Detecting Adverse Drug Reactions on Social Media Using Quantum Bi-LSTM with Attention

Dr. Ramesh Kumar Chuttugulla
Associate Professor
Department of CSE
Deccan College of Engineering and Technology
Affiliated to Osmania University
Hyderabad, Telangana
rameshkumar@deccancollege.ac.in

Syeda Qudeja
PG Scholar
Department of CSE
Deccan College of Engineering and Technology
Affiliated to Osmania University
Hyderabad, Telangana
squdeja@gmail.com

Abstract— **Combining drugs is common in treating diseases but raises the risk of adverse drug reactions (ADRs). Early detection of ADRs is essential for medication safety. Social media has emerged as a valuable source for identifying ADRs, but its data is massive, noisy, and sparse, making extraction challenging. Deep learning improves detection accuracy but requires heavy computation. Quantum computing, with its parallel processing and lower resource needs, offers a promising alternative. A new model, Quantum Bi-LSTM with Attention (QBi-LSTMA), integrates quantum computing and attention mechanisms into a Bi-LSTM network for ADR detection. The model uses six stacked variable quantum circuit components, simplifies the Bi-LSTM structure by removing gate biases, and efficiently updates network parameters with weight and activation qubits. Tested on the SMM4H dataset, QBi-LSTMA outperforms traditional and deep learning-based models, showing strong potential for large-scale ADR detection.**

Keywords— Adverse drug reactions (ADRs), Social media big data, quantum Bi-LSTM with attention (QBi-LSTMA), bi-directional long short-term memory (Bi-LSTM).

I. Introduction

Drug combinations are widely used in clinical practice [1], but they often cause adverse drug reactions (ADRs), especially as the number of drugs increases [2], [3]. ADRs can lengthen hospital stays, raise medical costs, and even increase mortality. While most reactions are mild, severe cases may cause shock or death [4]–[6]. ADRs are now the fourth leading cause of death in developed countries [7]–[9]. Margraff and Bertram [9] noted that patient self-reporting exists in 44 countries but accounts for only 9% of reports, with the majority reported by healthcare professionals. Dorji et al. [10] found that doctors and pharmacists are more aware of ADRs than nurses or traditional medicine practitioners, though overall reporting remains low.

Detecting ADRs is challenging because reactions may appear only under specific conditions or in certain populations, influenced by pharmacological, immune, genetic, and demographic factors [11]. Traditional detection relies on databases and expert review, which are time-consuming [11].

The rise of social media provides new opportunities for ADR detection [11], [12]. Unlike clinical trials and conventional monitoring, social media mining can reveal ADRs in real time [12], [13]. Ho et al. [14] reviewed two decades of ADR studies, highlighting data-driven techniques as particularly effective. Liu et al. [15] used lexical, syntactic, and semantic features for ADR extraction, while Azadeh et al. [16] classified methods by data source and evaluation metrics.

Deep learning has improved ADR detection by handling large datasets effectively [17]–[19]. Masino et al. [20] used CNNs to detect rare ADR mentions on Twitter. Tang et al. [21] combined LSTM with CRF for better F-scores, and Anne et al. [22] used RNNs with word embeddings for improved context-aware detection. Fan et al. [23] applied BERT to outperform prior models in ADR and medical entity extraction.

Despite progress, deep learning models require extensive computation and long training times. Quantum machine learning offers faster processing and better scalability [24]. Jia et al. [25] demonstrated neural networks representing quantum states, Cong et al. [26] introduced QCNNs, and Shekhar et al. [22] showed QLSTMs can learn patterns from social media efficiently.

Current methods still rely on manual feature engineering and shallow models. To address this, the Quantum Bi-LSTM with Attention (QBi-LSTMA) model was developed, combining quantum computing with Bidirectional LSTM and attention mechanisms. This approach accelerates training, reduces computational costs, and improves ADR detection on complex social media data.

Unlike conventional deep learning approaches, the proposed model minimizes dependence on pre-engineered linguistic features or high-quality input representations. It also operates efficiently without requiring massive computational power. The major contributions of this work are:

- The construction of a QBi-LSTMA model that integrates attention mechanisms with quantum computing in a Bi-LSTM framework.
- Improved performance and extraction accuracy compared to standard Bi-LSTM-based ADR detection methods.
- Extensive experimental validation to demonstrate the model’s robustness and reliability.

The remainder of this study includes an overview of related works such as Variational Quantum Circuits (VQC) and QLSTM in Section 2, a detailed description of the proposed ADR detection method in Section 3, followed by experiments and results in Section 4. Finally, Section 5 presents the conclusion and future research directions.

II. RELATED WORKS

A. VARIATIONAL QUANTUM CIRCUITS (VQC)

A Variational Quantum Circuit (VQC) is a special type of quantum circuit designed with adjustable parameters that can be fine-tuned through repeated optimization steps. It follows a hybrid quantum–classical framework, combining the unique advantages of both quantum and classical computing systems. This combination allows VQCs to achieve higher expressiveness compared to traditional neural networks, even when using fewer parameters. These parameters are continuously refined using iterative optimization methods performed by a classical computer.

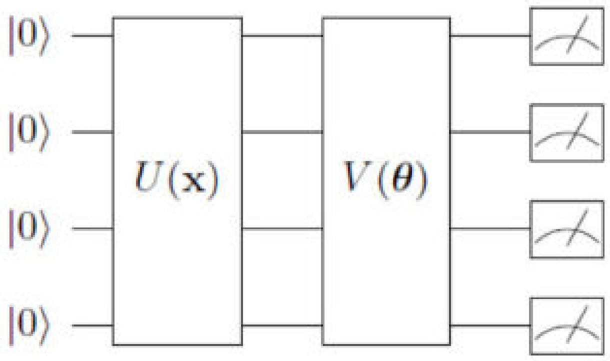


FIGURE 1. The general VQC structure

The general structure of a VQC is illustrated in Fig. 1. In this structure, $U(x)$ represents the quantum operation responsible for encoding classical input data x into the quantum state of the system. This part of the circuit is fixed and does not undergo optimization. On the other hand, $V(\theta)$ represents the variational circuit block that contains trainable parameters θ. These parameters are adjusted using gradient-based optimization techniques to minimize error and improve performance over successive iterations. Through this process, the VQC learns an optimal configuration that effectively captures the desired data patterns.

N-qubit state is denoted by

$$|\psi\rangle = \sum_{(q_1,q_2,\cdots,q_N\in\{0,1\})} c_{q_1,q_2,\cdots,q_N} |q_1\rangle \otimes |q_2\rangle \otimes \ldots \otimes |q_N\rangle \tag{1}$$

where Cq1,q2,...qN is the complex amplitude of each basis state and each quantum qi ∈ {0, 1}, the square of the amplitude Cq1,q2,...qN is the measurement probability of the measured state |q1⟩ ⊗ |q2⟩ . . . ⊗ | qN ⟩, and the total probability should sumto1,i.e.,

$$\sum_{(q_1,q_2,\ldots q_N)\in\{0,1\}} \left\| C_{q_1,q_2,\ldots,q_N} \right\|^2 = 1.$$

To convert the initial state, |0i ⊗ · · · ⊗ |0i to an unbiased state,

$$\begin{aligned}(H|0\rangle)^{\otimes N} &= \frac{1}{\sqrt{2^N}} (|0\rangle \otimes \cdots \otimes |0\rangle + \cdots + |1\rangle \otimes \cdots \otimes |1\rangle) \\ &\equiv \frac{1}{\sqrt{2^N}} \sum_{i=0}^{2^N-1} |i\rangle \end{aligned} \tag{2}$$

where i is the decimal number marking the corresponding bit string.

B. QUANTUM LONG SHORT-TERM MEMORY(QLSTM)

The Long Short-Term Memory (LSTM) network introduces three important gates — the input gate, the forget gate, and the output gate. Each gate plays a key role in managing information flow through the network. The input gate decides how much of the current input should be stored in the cell state. The forget gate determines how much information from the previous cell state should be retained or discarded in the next time step. The output gate controls how much of the updated cell state contributes to the final output at the current moment. Together, these mechanisms allow the LSTM to maintain both short-term and long-term dependencies over long sequences, enabling better understanding of temporal patterns. However, both LSTM and its bidirectional variant (Bi-LSTM) require training a large number of parameters, making the model computationally intensive. The overall structure of a standard LSTM cell is illustrated in Fig. 2A.

In a similar way, the Quantum Long Short-Term Memory (QLSTM) model integrates the principles of quantum computing into the conventional LSTM structure. This hybrid design aims to improve efficiency and representation power. As shown in Fig. 2B, the QLSTM cell is made up of six stacked Variational Quantum Circuits (VQCs). The activation functions σ and tanh represent the sigmoid and hyperbolic tangent nonlinear functions, respectively. Here, xt denotes the input at time t, ht represents the hidden state, ct stands for the cell state, and yt indicates the output. The symbols ⊗ and ⊕ refer to element-wise multiplication and addition operations. By combining classical LSTM dynamics with quantum computation, QLSTM aims to capture complex temporal relationships more efficiently and effectively.

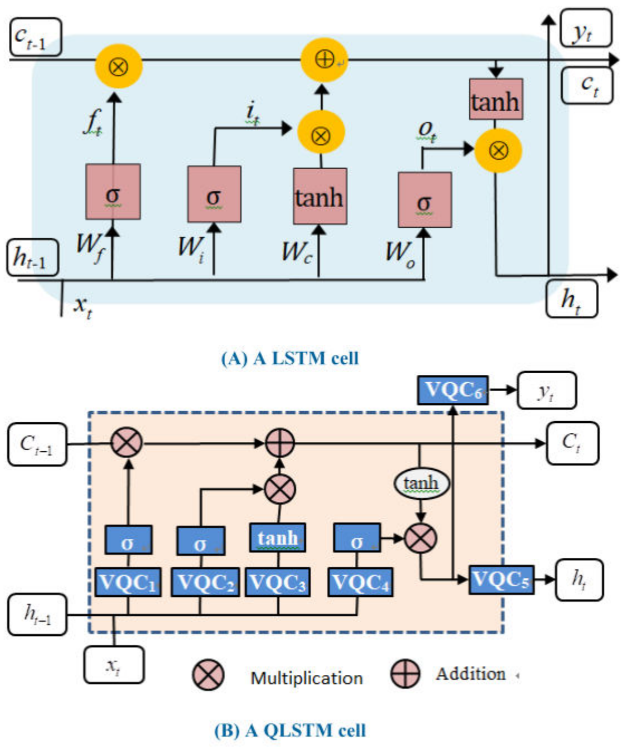


FIGURE 2. The architecture of LSTM and QLSTM

In QLSTM, the first variational quantum circuit, VQC1, is responsible for controlling the vector vt. It produces an output vector ft with values that lie within the interval [0, 1] after passing through the activation function σ. The second circuit, VQC2, also works on vt and applies σ to determine which values should be added to the cell state. VQC3 processes the same concatenated input and passes it through the tanh function to generate a new candidate for the cell state, denoted as $C{\sim}t$. The output from VQC2 is then multiplied element-wise with $C{\sim}t$. This resulting vector is finally used to update the current cell state.

VQC4 processes vt and applies σ to identify which parts of the cell state Ct are important for producing the output. The cell state itself is passed through tanh and then multiplied element-wise by the output from VQC4. After this, the resulting vector is further processed by either VQC5 or VQC6. VQC5 is used to transform the cell state Ct into the hidden state ht. Similarly, VQC6 transforms Ct into the output yt For VQC1 through VQC4, the input is the concatenation vt, which combines the previous hidden state $ht{-}1$ and the current input vector xt. Each circuit produces an output vector obtained from measurements at the end of each VQC. These measured values are designed to be the Pauli Z expectation values of each qubit. They are then processed using σ or tanh as appropriate. The process of feature extraction in this system is described as follows:

$$\begin{aligned}
f_t &= \sigma\left(VQC_1\left(v_t\right)\right); i_t = \sigma\left(VQC_2\left(v_t\right)\right)\\
\tilde{C}_t &= \tanh\left(VQC_3\left(v_t\right)\right)\\
c_t &= f_t * c_{t-1} + i_t * \tilde{C}_t, o_t = \sigma\left(VQC_4\left(v_t\right)\right)\\
h_t &= VQC_5\left(o_t * \tanh\left(c_t\right)\right)\\
y_t &= VQC_6\left(o_t * \tanh\left(c_t\right)\right) \qquad (3)
\end{aligned}$$

Unlike traditional LSTM networks, QLSTM achieves its iterative transfer relationships through four key components: weighting, activation, aggregation, and excitation. It takes the classical LSTM framework and extends it into the quantum domain by replacing the classical neural network elements inside the LSTM cells with parameterized quantum circuits, also known as VQCs.

To make the network structure simpler and more efficient, the main topology of the LSTM is preserved, but the biases in the input gate, forgetting gate, candidate memory unit, and output gate are eliminated. Instead, the network updates its weights by adjusting the weight and activation value qubits. This approach allows the network to adapt its parameters in a quantum-enhanced way. In many situations, QLSTM can operate with significantly fewer parameters compared to standard neural networks. This reduction in complexity makes them highly suitable for modeling intricate and dynamic environments. Furthermore, QLSTM networks can serve dual purposes: they not only extract important features from data but also perform data compression, offering a more efficient and powerful alternative to classical methods.

III. METHODOLOGY

Detecting adverse drug reactions (ADR) can be treated as a simple yes–no classification task. The QBi-LSTMA model was created for this challenge by combining Bi-LSTM, attention, and quantum computing. The ADR detection method uses several stages: input processing, embedding, QBi-LSTM feature extraction, attention weighting, and final Softmax classification. The approach assumes that if two drugs appear in the same sentence, their relationship can be inferred from other co-occurring pairs. As shown in Fig. 3, the full process runs through five main steps, which are explained in the following sections.

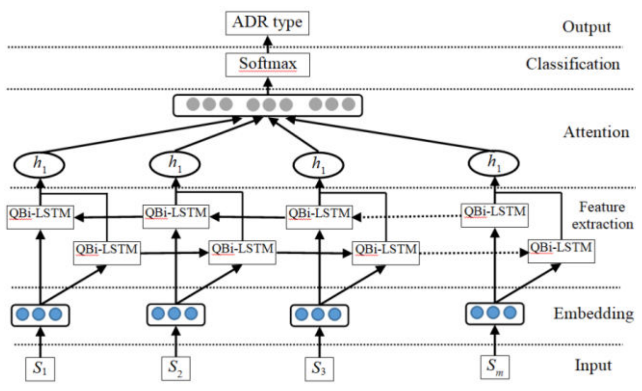


FIGURE 3. Proposed method flowchart

A. INPUT

The raw Twitter text often contains noise such as URLs, random symbols, and meaningless characters, which lowers ADR detection accuracy. To clean this data, the text is split into fine-grained tokens, and a custom stop-word list is applied to remove terms unrelated to drugs or adverse reactions. This reduces the search space and improves model performance. After preprocessing, the sentence set {S1, S2, …, Sm} is sent to the model. A Scrapy-based crawler is also used to collect tweets using their unique post IDs, and all retrieved data is stored in a single text file for consistent training and analysis.

. B. EMBEDDING

Text classification and clustering models require converting input text into fixed-length vectors. Traditional methods like bag-of-words and n-grams work, but they fail to fully capture the contextual, morphological, and shape features found in social media text. To address this, each word is transformed into a low-dimensional vector representing its key characteristics. The process uses word segmentation and word embedding. After segmentation, each word is mapped into three vectors: a word vector, an entity-type vector, and a part-of-speech vector. These are combined to form a single feature vector for the word. Word segmentation breaks a sentence into individual tokens, while embeddings convert them into machine-readable numerical vectors.

The final embedding for each word has a dimension d, defined as a model hyperparameter. After processing all words in a sentence, an embedding matrix $[E1,E2,\dots,En]$ is formed, where n is the maximum sentence length in the dataset. Repeating this for all m sentences produces a three-dimensional embedding matrix of size $m\times n\times d$, capturing semantic, morphological, and syntactic information effectively.

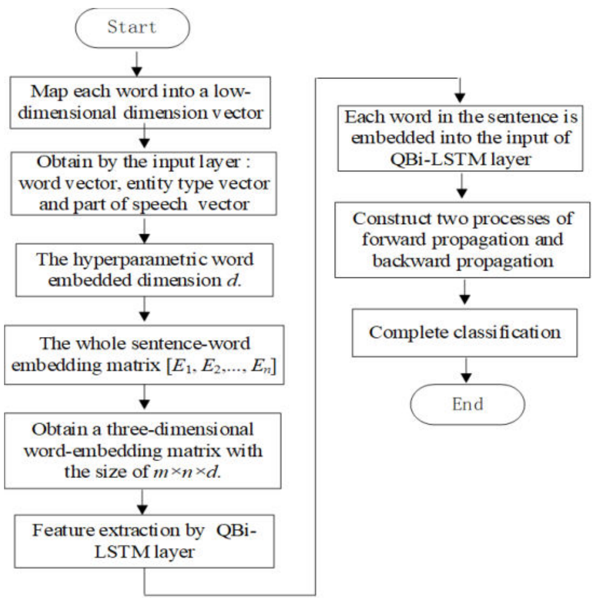

Start
Map each word into a low-dimensional dimension vector
Obtain by the input layer : word vector, entity type vector and part of speech vector
The hyperparametric word embedded dimension d.
The whole sentence-word embedding matrix $[E_1, E_2,\dots, E_n]$
Obtain a three-dimensional word-embedding matrix with the size of $m\times n\times d$.
Feature extraction by QBi-LSTM layer
Each word in the sentence is embedded into the input of QBi-LSTM layer
Construct two processes of forward propagation and backward propagation
Complete classification
End

FIGURE 4. Flow diagram for feature extraction

C. FEATURE EXTRACTION BY QBi-LSTM

QBi-LSTM is used to extract richer features from the text data described in Section B. Each sentence is first converted into word embeddings and then passed through the QBi-LSTM layer, which learns context from the entire sequence. Unlike a regular Bi-LSTM, the quantum-enhanced design improves memory and captures deeper word relationships. The model processes the sentence in both forward and backward directions, and the combined outputs provide a stronger and more complete representation for classification. FIGURE 4 shows the step-by-step flow of how the input sentence is transformed into meaningful features for later stages.

D. ATTENTION MECHANISM

A weight vector is generated and applied to the word-level outputs at each time step, merging them into a single sentence-level feature. The attention layer then assigns importance scores to the QBi-LSTM states, focusing on the most relevant parts of the sequence. By highlighting these key signals, the model improves the accuracy of ADR detection. The attention process works as follows:

$$M = \tanh(H), \quad \alpha = soft\max\left(w^T M\right)$$
$$r = H\alpha^T, \quad H^* = \tanh(r) \tag{4}$$

H represents all the word features extracted from a sentence using the QBi-LSTM model. In contrast, w is the parameter vector the model learns during training. Simply put, H holds the sentence information, while w determines how the model learns from it.

E. CLASSIFICATION AND OUTPUT

The attention outputs are first merged through concatenation and then passed to a Softmax layer. This layer computes the probability that each candidate drug triggers an adverse reaction. In simple terms, it predicts the likelihood $y=c$ for each drug. This sequential flow—from attention to concatenation to classification—helps the model identify which drug is most likely to cause an ADR.

$$P(y=c) = soft\max(w\cdot\gamma + b)$$
$$\bar{y} = \arg\max_{y\in c} P(y=c) \tag{5}$$

In this model, W and b denote the weight matrix and bias term. The set C includes all possible ADR labels, and the label with the highest predicted probability is selected as y for the given drug. The QBi-LSTM has many trainable parameters, which are optimized using gradient descent. For each batch of L samples, the loss gradients are computed through the chain rule, and the parameters are updated using the learning rate λ. This cycle repeats until the model converges to a stable and satisfactory performance.

$$Loss = \sum_{i=1}^{L} -\log p\left(x_i|y_i\right), \theta = \theta - \lambda \frac{\partial Loss}{\theta} \qquad (6)$$

θ denotes the model's parameter. A fixed learning rate λ can make training unstable, causing the loss to bounce instead of converging smoothly. This leads to an unreliable and harder-to-control training process.

Corpus set	Samples at publication	Available samples	Positive	Negative
TwiMed	1000	608	234	374
TwitterADR	10 822	6 966	765	6 201

TABLE 1. Statistical Information of two Corpuses

IV. EXPERIMENTS AND RESULTS

To assess the performance of the proposed QBi-LSTM method for ADR detection, a series of comparative experiments were conducted on social medical data. The outcomes were evaluated against several existing techniques, including LNS, CNNWEF, RNN, and ATT-RNN. These models represent traditional statistical methods, convolutional networks with word embeddings, and attention-enhanced recurrent networks. The comparison clearly illustrates where the QBi-LSTM approach offers improvements over current ADR detection methods.

A. DATASET DESCRIPTION

Twitter is an appealing source for ADR analysis because of its large and varied user base, where millions share daily experiences. Yet ADR-related posts make up only a tiny fraction of all tweets, making manual review unrealistic. To handle this, we used two Twitter-based datasets: TwiMed and TwitterADR, both annotated with user IDs and category labels. As shown in Table 1, only about 60% of posts in each set are usable. TwiMed has a nearly balanced mix of positive and negative samples (about 1:1.6), while TwitterADR is larger and much more imbalanced, with roughly a 1:8.1 ratio. These differences can influence how well models learn and generalize.

B. EXPERIMENTAL SETTING

The experiments run on a system with 32 GB RAM, an Intel i5-4200U (2.30 GHz), and a GTX 1080 Ti GPU. Ubuntu 14.0 is used, along with TensorFlow 1.7.0 and Keras. The model includes LSTM layers, trained with an initial learning rate of 0.001 and a decay factor of 0.1. Training continues for 300 epochs with a batch size of 10 and a dropout rate of 0.5. All other parameters are randomly initialized to support stable learning.

C. SYSTEM CONFIGURATION

The experiments followed a ten-fold cross-validation setup. For each fold, the data was split into training and test sets, with the training portion further divided into 90% training and 10% validation. Word embeddings of size 100 were used, while the embedding layer operated with 50 dimensions. The model was trained using stochastic gradient descent with cross-entropy loss. Training stopped automatically if the validation performance failed to improve for five epochs, ensuring stable and reliable evaluation.

D. EVALUATION METRICS

The QBi-LSTMA model's performance is assessed using precision, recall, and F1-score for the positive class, which represents instances containing ADR descriptions. Precision shows how many predicted positives are correct, while recall shows how many actual positives the model successfully finds. The F1-score, calculated as 2PR/(P+R), combines both measures to give a balanced view. Together, these metrics indicate how effectively the model detects ADR-related content.

E. RESULTS

The impact of stacked QBi-LSTM layers on TwiMed performance is evaluated, as shown in Figure 5. Precision increases as layers are added, with four layers giving the best results. Beyond this point, performance stops improving and may even decline due to training difficulty and overfitting. Therefore, all later experiments use four stacked layers for stable and efficient learning.

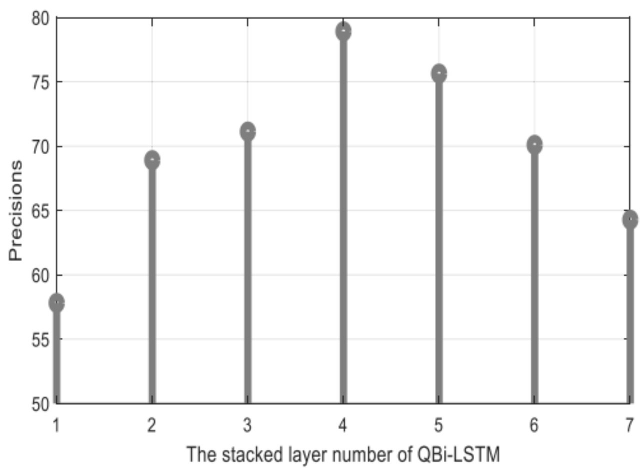


FIGURE 5. Precision versus the stacked layer number of QBi-LSTM.

The model's convergence was analyzed across different iteration counts. As shown in Figure 6, the training and testing losses for QBi-LSTMA on the TwiMed dataset steadily decrease with more iterations. Loss values drop sharply in the early phase and start to level off after about 2000 iterations. The model at the 2500th iteration is chosen for training, as it offers stable performance. Overall, the training loss remains slightly lower than the testing loss, indicating good learning without overfitting.

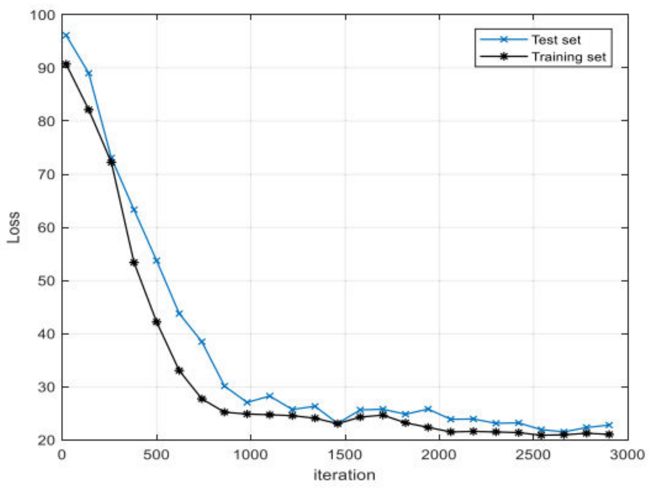


FIGURE 6. Losses with the different number of iterations

The attention mechanism enhances the model by enforcing semantic alignment. Comparing QBi-LSTM with attention (QBi-LSTMA), without attention, and standard Bi-LSTM (Figure 7), QBi-LSTMA clearly performs best. Attention helps capture features across characters, words, sentences, and contexts, improving entity recognition via richer word representations. Additionally, quantum networks in QBi-LSTM provide strong nonlinear approximation, fast convergence, and better generalization than classical Bi-LSTM.

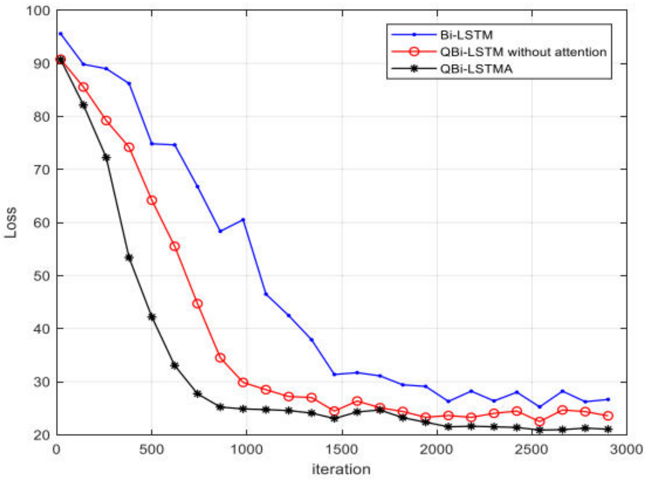


FIGURE 7. Losses with the different number of iterations by QBi-LSTMA, Bi-LSTM and QBi-LSTM without attention

We tested various embedding sizes with QBi-LSTMA to improve recognition of ADR-related text from multiple sources. Figure 8 shows that embedding length significantly affects performance. Short embeddings miss semantic detail, while longer ones may add noise. Experiments show that a length of 50 offers the best balance.

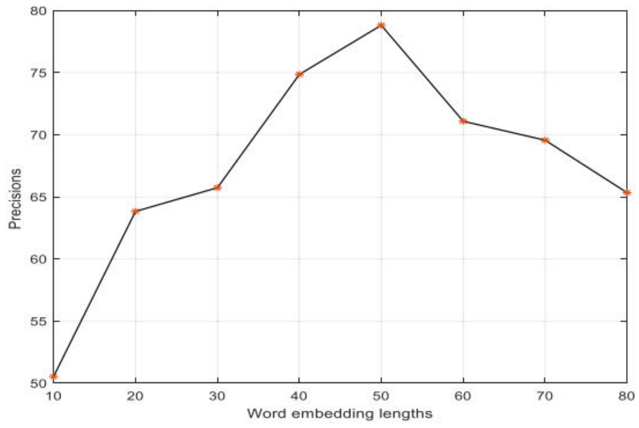


FIGURE 8. Precisions by QBi-LSTMA versus word embedding lengths

Figures 5–8 show the finalized QBi-LSTMA setup with four stacked layers, 2500 iterations, and 50-dimensional word embeddings. Ten-fold cross-validation is repeated ten times, and the average results are reported in Tables 2 and 3. Table 2 also compares parameter sizes across four network variants to assess training efficiency.

Method / Results	LNS	CNNWEF	RNN	ATT-RNN	QBi-LSTMA
P	32.37	58.54	62.35	75.10	78.41
R	66.30	61.35	71.24	71.12	69.38
F1	43.50	59.91	66.50	72.65	73.62

embedding lengths

TABLE 2. The ADR detection results on TwiMed by LNS [32], CNNWEF [20], RNN [22], ATT-RNN [28] and QBi-LSTMA

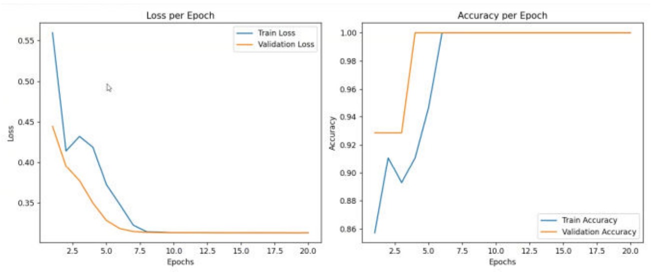


FIGURE 9. Loss per Epoch and Accuracy per Epoch

Analysis of results highlights the strengths and weaknesses of LNS, CNNWEF, RNN, ATT-RNN, and QBi-LSTMA, as summarized in Table 4. Comparing the results in Tables 2 to 4 shows that QBi-LSTMA achieves the highest performance in ADR detection. Deep learning-based methods generally outperform the traditional LNS approach, while RNN and ATT-RNN surpass CNNWEF. The reasons for this are clear.

Method / Results	LNS	CNNWEF	RNN	ATT-RNN	QBi-LSTMA
P	58.23	63.05	61.21	67.13	66.25
R	54.11	61.37	51.49	65.64	65.71
F1	56.09	62.20	56.01	65.67	65.98
Parameter	--	42.71MB	2.70MB	4.04MB	1.83MB

TABLE 3. The ADR detection results on TwitterADR by LNS [32],CNNWEF [20], RNN [22], ATT-RNN [28] and QBi-LSTMA.

LNS is simple, with few trainable parameters and no memory, resulting in low accuracy. CNNWEF excels at feature extraction but struggles with memory and text tasks like ADR detection. RNN captures short-term dependencies, while ATT-RNN improves focus via attention, yet both suffer from limited memory and gradient vanishing.

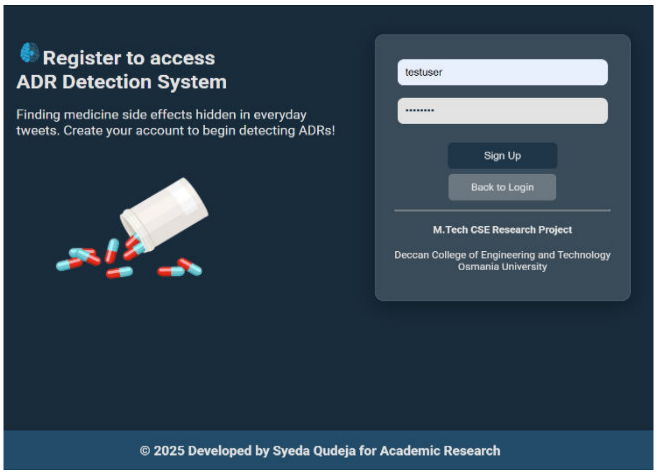
QBi-LSTMA overcomes these issues. Like LSTM, it retains long-term information using cell states and manages dependencies with four parameter matrices. By removing gate biases and using quantum-inspired weights and activation qubits, it improves learning efficiency. Unlike CNNWEF, it handles positional and semantic information effectively, making it better suited for complex text classification.

Method / Merit & demerit	LNS	CNNWEF	RNN	ATT-RNN	QBi-LSTMA
Accuracy	Lowest	Low	Low	High	Highest
Training time	Least	Most	Medium	Less	Least
Memory	Weakest	Weak	Medium	Strongest	Strong
Feature learning ability	Weakest	Strongest	Medium	Strong	Strong
Generalization	Weakest	Weak	Medium	Strong	Strong

TABLE 4. The characteristics of Bi-LSTM, Bi-LSTMA and QBi-LSTMA.

QBi-LSTMA excels by combining Bi-LSTM, attention, and quantum computation, offering better nonlinear approximation and generalization than traditional LSTM models. With just six VQCs, it keeps parameter use low. Overall, it outperforms other models in ADR detection, highlighting the advantage of blending attention, deep learning, and quantum-inspired methods.

The ADR Detection System is a web-based application designed to identify adverse drug reactions in social media content. Users begin by registering an account with their credentials, which they then use to log in to the platform. Upon successful authentication, they access the system's homepage. From there, users can input a tweet into a designated textbox and initiate the prediction process. The system analyzes the tweet content using machine learning algorithms to determine whether it contains indicators of an adverse drug reaction, classifying it as either "ADR" or "Not ADR." Users can test multiple tweets by clicking the Back button to return to the input screen and submitting additional content for analysis, allowing for iterative exploration of the system's detection capabilities.


Register to access ADR Detection System

Finding medicine side effects hidden in everyday tweets. Create your account to begin detecting ADRs!

testuser

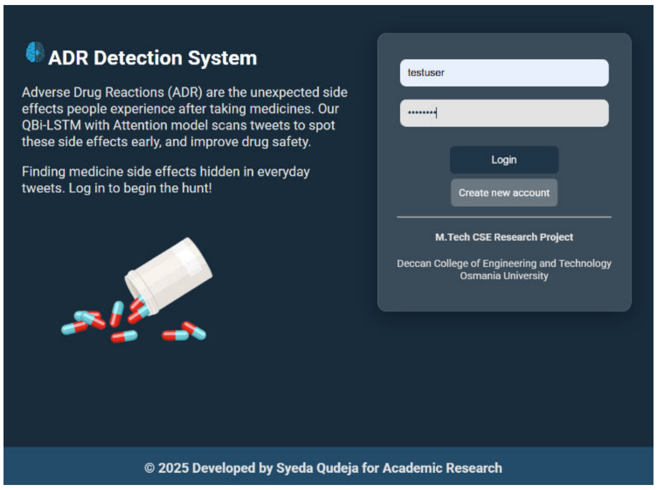
••••••••

M.Tech CSE Research Project

Deccan College of Engineering and Technology Osmania University

© 2025 Developed by Syeda Qudeja for Academic Research

FIGURE 10. Register into ADR Detection System


ADR Detection System

Adverse Drug Reactions (ADR) are the unexpected side effects people experience after taking medicines. Our QBi-LSTM with Attention model scans tweets to spot these side effects early, and improve drug safety.

Finding medicine side effects hidden in everyday tweets. Log in to begin the hunt!

testuser

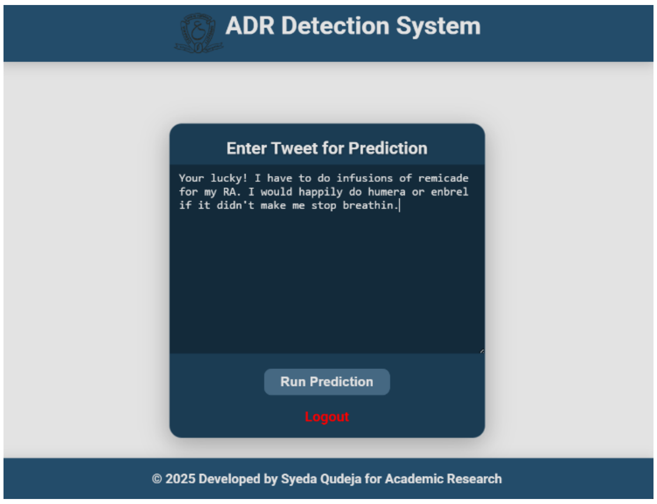
••••••••

M.Tech CSE Research Project

Deccan College of Engineering and Technology Osmania University

© 2025 Developed by Syeda Qudeja for Academic Research

FIGURE 11. Login into ADR Detection System

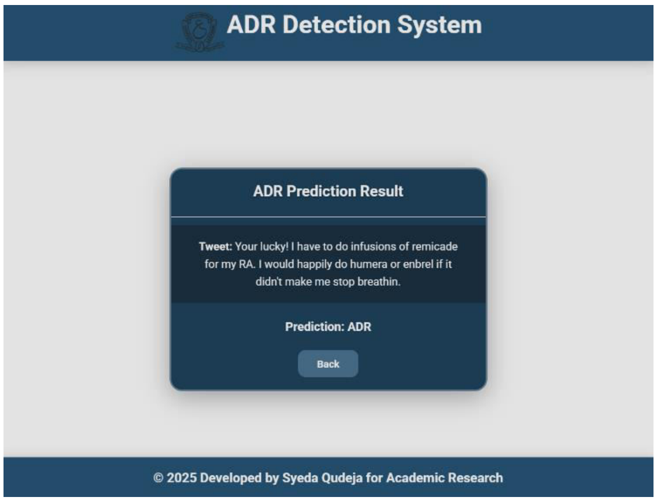

ADR Detection System

Enter Tweet for Prediction

Your lucky! I have to do infusions of remicade for my RA. I would happily do humera or enbrel if it didn't make me stop breathin.

© 2025 Developed by Syeda Qudeja for Academic Research

FIGURE 12. Enter Tweet for Prediction


ADR Detection System

ADR Prediction Result

Tweet: Your lucky! I have to do infusions of remicade for my RA. I would happily do humera or enbrel if it didn't make me stop breathin.

Prediction: ADR

© 2025 Developed by Syeda Qudeja for Academic Research

FIGURE 13. ADR Prediction Result

V. Conclusion

Adverse Drug Reactions (ADRs) are a major threat to patient safety and a key concern in drug development. Detecting all possible ADRs is difficult due to their complexity and the vast, noisy data available on social media. To tackle this, a QBi-LSTMA-based approach combines Bi-LSTM networks, attention mechanisms, and quantum computing to enhance accuracy, efficiency, and robustness. Experiments show strong performance in ADR detection. Future work will focus on optimizing the model for mobile devices, exploring multi-stage classifiers, and assessing its generalization and computational efficiency in real-world settings.

References

[1] A. Poleksic and L. Xie, "Database of adverse events associated with drugs and drug combinations," Sci. Rep., vol. 9, no. 1, p. 20025, Dec. 2019, doi: 10.1038/s41598-019-56525-5.

[2] P. Crits-Christoph, M. G. Newman, K. Rickels, R. Gallop, M. B. C. Gibbons, J. L. Hamilton, S. Ring-Kurtz, and A. M. Pastva, "Combined medication and cognitive therapy for generalized anxiety disorder," J. Anxiety Disorders, vol. 25, no. 8, pp. 1087–1094, Dec. 2011.

[3] R. Harpaz, H. S. Chase, and C. Friedman, "Mining multi-item drug adverse effect associations in spontaneous reporting systems," BMC Bioinf., vol. 11, no. S9, pp. 1–8, Oct. 2010, doi: 10.1186/1471-2105-11-S9-S7.

[4] H. Yang and C. C. Yang, "Discovering drug-drug interactions and associated adverse drug reactions with triad prediction in heterogeneous healthcare networks," in Proc. IEEE Int. Conf. Healthcare Informat. (ICHI), Oct. 2016, pp. 244–254.

[5] J. Strandell, O. Caster, J. Hopstadius, I. R. Edwards, and G. N. Norén, "The development and evaluation of triage algorithms for early discovery of adverse drug interactions," Drug Saf., vol. 36, no. 5, pp. 371–388, May 2013.

[6] M. Liu, Y. Hu, and B. Tang, "Role of text mining in early identification of potential drug safety issues," in Biomedical Literature Mining. New York, NY, USA : Humana Press, 2014, pp. 227–251, doi: 10.1007/978-1-4939-0709-0_13.

[7] S. V. Iyer, R. Harpaz, P. LePendu, A. Bauer-Mehren, and N. H. Shah, "Mining clinical text for signals of adverse drug-drug interactions," J. Amer. Med. Inform. Assoc., vol. 21, no. 2, pp. 353–362, Mar. 2014.

[8] D. A. Nguyen, C. H. Nguyen, and H. Mamitsuka, "A survey on adverse drug reaction studies: Data, tasks and machine learning methods," Briefings Bioinf., vol. 22, no. 1, pp. 164–177, Jan. 2021, doi: 10.1093/bib/bbz140.

[9] F. Margraff and D. Bertram, "Adverse drug reaction reporting by patients: An overview of fifty countries," Drug Saf., vol. 37, no. 6, pp. 409–419, Jun. 2014.

[10] C. Dorji, P. Tragulpiankit, A. Riewpaiboon, and T. Tobgay, "Knowledge of adverse drug reaction reporting among healthcare professionals in Bhutan: A cross-sectional survey," Drug Saf., vol. 39, no. 12, pp. 1239–1250, Dec. 2016.

[11] Y. Zhou, V. Miller, M. Hogan, and L. Callahan, "An overview of adverse drug reaction monitoring in China," Int. J. Pharmaceutical Med., vol. 20, no. 2, pp. 79–85, 2006.

[12] A. Sarker, R. Ginn, A. Nikfarjam, K. O'Connor, K. Smith, S. Jayaraman, T. Upadhaya, and G. Gonzalez, "Utilizing social media data for pharmacovigilance: A review," J. Biomed. Informat., vol. 54, pp. 202–212, Apr. 2015.

[13] I. Korkontzelos, A. Nikfarjam, M. Shardlow, A. Sarker, S. Ananiadou, and G. H. Gonzalez, "Analysis of the effect of sentiment analysis on extracting adverse drug reactions from tweets and forum posts," J. Biomed. Informat., vol. 62, pp. 148–158, Aug. 2016.

[14] T.-B. Ho, L. Le, D. T. Thai, and S. Taewijit, "Data-driven approach to detect and predict adverse drug reactions," Current Pharmaceutical Des., vol. 22, no. 23, pp. 3498–3526, Jun. 2016.

[15] J. Liu, S. Zhao, and X. Zhang, "An ensemble method for extracting adverse drug events from social media," Artif. Intell. Med., vol. 70, pp. 62–76, Jun. 2016.

[16] A. Nikfarjam, A. Sarker, K. O'Connor, R. Ginn, and G. Gonzalez, "Pharmacovigilance from social media: Mining adverse drug reaction mentions using sequence labeling with word embedding cluster features," J. Amer. Med. Inf. Assoc., vol. 22, no. 3, pp. 671–681, May 2015.

[17] S. U. Khan, T. Hussain, A. Ullah, and S. W. Baik, "Deep-ReID: Deep features and autoencoder assisted image patching strategy for person re-identification in smart cities surveillance," Multimedia Tools Appl., pp. 1–22, Jan. 2021. [Online]. Available: https://link.springer.com/article/10.1007/s11042-020-10145-8#citeas, doi: 10.1007/s11042-020-10145-8.

[18] S. U. Khan, I. U. Haq, S. Rho, S. W. Baik, and M. Y. Lee, "Cover the violence: A novel deep-learning-based approach towards violence-detection in movies," Appl. Sci., vol. 9, no. 22, p. 4963, Nov. 2019.

[19] E. Florez, F. Precioso, R. Pighetti, and M. Riveill, "Deep learning for identification of adverse drug reaction relations," in Proc. Int. Symp. Signal Process. Syst., 2019, pp. 149–153.

[20] A. J. Masino, D. Forsyth, and A. G. Fiks, "Detecting adverse drug reactions on Twitter with convolutional neural networks and word embedding features," J. Healthcare Inform. Res., vol. 2, nos. 1–2, pp. 25–43, Jun. 2018.

[21] B. Tang, J. Hu, X. Wang, and Q. Chen, "Recognizing continuous and discontinuous adverse drug reaction mentions from social media using LSTM-CRF," Wireless Commun. Mobile Comput., vol. 2018, pp. 1–8, Apr. 2018.

[22] A. Cocos, A. G. Fiks, and A. J. Masino, "Deep learning for pharmacovigilance: Recurrent neural network architectures for labeling adverse drug reactions in Twitter posts," J. Amer. Med. Inform. Assoc., vol. 24, no. 4, pp. 813–821, Jul. 2017.

[23] B. Fan, W. Fan, and C. Smith, "Adverse drug event detection and extraction from open data: A deep learning approach," Inf. Process. Manage., vol. 57, no. 1, Jan. 2020, Art. no. 102131, doi: 10.1016/j.ipm.2019.102131.

[24] A. Sordoni, J.-Y. Nie, and Y. Bengio, "Modeling term dependencies with quantum language models for IR," in Proc. 36th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr., Dublin, Ireland, Jul. 2013, pp. 653–662.

[25] Z. A. Jia, B. Yi, R. Zhai, Y.-C. Wu, G.-C. Guo, and G.-P. Guo, "Quantum neural network states: A brief review of methods and applications," Adv. Quantum Technol., vol. 2, nos. 7–8, 2019, Art. no. 1800077, doi: 10.1002/qute.201800077.

[26] I. Cong, S. Choi, and D. L. Mikhail, "Quantum convolutional neural networks," Nature Phys., vol. 15, no. 12, pp. 1273–1278, 2019.

[27] S. Shekhar, D. K. Sharma, and M. M. S. Beg, "Language identification framework in code-mixed social media text based on quantum LSTM—The word belongs to which language? " Modern Phys. Lett. B, vol. 34, no. 6, Feb. 2020, Art. no. 2050086, doi: 10.1142/S0217984920500864.

[28] C. Du and L. Huang, "Text classification research with attention-based recurrent neural networks," Int. J. Comput. Commun. Control, vol. 13, no. 1, p. 50, Feb. 2018.

[29] N. Alvaro, Y. Miyao, and N. Collier, “TwiMed: Twitter and PubMed comparable corpus of drugs, diseases, symptoms, and their relations,” JMIR Public Health Surveill., vol. 3, no. 2, p. e24, May 2017.

[30] A. Sarker and G. Gonzalez, “Portable automatic text classification for adverse drug reaction detection via multi-corpus training,” J. Biomed. Informat., vol. 53, pp. 196–207, Feb. 2015.

[31] A. Sarker, A. Nikfarjam, and G. Gonzalez, “Social media mining shared task workshop,” in Proc. Pacific Symp. Biocomputing, 2016, pp. 581–592, doi: 10.13140/RG.2.1.3886.7924.

[32] W. Zhang, X. Yue, F. Liu, Y. Chen, S. Tu, and X. Zhang, “A unified frame of predicting side effects of drugs by using linear neighborhood similarity,” BMC Syst. Biol., vol. 11, no. S6, pp. 23–34, Dec. 2017, doi: 10.1186/s12918-017-0477-2.

AUTHORS BIOGRAPHY

Ch. Ramesh Kumar, Ph.D., is an Associate Professor of Computer Science and Engineering at Deccan College of Engineering and Technology, Hyderabad, Telangana, India. He has several international publications to his credit. His research interests include software reuse, software performance, software testing, data mining, and cloud computing.

Syeda Qudeja is currently pursuing her Master of Technology in Computer Science and Engineering at Deccan College of Engineering and Technology, affiliated with Osmania University, Hyderabad, Telangana, India. Her research interests include deep learning, big data, and pharmacovigilance.