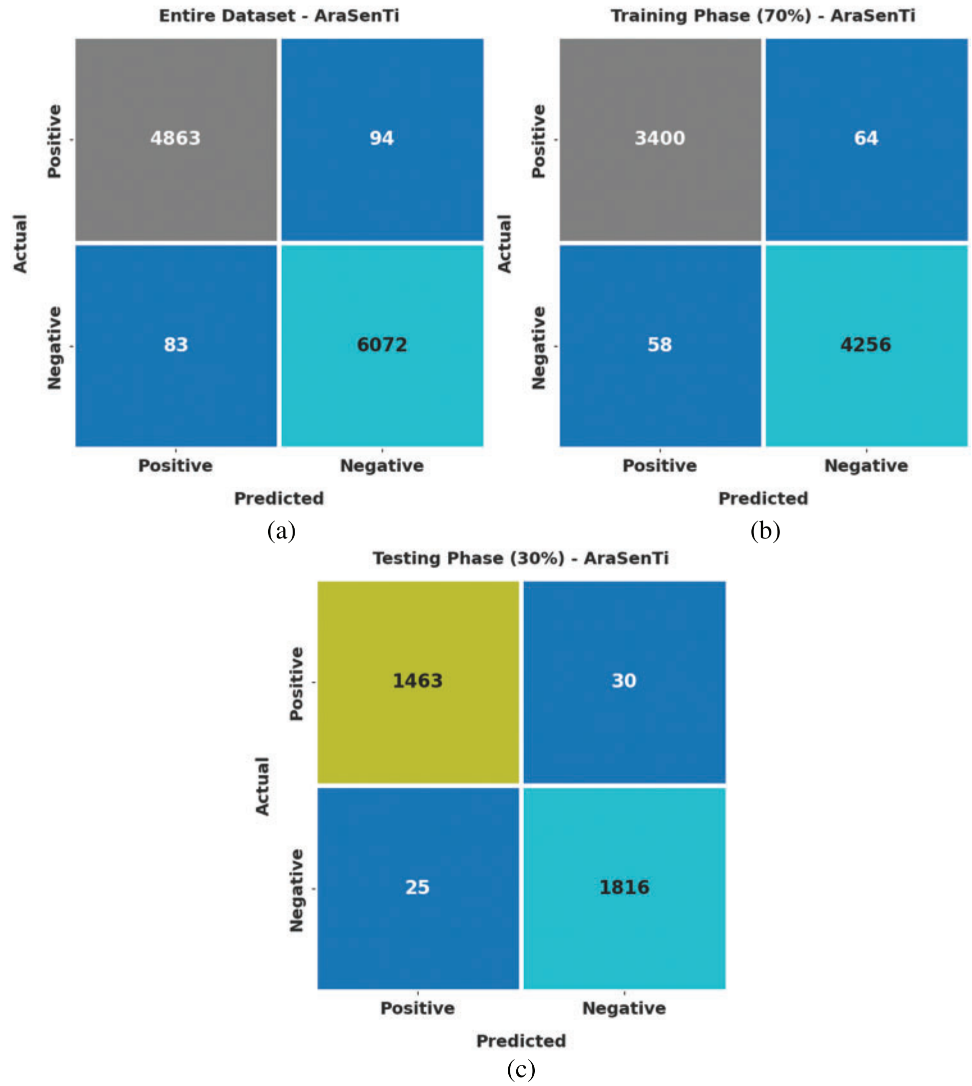
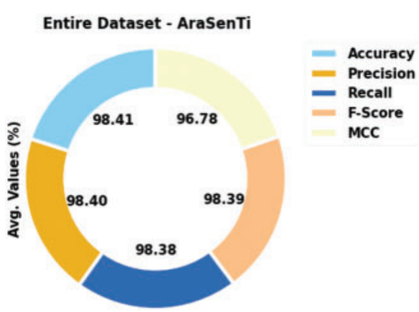


Figure 8: Confusion matrices of the QPSODL-SAAT approach using AraSenTi dataset (a) Entire dataset, (b) 70% of the TR data and (c) 30% of the TS data

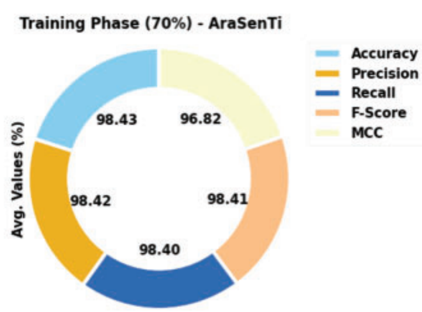
Table 4 and Fig. 9 depict the overall SA results accomplished by the proposed QPSODL-SAAT method on AraSenTi dataset. With whole dataset, the presented QPSODL-SAAT system obtained an average $accu_y$ of 98.41%, $prec_n$ of 98.40%, $reca_l$ of 98.38%, F_{score} of 98.39% and an MCC of 96.78%. On 70% of the TR data, the presented QPSODL-SAAT approach reached an average $accu_y$ of 98.43%, $prec_n$ of 98.42%, $reca_l$ of 98.40%, F_{score} of 98.41%, and an MCC of 96.82%. In addition to these, with 30% of the TS data, the proposed QPSODL-SAAT method accomplished an average $accu_y$ of 98.35%, $prec_n$ of 98.35%, $reca_l$ of 98.32%, F_{score} of 98.33% and an MCC of 96.66%.

Table 4: Analytical results of the QPSODL-SAAT approach on AraSenTi dataset

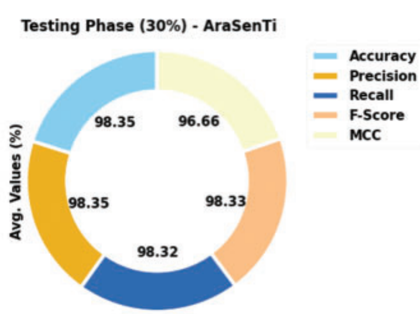
AraSenTi dataset					
Labels	Accuracy	Precision	Recall	F-Score	MCC
Entire dataset					
Positive	98.41	98.32	98.10	98.21	96.78
Negative	98.41	98.48	98.65	98.56	96.78
Average	98.41	98.40	98.38	98.39	96.78
Training phase (70%)					
Positive	98.43	98.32	98.15	98.24	96.82
Negative	98.43	98.52	98.66	98.59	96.82
Average	98.43	98.42	98.40	98.41	96.82
Testing phase (30%)					
Positive	98.35	98.32	97.99	98.15	96.66
Negative	98.35	98.37	98.64	98.51	96.66
Average	98.35	98.35	98.32	98.33	96.66



(a)



(b)



(c)

Figure 9: Average analysis results of the QPSODL-SAAT approach on AraSenTi dataset (a) Entire dataset, (b) 70% of the TR data and (c) 30% of the TS data

Both TA and VA values, acquired by the proposed QPSODL-SAAT methodology on AraSenTi data, are demonstrated in Fig. 10. The experimental results portray that the proposed QPSODL-SAAT methodology accomplished the high TA and VA values whereas the VA values were higher than the TA values.

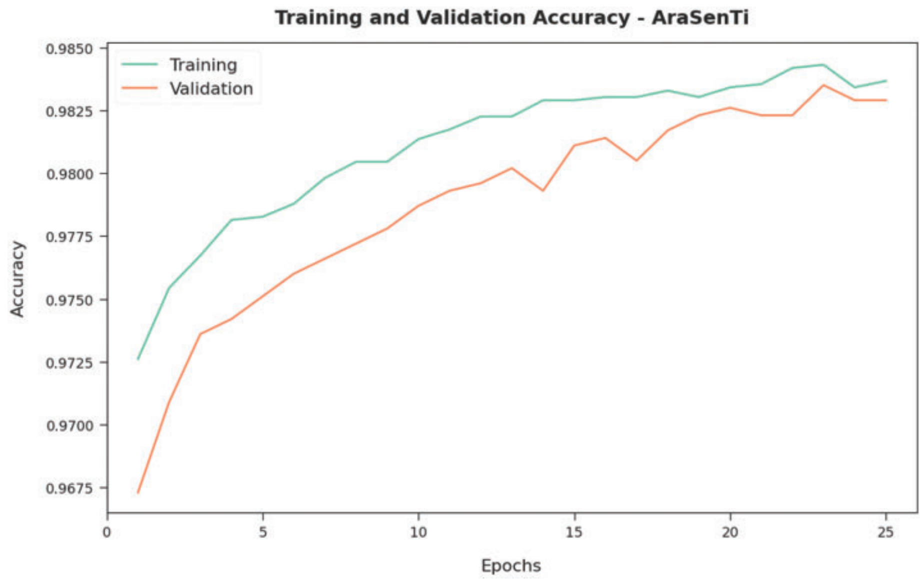


Figure 10: TA and VA analyses results of the QPSODL-SAAT approach on AraSenTi dataset

Both TL and VL values, accomplished by QPSODL-SAAT method on AraSenTi data, are illustrated in Fig. 11. The experimental results imply that the proposed QPSODL-SAAT approach obtained the minimal TL and VL values whereas the VL values were lower than the TL values.

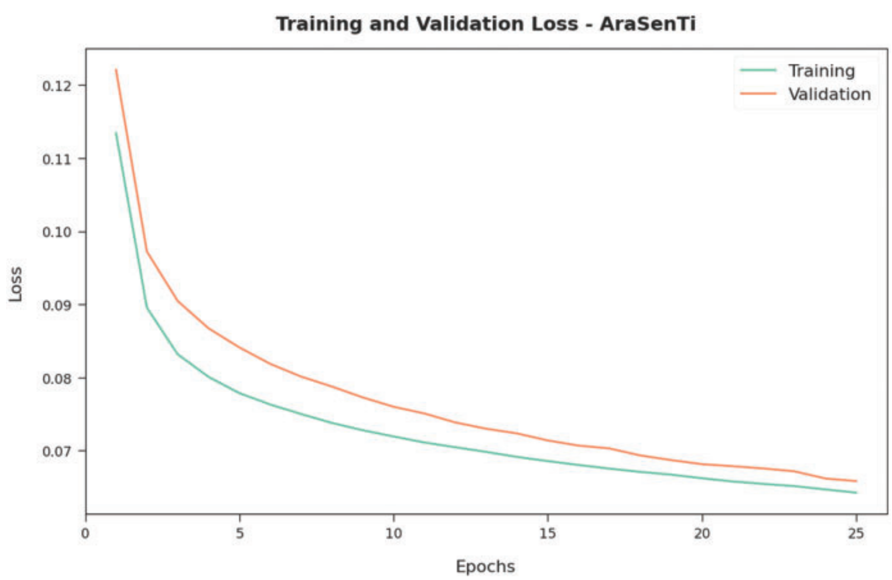


Figure 11: TL and VL analyses results of the QPSODL-SAAT approach on AraSenTi dataset

Table 5 and Fig. 12 portray the comparative analysis outcomes accomplished by the proposed QPSODL-SAAT approach and other existing methods such as the Surface Features (SF), Generic Embeddings (GE), Arabic Sentiment Embeddings constructed using the Hybrid (ASEH) and the Bidirectional Encoder Representations from Transformers (BERT) Model on AraSenTi dataset [24]. The attained outcomes establish the supremacy of the proposed QPSODL-SAAT approach over other methodologies. With regards to $accu_y$, the proposed QPSODL-SAAT model achieved a maximum $accu_y$ of 98.35% whereas the SF, GE, ASEH and the BERT technique accomplished the least $accu_y$ values, such as 90.60%, 91.81%, 95.11% and 93.76% correspondingly.

Table 5: Comparative analysis results of the QPSODL-SAAT approach and other existing methodologies on AraSenTi dataset

AraSenTi Dataset				
Methods	Accuracy	Precision	Recall	F-score
QPSODL-SAAT	98.35	98.35	98.32	98.33
SF model	90.60	89.94	91.74	88.00
GE model	91.81	93.17	93.92	91.94
ASEH model	95.11	95.07	94.05	93.07
BERT model	93.76	95.16	94.05	96.88

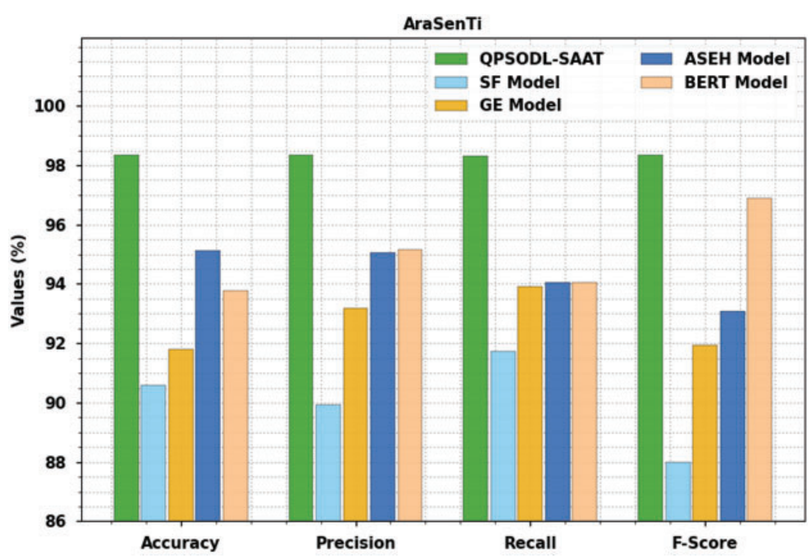


Figure 12: Comparative analysis results of the QPSODL-SAAT approach and other existing methodologies on AraSenTi dataset

Meanwhile, with regards to $reca_l$, the proposed QPSODL-SAAT method presented a maximum $reca_l$ of 98.32% whereas the SF, GE, ASEH and the BERT approaches accomplished the least $reca_l$ values such as 91.74%, 93.92%, 94.05% and 94.05% correspondingly. Eventually, with regard to F_{score}, the proposed QPSODL-SAAT method achieved a maximum F_{score} of 98.33% while the SF, GE, ASEH and the BERT approaches achieved the least F_{score} values such as 88%, 91.94%, 93.07% and 96.88% correspondingly.

4 Conclusion

In this study, a new QPSODL-SAAT model has been developed for recognition and the classification of the sentiments found in Arabic tweets. Initially, the data pre-processing is performed to convert the raw tweets into a useful format. Then, the word2vec model is applied to generate the feature vectors. To identify and classify the sentiments, the BiGRU classifier is utilized. Finally, the QPSO algorithm is exploited for optimal fine-tuning of the hyperparameters involved in the BiGRU model. The proposed QPSODL-SAAT model was experimentally validated using two standard datasets. The extensive comparative analyses outcomes established the supremacy of the proposed QPSODL-SAAT model over recent approaches. Therefore, the QPSODL-SAAT model can be utilized for an effectual recognition of the sentiments found in Arabic tweets. In the future, the presented model can also be extended to identify the sarcastic tweets made in Arabic.

Funding Statement: The authors extend their appreciation to the Deanship of Scientific Research at King Khalid University for funding this work through Small Groups Project under Grant Number (120/43). 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: (22UQU4310373DSR36).

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

References

[1] M. K. Alsmadi, I. A. Marashdeh, M. Alzaqebah, G. Jaradat, F. A. Alghamdi *et al.,* “Digitalization of learning in Saudi Arabia during the COVID-19 outbreak: A survey,” *Informatics in Medicine Unlocked*, vol. 25, pp. 100632, 2021.
[2] B. Ihnaini and M. Mahmuddin, “Lexicon-based sentiment analysis of Arabic tweets: A survey,” *Journal of Engineering and Applied Sciences*, vol. 13, no. 17, pp. 7313–7322, 2018.
[3] N. Boudad, R. Faizi, R. O. H. Thami and R. Chiheb, “Sentiment analysis in Arabic: A review of the literature,” *Ain Shams Engineering Journal*, vol. 9, no. 4, pp. 2479–2490, 2018.
[4] F. N. Al-Wesabi, “Entropy-based watermarking approach for sensitive tamper detection of Arabic text,” *Computers, Materials & Continua*, vol. 67, no. 3, pp. 3635–3648, 2021.
[5] E. A. H. Khalil, E. M. F. E. Houby and H. K. Mohamed, “Deep learning for emotion analysis in Arabic tweets,” *Journal of Big Data*, vol. 8, no. 1, pp. 136, 2021.
[6] 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.
[7] D. Gamal, M. Alfonse, E. S. M. El-Horbaty and A. B. M. Salem, “Twitter benchmark dataset for Arabic sentiment analysis,” *International Journal of Modern Education and Computer Science*, vol. 11, no. 1, pp. 33, 2019.
[8] 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.
[9] N. Al-Twairesh, “The evolution of language models applied to emotion analysis of Arabic tweets,” *Information*, vol. 12, no. 2, pp. 84, 2021.

[10] 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.
[11] M. Baali and N. Ghneim, "Emotion analysis of Arabic tweets using deep learning approach," *Journal of Big Data*, vol. 6, no. 1, pp. 89, 2019.
[12] M. Alqmase, H. Al-Muhtaseb and H. Rabaan, "Sports-fanaticism formalism for sentiment analysis in Arabic text," *Social Network Analysis and Mining*, vol. 11, no. 1, pp. 52, 2021.
[13] I. A. Farha and W. Magdy, "A comparative study of effective approaches for Arabic sentiment analysis," *Information Processing & Management*, vol. 58, no. 2, pp. 102438, 2021.
[14] M. Aljabri, S. M. B. Chrouf, N. A. Alzahrani, L. Alghamdi, R. Alfehaid *et al.,* "Sentiment analysis of Arabic tweets regarding distance learning in Saudi Arabia during the covid-19 pandemic," *Sensors*, vol. 21, no. 16, pp. 5431, 2021.
[15] Q. Baker, F. Shatnawi, S. Rawashdeh, M. A. Smadi and Y. Jararweh, "Detecting epidemic diseases using sentiment analysis of Arabic tweets," *Journal of Universal Computer Science*, vol. 26, no. 1, pp. 50–70, 2020.
[16] K. Elshakankery and M. F. Ahmed, "HILATSA: A hybrid incremental learning approach for Arabic tweets sentiment analysis," *Egyptian Informatics Journal*, vol. 20, no. 3, pp. 163–171, 2019.
[17] S. Behdenna, F. Barigou and G. Belalem, "Sentiment analysis of Arabic tweets: Opinion target extraction," *Journal of Digital Information Management*, vol. 16, no. 6, pp. 324, 2018.
[18] A. Mohammed and R. Kora, "Deep learning approaches for Arabic sentiment analysis," *Social Network Analysis and Mining*, vol. 9, no. 1, pp. 52, 2019.
[19] T. A. Moslmi, M. Albared, A. A. Shabi, N. Omar and S. Abdullah, "Arabic senti-lexicon: Constructing publicly available language resources for Arabic sentiment analysis," *Journal of Information Science*, vol. 44, no. 3, pp. 345–362, 2018.
[20] H. Gul, N. Javaid, I. Ullah, A. M. Qamar, M. K. Afzal *et al.,* "Detection of non-technical losses using sostlink and bidirectional gated recurrent unit to secure smart meters," *Applied Sciences*, vol. 10, no. 9, pp. 3151, 2020.
[21] S. Djemame, M. Batouche, H. Oulhadj and P. Siarry, "Solving reverse emergence with quantum PSO application to image processing," *Soft Computing*, vol. 23, no. 16, pp. 6921–6935, 2019.
[22] S. Rosenthal, N. Farra and P. Nakov, "SemEval-2017 task 4: Sentiment analysis in twitter," in *Proc. of the 10th Int. Workshop on Semantic Evaluation (SemEval 2016)*, San Diego, US, pp. 1–18, 2016.
[23] N. A. Twairesh, H. Al-Khalifa, A. Al-Salman and Y. Al-Ohali, "Arasenti-tweet: A corpus for Arabic sentiment analysis of Saudi tweets," *Procedia Computer Science*, vol. 117, pp. 63–72, 2017.
[24] N. A. Twairesh and H. A. Negheimish, "Surface and deep features ensemble for sentiment analysis of Arabic tweets," *IEEE Access*, vol. 7, pp. 84122–84131, 2019.