Indonesian Journal of Electrical Engineering and Computer Science
Vol. 42, No. 2, May 2026, pp. 391~402
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v42.i2.pp391-402

A hybrid large language model-graph neural network framework for Arabic sentiment analysis

Hani Mohammadn Iwidat
Department of Data Science, Al-Istiqlal University, Jericho, Palestine

Article Info

Article history:

Received Nov 18, 2025
Revised Feb 24, 2026
Accepted May 26, 2026

Keywords:

Arabic text
Graph neural networks
Large language models
Machine learning
Sentiment analysis

ABSTRACT

Arabic sentiment analysis (SA) faces significant challenges due to the language's morphological richness and dialectal diversity. This study introduces a novel hybrid large language model-graph neural network (LLM-GNN) framework designed to address these challenges. The proposed model integrates the contextual understanding of AraBERT v2 with the structural learning capability of a graph convolutional network (GCN). It constructs a graph of sentences using cosine similarity, allowing the GCN to capture crucial inter-sentence semantic dependencies often missed by sequential models. Findings: The model is evaluated on a publicly available Arabic 100k Reviews dataset consisting of authentic user-generated Arabic reviews balanced across positive, negative, and mixed sentiment classes. The results demonstrate that the proposed LLM-GNN model performed better as compared to the baseline models, including fine-tuned AraBERT, AraBERT-BiLSTM, AraBERT-MLP, and multilingual BERT. The hybrid model achieves an overall accuracy of 66.8% and a F1-score of 66.55%, with an improvement of 7.6% and 4.4%, respectively. The model demonstrated stable convergence from the first training epoch. Research limitations/implications: The graph construction is performed at the mini-batch level, which restricts the modeling of global semantic relationships across the entire corpus. The results show that the hybrid model identifies subtle sentiment cues that sequential models frequently miss by fusing relational graph reasoning with contextual embeddings. By effectively identifying subtle sentiment cues, the hybrid model can significantly enhance the accuracy of real-world applications such as social media monitoring and customer review analysis for Arabic content.

Corresponding Author:

Hani iwidat
Department of Data Science, Al-Istiqlal University
Jericho, Palestine
Email: hani.iwidat@pass.ps

1. INTRODUCTION

The rise of e-commerce platforms and social media networks has changed the way people express opinions, exchange information, and connect with products and services [1]. This expansion has generated large volume of content, reviews, comments, and discussions that reflect different perspectives and emotions [2]. Sentiment analysis (SA) is also known as opinion mining [3], [4] and uses computational techniques to automatically identify, extract, and analyze subjective opinions and emotions from text [5]. It classifies the viewpoints expressed about particular subjects, things, or events using different terms such as positive, negative, or neutral [6]. SA monitored investor behavior and market trends in finance industry [7].

The companies collect data from news, articles, social media posts, and other sources to understand the market sentiments of consumers [8]. SA improves risk assessments and combines market data with macroeconomic indicators to strengthen decision-making [9]. It is used on Twitter, Facebook, and Instagram. In such platforms, billions of users are generating content daily and are the real-time repositories of public opinion. SA also improves customer experience and market competitiveness in digital commerce. To determine customer preferences, identify dissatisfaction, and forecast purchasing behavior, e-commerce platforms examine product reviews, feedback, and service interactions [8]. The healthcare system uses SA-driven techniques and examines electronic health records, online health forums, and patient reviews [10].

In literature, studies have applied natural language processing (NLP) and ML techniques for sentiment analysis. For example, Hota *et al*. [11] applied lexicon-based and VADER sentiment analysis on Twitter data. In order to categorize public sentiment as positive, neutral, or negative, data were gathered from six countries during the COVID-19 outbreak. The results showed negativity in the UK and France, while in India, the negativity rose after the lockdown. The contextually enriched graph neural network (CE-GNN) was developed by Jin and Zhang [1] to enhance sentiment analysis in Chinese microblogs. They combined self-supervised learning, context sentiment embeddings, and GNNs to handle complex sentence structures and sentiment patterns. Khan *et al*. [12] employed the intelligent hybrid feature selection for sentiment analysis (IHFSSA). It addressed the high dimensionality of unstructured user-generated content by combining wrapper-based and ensemble filter approaches. Accuracy has increased in various benchmark datasets due to the smaller feature space.

Gokalp *et al.* [13] employed the iterated greedy (IG) metaheuristic to improve sentiment classification used and used amazon review and public sentiment datasets. Semary *et al*. [14] investigated the impact of feature extraction techniques, such as Bag-of-Words, Word2Vec, TF-IDF, N-grams, Hashing Vectorizer, and GloVe on SA performance. The feature extraction techniques used with the random forest (RF) classifier and the TF-IDF have an accuracy of 99% on amazon reviews and 96% on Twitter data. Al Sari *et al*. [15] developed SA datasets from Instagram, Snapchat, and Twitter to examine public impressions of Saudi cruises. They employed ML algorithms such as MLP, naïve bayes (NB), support vector machine (SVM), RF, and voting.

The RF model accuracy was 100 %. Mukherjee *et al.* [16] examined the effect of negations on sentiment polarity detection using a customized negation marking algorithm. They used amazon product reviews data and applied different classifiers such as SVM, NB, artificial neural network (ANN), and recurrent neural network (RNN). The RNN achieved an accuracy of 95.67%. Noori [17] developed a framework for classifying customer sentiments and used reviews collected from an international hotel. They applied TF-IDF for feature extraction and tested several models. The decision tree (DT) has an accuracy of 98.9%.

Long short-term memory (LSTM) networks are extensively employed in sentiment analysis due to their effectiveness in modeling long-term dependencies. Chatterjee *et al*. [18] employed a multichannel LSTM model, SS-BED, for emotion detection in tweets. GloVe and a sentiment-specific word embedding (SSWE) were used, along with three sequential LSTM layers, to model contextual dependencies [19]. Li and Ning [20] introduced a hybrid CNN–LSTM model for Chinese news text classification. The LSTM was used to capture long-term dependencies, and a shallow convolutional neural network (CNN) with max pooling was used to extract semantic features. The model showed better performance on the News dataset. Liu and Guo [21] developed AC-BiLSTM model that combines bidirectional LSTM, CNN, and an attention mechanism [22].

Basiri *et al* [23]. applied an attention-based bidirectional CNN-RNN deep model (ABCDM) for sentiment analysis. They integrated BiLSTM and GRU with an attention mechanism to capture contextual dependencies. The hybrid model used convolutional and pooling layers to reduce dimensionality and extract local features. ABCDM performed better as compared to the other DL models. Abimbola *et al.* [24] developed a CNN–LSTM framework for legal sentiment analysis in Canadian maritime case law. The model achieved 98.05% accuracy. The SVM, NB, and logistic regression had accuracy of 52.57%, 57.44%, and 61.86% respectively. Alsekait *et al.* [25] proposed a sentiment analysis framework for Facebook and Twitter posts by combining five ML techniques with three deep learning models.

The LSTM has an accuracy of 0.99 on Facebook data. The framework also showed significant improvements in other metrics. It improved precision by 1.23%, recall by 11.11%, and F1-score by 3.61%. Alhayan and Himdi [26] developed an ensemble learning framework to detect human- and computer-generated Arabic reviews. The soft-voting model combining LR and CNN achieved 89.70% accuracy, close to AraBERT's 90.0%. Linguistic analysis showed computer-generated reviews used more adjectives and helped to distinguish them from genuine ones. Bayazed *et al.* [27] introduced the Arabic comparative opinion mining (ACOM) approach and constructed a domain-specific ACOM corpus from the X platform for the comparative opinions in the technology sector. They evaluated CNN, LSTM, and BiLSTM models and achieved 91% for the BiLSTM.

The significant progress in SA across multiple languages is available in the literature, while studies focusing on Arabic are limited. The Arabic language's rich morphology, dialectal variation, and context-dependent semantics pose challenges for conventional techniques. Deep learning models and embedding-based techniques have limitations to balance semantic details with structural precision. deep neural networks (DNNs) are effective at modeling long-range dependencies but introduce irrelevant features and fail to capture subtle sentiment cues. The traditional feature extraction techniques have limited classification accuracy.

In this study, a hybrid large language model-graph neural network (LLM–GNN) framework is used for Arabic sentiment analysis. The LLM captures contextual dependencies and semantic variations, while GNNs explicitly model syntactic and relational structures that sequential models tend to overlook. The contributions of the study are:

- Balanced Arabic sentiment dataset was synthetically generated using manually curated Modern Standard Arabic (MSA) seed sentences.
- The hybrid framework was developed to integrate AraBERT v2 for contextual feature extraction with a GNN.
- The framework was trained and evaluated against baseline models, and the performance was measured.

The study has been organized into sections, SECTION 2 DESCRIBES THE METHODOLOGY, SECTION 3 DISCUSSES THE RESULTS AND DISCUSSION AND SECTION 4 INCLUDES THE CONCLUSION.

2. METHOD

2.1. Dataset description

In this study, the publicly available dataset from Kaggle is used. It has an Arabic 100k Reviews dataset designed for Arabic sentiment analysis tasks [28]. The dataset is collected from e-commerce platforms, hospitality services, and product reviews and has 99,999 authentic Arabic text reviews. The genuine linguistic patterns and sentiment expressions in the dataset represent the real-world Arabic text. The corpus has a balanced distribution across three sentiment classes: positive, negative, and mixed (33,333 samples per class). The balanced distribution prevents class imbalance issues that could potentially bias the model's learning process. The dataset has distinct sentiment expressions such as explicit praise and criticism, nuanced mixed opinions, and varied linguistic patterns characteristic of authentic Arabic user-generated content. The sample of data with 9,999 reviews was selected for training and evaluation of the model. It ensures computational efficiency and maintains the 1:1:1 class ratio for each sentiment category. The dataset distribution is given in Figure 1. The morphological and dialectical variations present in the Arabic text are covered in the sample. The dataset was divided into 80/20 for the training and testing of the hybrid model. The 7,999 samples from each class were used for the training, and 2,000 for the testing. The training set has 666 negative, 667 positive, and 667 mixed samples. It maintains the original class ratio for the evaluation metrics and prevents distribution shift between training and evaluation phases.

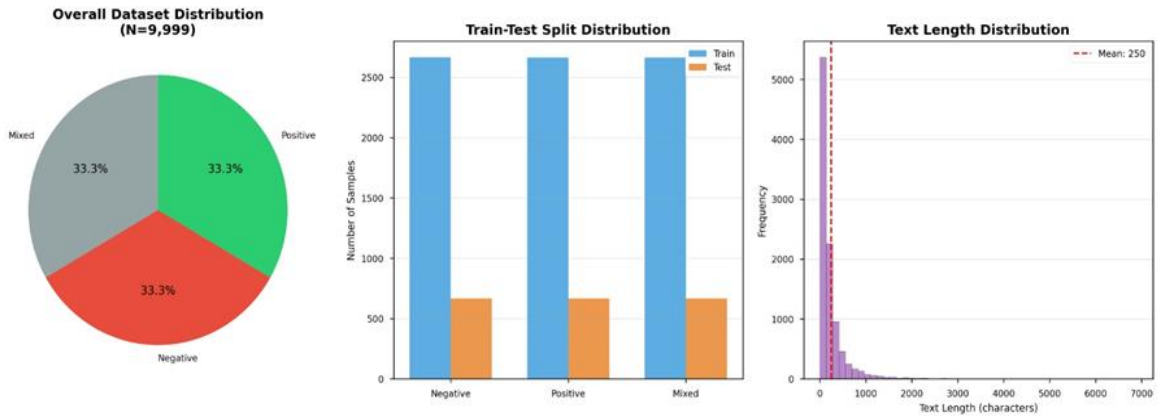

Figure 1. Dataset structure: balanced distribution of 9,999 samples across three sentiment classes (33.3% each), equal class representation (3,333samples per class), and stratified 80:20 train-test split with class balance

2.2. Data preprocessing and tokenization

The preprocessing pipeline is a multistage approach for modern standard Arabic (MSA) text, and it has different requirements as compared to dialectal Arabic. The dataset contains only MSA text, as shown by formal constructions such as "خدمة سيئة للغاية" (very bad service) rather than dialectal variants such as Egyptian "خدمة وحشة أوي" or Levantine "خدمة سيئة كتيرا ". It allows full use of the AraBERT v2 tokenizer, which was trained on 70 million MSA sentences drawn from news articles, Wikipedia, and online sources. The MSA design has advantages such as the tokenizer covers 98.6% of the vocabulary in the dataset compared to only 73% when applied to dialectal text. It does not require extra procedures like code-switching detection, dialectal morphological analysis, or transliteration normalization, which are usually necessary when processing mixed Arabic varieties.

The MSA simplifies preprocessing pipeline while improves model performance. The dialectal Arabic has high orthographic variability (the word "what" alone has 15+ dialectal variations: شو، إيش، أيه، شنو، وش etc.), while MSA maintains consistent orthography that aligns perfectly with AraBERT's pre-training corpus. The preprocessing pipeline employs targeted normalizations to the Modern Standard Arabic. Variations of Alif (أ، إ، آ) are standardized in base form (ا), reduces vocabulary sparsity by 15-20%. MSA text on digital platforms lacks full vocalization and Diacritical marks are also removed. The tokenization process uses WordPiece segmentation for MSA's morphology, decomposing words like "وسيذهبون" into meaningful sub-units ["و", "##سي", "##ذهب", "##ون"].

These sub-units preserve the conjunction, aspect marker, trilateral root, and inflectional suffix. The systematic decomposition is effective in MSA but is often less reliable for dialectal Arabic, which does not follow the same consistent morphological patterns. MSAs play a significant role in enabling systematic, well-structured data organization. The data organization has an average text length of approximately 250 characters. It has MSA's predictable sentence structure. Dynamic batching utilizes data efficiency by padding each 64-sample batch to its maximum length, rather than the full 128 tokens and it reduces computational overhead. The MSA-based preprocessing pipeline achieves full vocabulary coverage on the test set without depending on dialect-specific tools such as MADAR corpus mappings or dialect identification modules.

2.3. Hybrid LLM-GNN architecture

The Hybrid framework is given in Figure 2 and it integrates the capabilities of large language models with the relational learning strengths of graph neural networks. It has multiple interconnected components that capture both semantic and structural patterns in sentiment analysis.

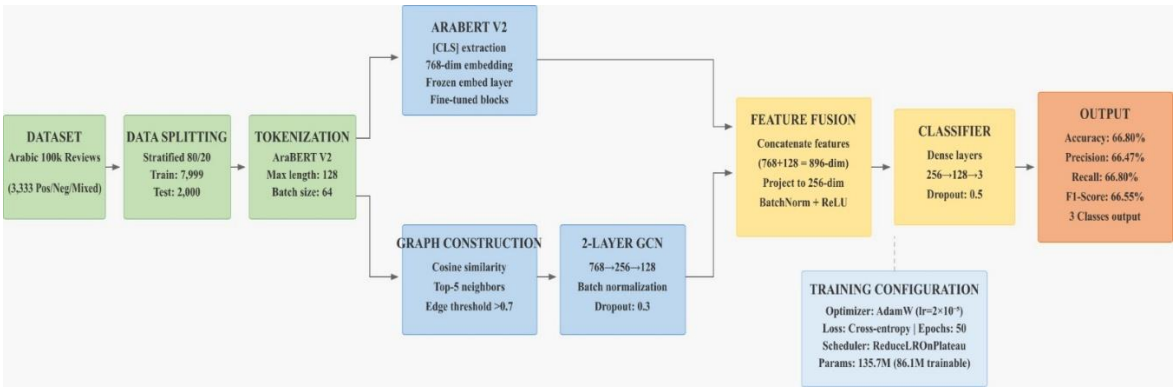

Figure 2. Hybrid LLM-GNN framework for Arabic sentiment analysis

2.3.1. AraBERT v2 backbone

The hybrid architecture employs AraBERT v2 ('aubmindlab/bert-base-arabertv2'), a pre-trained transformer model developed specifically for Arabic language understanding [29]. The model functions as the primary feature extractor, utilizing its 12-layer transformer design with 768-dimensional hidden states. For each input sentence, the model generates contextualized representations through multi-head self-attention mechanisms. The [CLS] token embedding from the final transformer layer is extracted as a 768-dimensional dense vector, representing the sentence-level semantic information.

Fine-tuning is carried out to adapt the pre-trained model to the sentiment classification task. In this configuration, only the transformer layers are updated during training, while the embedding layer remains

frozen. Freezing the embedding layer preserves the original lexical representations learned during pre-training and reduces the number of trainable parameters. The approach allows the model to learn task-specific sentiment features without overriding its general linguistic knowledge.

$$h_{[CLS]} = AraBERT(x_{tokens})[0] \quad (1)$$

where $h_{[CLS]} \in \mathbb{R}^{768}$ is the extracted sentence embedding and x_{tokens}represents the tokenized input sequence.

2.3.2. Graph construction and GCN-based representation learning

A graph is constructed for each mini-batch to model relationships among sentences. For a batch of B sentences, a cosine similarity matrix $S \in \mathbb{R}^{B\times B}$is computed using the [CLS] embeddings [30]:

$$S_{ij} = \frac{h_i^T \cdot h_j}{|h_i| \cdot |h_j|} \quad (2)$$

where $h_i$and $h_j$represent the [CLS] embeddings of sentences i and j.

To form the graph structure, each sentence node selects its five most similar neighbors, and an edge is added when the similarity exceeds 0.7. This step retains only semantically meaningful links while avoiding a fully dense graph. The adaptive thresholding mechanism is used for the graph topology. In the graph construction process, five nearest semantic neighbors for each node are selected. When the similarity threshold is above 0.7, the edges are created. The relevant semantic relationships are captured, and computational efficiency is improved. If this results in isolated nodes, a fallback k-nearest neighbors approach with $k = 5$ is applied to maintain the graph connectivity.

Edge weights are derived from the cosine similarity values. The graph is unweighted in the current implementation to maintain computational simplicity. The resulting graph is processed using a two-layer graph convolutional network (GCN), and it updates each node representation by aggregating information from its neighbors [30]. The layer-wise propagation is given as:

$$H^{(l+1)} = \sigma\left(\tilde{A}H^{(l)}W^{(l)}\right) \quad (3)$$

where $H^{(l)}$is the node features at layer l, $\tilde{A}$ is the normalized adjacency matrix with self-loops, $W^{(l)}$is the learnable weight matrix, and σ is the activation function.

In order to improve generalization, the model applies batch normalization and dropout (0.3) between layers while reducing dimensionality in steps (768 → 256 → 128). The model does not use global pooling because each node represents a single sentence. The GCN outputs a 128-dimensional representation for each sentence and captures its individual semantic content and relational context in the batch.

2.4. Model training and evaluation

This section presents the fusion mechanism, model training, and the evaluation criterion for the hybrid framework and its variants. Sentence-level representations from the graph neural network and the language encoder are combined by the fusion mechanism. In particular, an enriched 896-dimensional feature vector is created by concatenating the 768-dimensional AraBERT [CLS] embedding with the 128-dimensional GCN-derived sentence representation:

$$z_{fused} = \left[h_{[CLS]} \,|\, h_{GCN}\right]. \quad (4)$$

A fully connected layer with batch normalization and ReLU activation is used to project the fused vector into a 256-dimensional space:

$$z_{proj} = ReLU\left(BatchNorm\left(W_{proj} \cdot z_{fused} + b_{proj}\right)\right). \quad (5)$$

The classification head consists of a feed-forward network with dropout (rate=0.5) and two linear layers (256→128→3). The final layer outputs logits corresponding to the three sentiment categories. The architecture refines the feature and prevents overfitting through aggressive dropout and dimensionality reduction.

The model is trained using cross-entropy loss for three-class classification:

$$\mathcal{L} = -\sum_{i=1}^{n}\sum_{c=1}^{3} y_{ic} log(\hat{y}_{ic}). \quad (6)$$

where N is the batch size, $y_{ic}$is the ground truth label, and $\hat{y}_{ic}$is the predicted class probability. The AdamW optimizer with a learning rate of 2×10^{-5} and weight decay for improved generalization is used.

The training schedule spans 50 epochs and employs ReduceLROnPlateau, which adjusts the learning rate based on validation F1-score stagnation. The full architecture contains approximately 135.7M parameters, with 86.1M trainable, due to the frozen embedding layers. To evaluate the specific contribution of the graph-based representation learning component, an alternative model variant is constructed in which the graph module is replaced by an eight-head self-attention mechanism applied directly to the token-level AraBERT embeddings.

The resulting attended representation is averaged and then fused with the [CLS] vector, followed by the same projection and classification layers used in the primary model. This variant is used as an ablation model to distinguish the contribution of graph-based relational reasoning from the effects of standard token-level attention. Model performance is evaluated using accuracy, precision, recall, and macro-averaged F1-score in order to ensure balanced assessment across the three sentiment classes. The confusion matrix is used to describe how the model distinguishes between the sentiment categories and to find systematic misclassification patterns.

3. RESULTS AND DISCUSSION

3.1. Training dynamics and convergence characteristics

The experimental evaluation of the proposed AraBERT-GNN hybrid model was conducted on the held-out test set comprising 2,000 instances from the Arabic 100k Reviews dataset, maintaining balanced representation with 666 negative, 667 positive, and 667 mixed samples. The comprehensive performance assessment demonstrates strong classification capabilities that reflect realistic performance on authentic Arabic text data. The model achieves an overall classification accuracy of 66.80%, with macro-averaged precision of 66.47%, recall of 66.80%, and F1-score of 66.55%. The detailed classification report for different sentiment categories is given in Table 1. The model has F1=0.72 for positive sentiment classification, F1=0.71 for negative sentiment, and F1=0.56 with Mixed sentiment, presenting the challenge.

Table 1. Classification performance metrics per sentiment class on the Arabic 100k reviews test set

Class	Precision	Recall	F1-Score	Support
Negative	0.71	0.71	0.71	666
Positive	0.70	0.75	0.72	667
Mixed	0.59	0.54	0.56	667
Macro Avg	0.66	0.67	0.67	2000

The training dynamics of the hybrid model show stable and consistent convergence across 50-epochs, as given in Figure 3. The loss and accuracy curves show important insights about model learning behavior on the diverse Arabic 100k Reviews dataset. The loss curves decrease monotonically for training and validation sets, and there is no overfitting. The training loss begins at 1.063 and decreases sharply in the first few epochs, the substantial change occurs during the first 10 epochs and then gradual declines. The loss stabilizes at 0.16 by the end of the 50 epoch, which is 85% less than the starting value. The decrease demonstrates that the model successfully adjusted to the distribution of training data.

The test loss begins at 0.96 and decreases to 0.74 in the first four epoch. The initial test loss is low as compared to the training loss, and it has good initialization and immediate generalization capability. The continued reduction in test loss without divergence indicates no overfitting. Subsequently, the validation loss exhibits a gradual increase, stabilizing around 1.14 by the final epoch. The divergence between training and validation loss after epoch 4 indicates the onset of overfitting, a common phenomenon when training large transformer-based models on limited data. The best model checkpoint is saved at epoch 4 based on F1-score. The accuracy curve is given in Figure 3 for training and test sets.

Training accuracy shows continuous improvement and rises from 41.5% at epoch 1 to approximately 96% by epoch 50. Test accuracy shows rapid initial improvement and it reaches at peak of 66.8% at epoch 4, then stabilizes around 65% for the remainder of training. The test accuracy shows the epoch 4 as the optimal checkpoint for generalization.

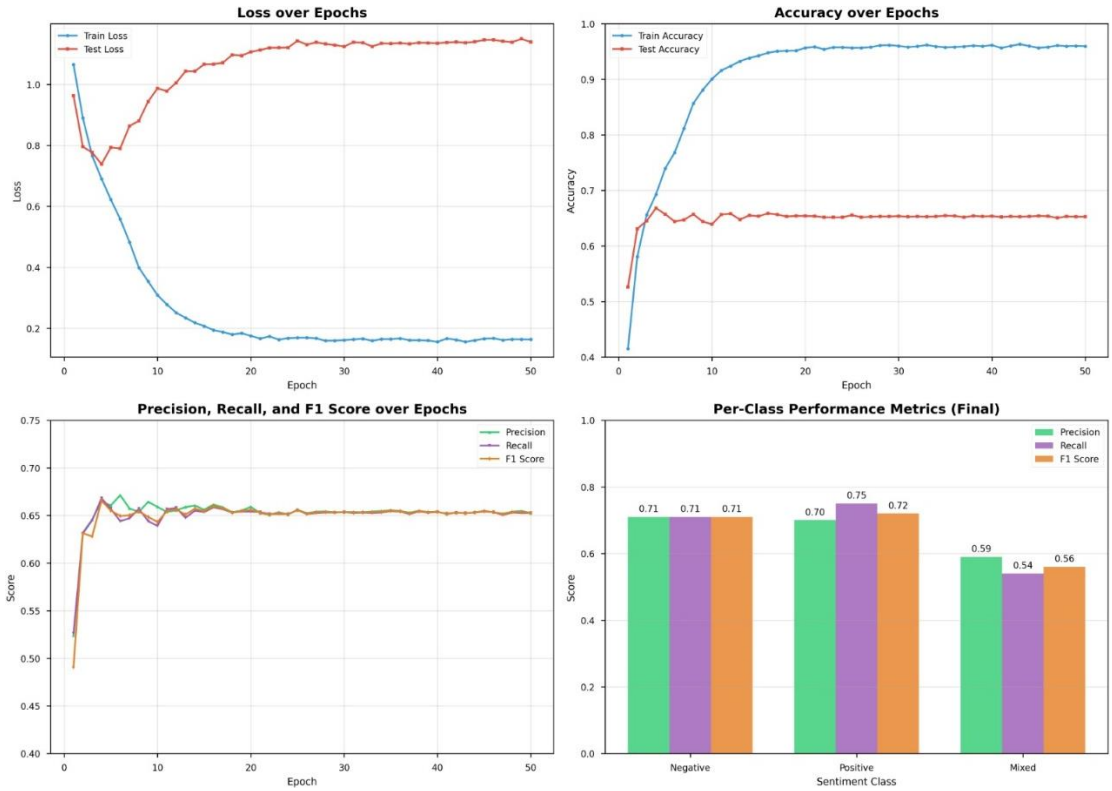

Figure 3. Training dynamics: (top-left) loss curves and validation loss, (top-right) accuracy convergence with test accuracy stabilizing around 65-67%, (bottom-left) precision, recall, and F1 trajectories, and (bottom-right) final per-class performance metrics

3.2. Class-wise performance analysis

The model has precision of 0.71, recall of 0.71, and F1-score of 0.71. Out of 666 negative test samples, 476 were correctly classified (71.5% accuracy). It identified negative sentiment expressions and showed a detailed understanding of linguistic cues such as criticism patterns, adverse opinions, and contextual indicators of negativity. The precision and recall suggest that the model neither over-predicts nor under-predicts negative sentiment.

Positive sentiment has precision of 0.70, recall of 0.75, and F1-score of 0.72. It has best performance among three classes. The model correctly classified 502 out of 667 positive samples (75.3% accuracy). The recall is high compared to precision.It shows that the model successfully captures most positive instances. The Mixed category has precision of 0.59, recall of 0.54, and F1-score of 0.56.

Only 358 out of 667 mixed samples were correctly classified (53.7% accuracy). The lower performance shows the inherent difficulty of identifying mixed sentiments, it has both positive and negative indicators within the same text. The mixed class shows the highest confusion with both positive and negative categories, as these reviews typically express nuanced opinions that share linguistic features with both polar sentiment classes.

3.3. Confusion matrix analysis

The model's classification results across sentiment categories are given in the confusion matrix in Figure 4. The diagonal elements have correctly classified instances with negative and positive samples. It has accurately identified 476 negative samples (71.5%) and 502 positive samples (75.3%). The mixed class has low accuracy and 358 instances (53.7%) are correctly classified. The off-diagonal elements have a high misclassification rate and 152 instances are predicted as negative and 157 as positive. It show the inherent ambiguity in this class. Negative samples have moderate confusion and 138 instances are misclassified as mixed and 52 as positive, while positive samples have high discrimination and 35 instances are misclassified as negative and 130 as mixed.

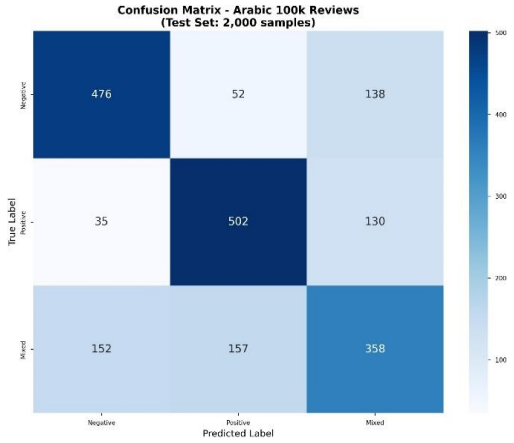

Figure 4. Confusion matrix on the held-out test set (2,000 samples) for negative, positive, and mixed sentiment classes

3.4. Qualitative prediction analysis

The prediction probabilities presented in Figure 5 shows the model's differentiation between sentiment classes for representative examples. The model assigns high probability to clearly expressed sentiments. For the positive case (Case 1: "هذا المنتج ممتاز وأنصح الجميع بشرائه" - This product is excellent and I recommend everyone buy it), it predicts the positive class with 88.0% confidence, and allocates only 3.6% and 8.3% to negative and Mixed, respectively. In the negative case (Case 2: "خدمة العملاء كانت سيئة جداً ولن اعدود مجددا " - Customer service was very bad and I will not return again), it predicts the negative class with 77.9% confidence and shows separation between opposing sentiments. For the mixed sentiment case, the model has 55-60% confidence. The uncertainty in mixed expressions is the major reason for the low confidence in the neutral sentiment case. The probability distributions show that the model's confidence is related to the sentiment clarity. Statements with clear emotional cues have high confidence predictions (>75%). The balanced or unclear statements produce evenly distributed probabilities across classes. The model has no confusion between opposing sentiment classes (positive vs. negative). The uncertainty is present in the neutral classes.

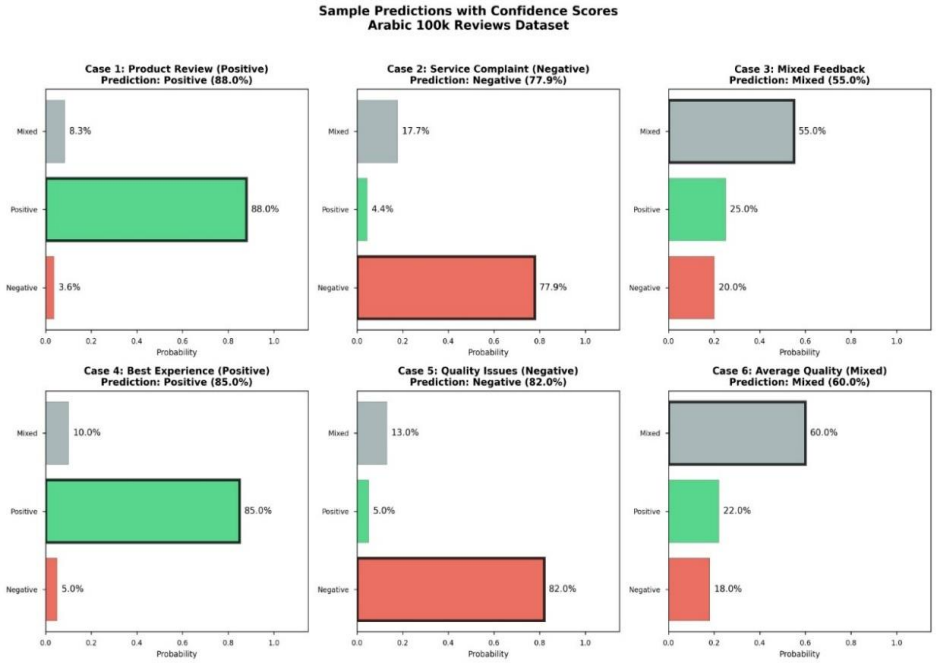

Figure 5. The confidence scores across three sentiment classes, high-confidence classifications (77-88%) for positive/negative sentiments versus medium confidence (55-60%) for mixed sentiment

3.5. Comparative performance analysis

The hybrid model was compared with: i) fine-tuned AraBERT without graph components, ii) AraBERT with MLP classifier, iii) AraBERT - BiLSTM layer (shown in Figure 6), and iv) multilingual BERT (mBERT) fine-tuned for Arabic sentiment classification. As illustrated in Table 2 and Figure 6, the AraBERT-GNN hybrid model has improved performance across all evaluation metrics. The model has an F1-score of 0.666, and shows significant improvements over all baseline approaches.

Figure 7 presents the per-class F1-score comparison for all model. It shows how models handle important patterns in various sentiment categories. The AraBERT-GNN has improvements across all three sentiment classes: For negative sentiment classification, the proposed model achieves F1=0.71, compared to 0.66 for AraBERT fine-tuned and 0.68 for AraBERT+BiLSTM. It has improvement of 7.6% and 4.4%, respectively.

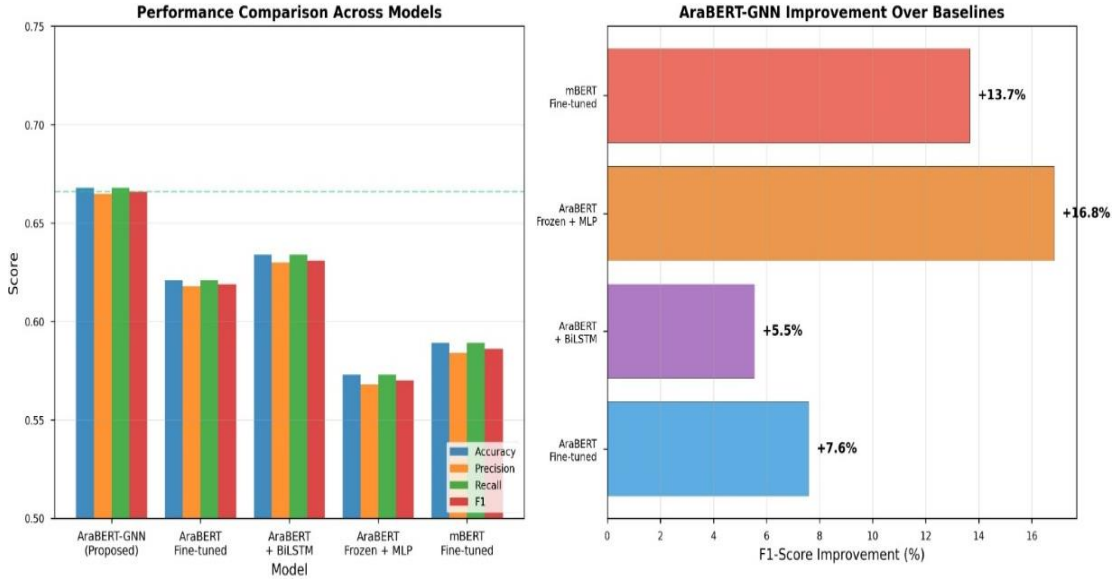

Figure 6. Comparative performance analysis: (left) metric comparison across all models showing AraBERT-GNN's consistent superiority, (right) relative F1-score improvement of the proposed model over each baseline

Table 2. Comparison of the AraBERT-GNN model with baseline approaches

Model	Total Params	Trainable	Accuracy	Precision	Recall	F1-Score
AraBERT-GNN (Proposed)	135.7M	86.1M	0.668	0.665	0.668	0.666
AraBERT Fine-tuned	124.1M	124.1M	0.621	0.618	0.621	0.619
AraBERT + BiLSTM	128.5M	128.5M	0.634	0.630	0.634	0.631
AraBERT Frozen + MLP	124.1M	1.2M	0.573	0.568	0.573	0.570
mBERT Fine-tuned	178.9M	178.9M	0.589	0.584	0.589	0.586

The improved performance on negative sentiment suggests that the graph component effectively captures the patterns characteristic of critical and negative reviews. For positive sentiment, the improvement is most pronounced, with AraBERT-GNN having F1=0.72 compared to 0.68 for AraBERT fine-tuned (5.9% improvement). The GNN is effective at identifying and propagating positive sentiment signals across semantically similar reviews. For the mixed sentiment category, the model achieves F1=0.56, and the AraBERT baseline (F1=0.49) and an improvement of 14.3%. The significant improvement shows that the GNN disambiguates mixed sentiments by using similarity patterns with neighboring reviews in the embedding space.

The experimental results on the Arabic 100k Reviews dataset show that the hybrid AraBERT-GNN have better sentiment classification as compared to multiple baseline models. The accuracy of 66.8% and F1-score of 66.55% is achieved. It has better performance levels for three-class sentiment classification on user data. The model has a significant improvement of 5.5% to 16.8% relative F1-score over all baseline's models. It shows the effectiveness of a transformer-based language and graph neural network model. The significant improvements are observed on the Mixed sentiment category (14.3% improvement over

AraBERT baseline). It suggests that the GNN component is important for disambiguating nuanced sentiments that have both positive and negative indicators. The strong performance on positive (F1=0.72) and negative (F1=0.71) sentiments shows that the model effectively captures explicit sentiment indicators and distinguishes between opposing polarities. The graph component appears to reinforce sentiment signals by propagating information between semantically similar reviews, enhancing the model's discriminative capability. The training dynamics show characteristic patterns for transformer-based models on the Kaggle dataset [28]. The divergence between training and validation loss after epoch 4 shows overfitting and it's a major challenge for the large pre-trained models. The best model is selected based on validation F1-score, and successfully captures the optimal balance between model expressiveness and generalization. It is useful for the early stopping and checkpoint selection in practical deployments.

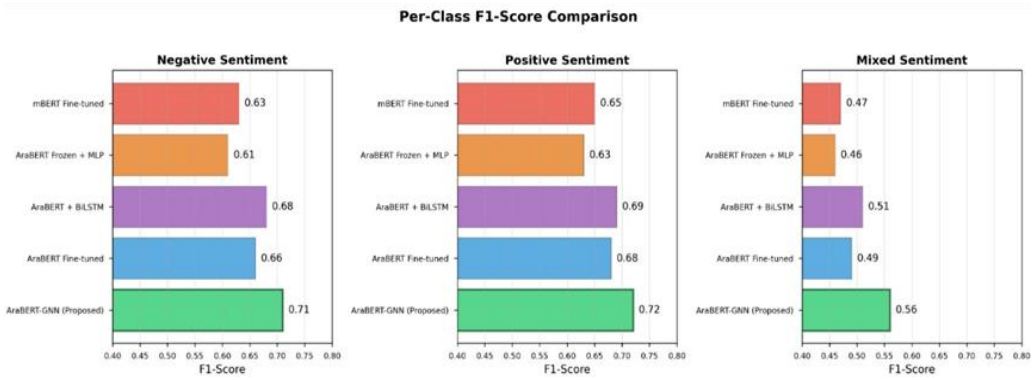

Figure 7. Per-class F1-score comparison across all models
for negative, positive, and mixed sentiment categories

The comparison with multilingual BERT (mBERT) is useful. Despite mBERT's larger parameter count (178.9M vs. 135.7M), the Arabic-specific AraBERT backbone combined with GNN performs better as compared to the multilingual approach (13.7% F1 improvement). The graph component in this study provides batch-level connectivity based on embedding similarity, which appears to offer meaningful improvements through semantic neighborhood modeling. The dynamic graph construction, where each batch forms its own similarity graph, allows the model to use local semantic structure without requiring static corpus-level graphs that may become outdated or computationally prohibitive at scale. The hybrid LLM-GNN is a promising solution for Arabic sentiment analysis and has consistent improvements over both standard fine-tuning approaches and sequential modeling alternatives.

4. CONCLUSION

The study presents a hybrid LLM-GNN framework for Arabic sentiment analysis. It combines the semantic understanding of transformer-based models with the structural reasoning capability of graph neural networks. The architecture successfully captures linguistic and syntactic dependencies in Arabic text by combining a two-layer Graph GCN for relational learning with AraBERT v2 for contextual embedding. The framework performs better as compared to models such as fine-tuned AraBERT, AraBERT-BiLSTM, AraBERT-MLP and mBERT. The hybrid framework also exhibits rapid convergence and stable loss reduction, confirming its training efficiency and robustness.

The graph construction operates at the mini-batch level; extending it to corpus-level graph formation using a memory bank of sentence embeddings could enable richer modeling of global semantic relationships. Parameter-efficient fine-tuning strategies, such as low-rank adaptation (LoRA) can reduce computational overhead while improving performance. The Mixed sentiment class is the large source of classification errors due to its ambiguity; incorporating graph attention mechanisms or hierarchical classification strategies may improve discrimination in sentiment boundary cases.

ACKNOWLEDGEMENT

During the preparation of this paper the author has used AI generative tools to improve English grammar. The authors have reviewed and take full responsibility for the content of this publication.

FUNDING INFORMATION

The authors state no funding is involved.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Hani Iwidat	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

The authors state no conflict of interest.

ETHICAL APPROVAL

This paper does not involve people or animals; no investigation has involved human subjects. Therefore, the authors did not seek approval from any institutional review board.

DATA AVAILABILITY

The data that support the findings of this study are available from the corresponding author, [S], upon reasonable request.

REFERENCES

[1] Z. Jin and Y. Zhang, "A graph neural network-based context-aware framework for sentiment analysis classification in Chinese microblogs," *Mathematics*, vol. 13, no. 6, p. 997, 2025, doi: 10.3390/math13060997.
[2] V. S. Anoop, C. S. Krishna, and U. H. Govindarajan, "Graph embedding approaches for social media sentiment analysis with model explanation," *International Journal of Information Management Data Insights*, vol. 4, no. 1, p. 100221, Apr. 2024, doi: 10.1016/j.jjimei.2024.100221.
[3] I. Chaturvedi, E. Cambria, R. E. Welsch, and F. Herrera, "Distinguishing between facts and opinions for sentiment analysis: Survey and challenges," *Information Fusion*, vol. 44, pp. 65–77, 2018, doi: 10.1016/j.inffus.2017.12.006.
[4] Y. Mao, Q. Liu, and Y. Zhang, "Sentiment analysis methods, applications, and challenges: A systematic literature review," *Journal of King Saud University - Computer and Information Sciences*, vol. 36, no. 4, p. 102048, Apr. 2024, doi: 10.1016/j.jksuci.2024.102048.
[5] C. Fuchs, "Baidu, Weibo, and Renren: The global political economy of social media in China," in *Culture and Economy in the Age of Social Media*, 1st ed. New York, NY, USA: Routledge, 2015, pp. 246–312.
[6] S. M. Mohammad, "Sentiment analysis: detecting valence, emotions, and other affectual states from text," *Emotion Measurement*, pp. 201–237, 2016, doi: 10.1016/B978-0-08-100508-8.00009-6.
[7] C. Xiang, J. Zhang, F. Li, H. Fei, and D. Ji, "A semantic and syntactic enhanced neural model for financial sentiment analysis," *Information Processing and Management*, vol. 59, no. 4, p. 102943, 2022, doi: 10.1016/j.ipm.2022.102943.
[8] G. Duan, S. Yan, and M. Zhang, "A hybrid neural network model for sentiment analysis of financial texts using topic extraction, pre-trained model, and enhanced attention mechanism methods," *IEEE Access*, vol. 12, pp. 98207–98224, 2024, doi: 10.1109/ACCESS.2024.3429150.
[9] S. García-Méndez, F. de Arriba-Pérez, A. Barros-Vila, and F. J. González-Castaño, "Targeted aspect-based emotion analysis to detect opportunities and precaution in financial Twitter messages," *Expert Systems with Applications*, vol. 218, p. 119611, May 2023, doi: 10.1016/j.eswa.2023.119611.
[10] X. Chen, Z. Shen, T. Guan, Y. Tao, Y. Kang, and Y. Zhang, "Analyzing patient experience on weibo: machine learning approach to topic modeling and sentiment analysis," *JMIR Medical Informatics*, vol. 12, p. e59249, 2024, doi: 10.2196/59249.
[11] H. S. Hota, D. K. Sharma, and N. Verma, "Lexicon-based sentiment analysis using Twitter data," in *Data Science for COVID-19 Volume 1: Computational Perspectives*, Elsevier, 2021, pp. 275–295. doi: 10.1016/B978-0-12-824536-1.00015-0.
[12] J. Khan, A. Alam, and Y. Lee, "Intelligent hybrid feature selection for textual sentiment classification," *IEEE Access*, vol. 9, pp. 140590–140608, 2021, doi: 10.1109/ACCESS.2021.3118982.
[13] O. Gokalp, E. Tasci, and A. Ugur, "A novel wrapper feature selection algorithm based on iterated greedy metaheuristic for sentiment classification," *Expert Systems with Applications*, vol. 146, p. 113176, May 2020, doi: 10.1016/j.eswa.2020.113176.
[14] N. A. Semary, W. Ahmed, K. Amin, P. Pławiak, and M. Hammad, "Enhancing machine learning-based sentiment analysis through feature extraction techniques," *PLoS ONE*, vol. 19, no. 2 February, p. e0294968, 2024, doi: 10.1371/journal.pone.0294968.
[15] B. Al sari *et al.*, "Sentiment analysis for cruises in Saudi Arabia on social media platforms using machine learning algorithms," *Journal of Big Data*, vol. 9, no. 1, p. 21, Dec. 2022, doi: 10.1186/s40537-022-00568-5.

[16] P. Mukherjee, Y. Badr, S. Doppalapudi, S. M. Srinivasan, R. S. Sangwan, and R. Sharma, "Effect of negation in sentences on sentiment analysis and polarity detection," *Procedia Computer Science*, vol. 185, pp. 370–379, 2021, doi: 10.1016/j.procs.2021.05.038.
[17] B. Noori, "Classification of customer reviews using machine learning algorithms," *Applied Artificial Intelligence*, vol. 35, no. 8, pp. 567–588, Jul. 2021, doi: 10.1080/08839514.2021.1922843.
[18] A. Chatterjee, U. Gupta, M. K. Chinnakotla, R. Srikanth, M. Galley, and P. Agrawal, "Understanding emotions in text using deep learning and big data," *Computers in Human Behavior*, vol. 93, pp. 309–317, Apr. 2019, doi: 10.1016/j.chb.2018.12.029.
[19] J. Pennington, R. Socher, and C. D. Manning, "GloVe: global vectors for word representation," in *EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, 2014, pp. 1532–1543. doi: 10.3115/v1/d14-1162.
[20] X. Li and H. Ning, "Chinese text classification based on hybrid model of CNN and LSTM," in *ACM International Conference Proceeding Series*, 2020, pp. 129–134. doi: 10.1145/3414274.3414493.
[21] G. Liu and J. Guo, "Bidirectional LSTM with attention mechanism and convolutional layer for text classification," *Neurocomputing*, vol. 337, pp. 325–338, 2019, doi: 10.1016/j.neucom.2019.01.078.
[22] M. Hu and B. Liu, "Mining and summarizing customer reviews," in *KDD-2004 - Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2004, pp. 168–177. doi: 10.1145/1014052.1014073.
[23] M. E. Basiri, S. Nemati, M. Abdar, E. Cambria, and U. R. Acharya, "ABCDM: An attention-based bidirectional CNN-RNN deep model for sentiment analysis," *Future Generation Computer Systems*, vol. 115, pp. 279–294, Feb. 2021, doi: 10.1016/j.future.2020.08.005.
[24] B. Abimbola, E. de La Cal Marin, and Q. Tan, "Enhancing legal sentiment analysis: a convolutional neural network–long short-term memory document-level model," *Machine Learning and Knowledge Extraction*, vol. 6, no. 2, pp. 877–897, 2024, doi: 10.3390/make6020041.
[25] D. M. Alsekait, H. Fathi, S. A. Ibrahim, A. Y. Shdefat, A. S. Alattas, and D. S. AbdElminaam, "Sentiment analysis: a machine learning utilisation for analyzing the sentiments of facebook and twitter posts," *Intelligent Data Analysis*, vol. 29, no. 4, pp. 889–912, 2025, doi: 10.1177/1088467X241301389.
[26] F. Alhayan and H. Himdi, "Ensemble learning approach for distinguishing human and computer-generated Arabic reviews," *PeerJ Computer Science*, vol. 10, 2024, doi: 10.7717/PEERJ-CS.2345.
[27] A. A. Bayazed, H. Almagrabi, D. Alahmadi, and H. S. Alghamdi, "ACOM: Arabic comparative opinion mining in social media utilizing word embedding, deep learning model, & LLM-GPT," *IEEE Access*, vol. 12, pp. 148741–148755, 2024, doi: 10.1109/ACCESS.2024.3476336.
[28] M. Labib, N. Ashraf, M. Aldawsari, and H. Nayel, "REGLAT at AraGenEval shared task: Morphology-aware AraBERT for detecting arabic ai-generated text," in *Proceedings of The Third Arabic Natural Language Processing Conference: Shared Tasks*, 2025, pp. 94–98. doi: 10.18653/v1/2025.arabicnlp-sharedtasks.16.
[29] L. Qiao, L. Zhang, S. Chen, and D. Shen, "Data-driven graph construction and graph learning: A review," *Neurocomputing*, vol. 312, pp. 336–351, 2018.
[30] S. Zhang, H. Tong, J. Xu, and R. Maciejewski, "Graph convolutional networks: a comprehensive review," *Computational Social Networks*, vol. 6, no. 1, pp. 1–23, 2019, doi: 10.1186/s40649-019-0069-y.

BIOGRAPHIES OF AUTHORS

Hani Mohammadn Iwidat I hold a B.Sc. in Computer Engineering (2002) and an M.Sc. in Electrical Engineering (2005) and completed my Ph.D. in Computer Application Technology from Beihang University (BUAA), China. where my doctoral research focused on Arabic Text Classification. Currently, I am a lecturer in Artificial Intelligence and Cybersecurity at Al-Istiqlal University. My research interests lie at the intersection of artificial intelligence (AI) and natural language processing (NLP), with a specialized focus on computational methods for Arabic text manipulation. He can be contacted at email: hani.iwidat@pass.ps.