JOURNAL OF SOFT COMPUTING AND DATA MINING
e-ISSN: 2716-621X

JSCDM

Vol. 6 No. 1 (2025) 95-106
https://publisher.uthm.edu.my/ojs/index.php/jscdm

Predicting Conversations: Innovating Chatbot Technology with a Hybrid PSO-LSTM Deep Learning Model

Huda W. Ahmed[1], Maysaa H. Abdulameer[1], Ruwaida M. Yas[1]*, Massila Kamalrudin[2], Mustafa Musa[3]

[1] *Informatics Institute for Post Graduate Studies, University of Information Technology and Communications, Baghdad, IRAQ*
[2] *Faculty of Information and Communication Technology, Universiti Teknikal Malaysia, Melaka, MALAYSIA*
[3] *Center of Research and Innovation Management, Universiti Teknikal Malaysia, Melaka, MALAYSIA*

*Corresponding Author: roueida.m.yas@iips.edu.iq
DOI: https://doi.org/10.30880/jscdm.2025.06.01.007

Article Info

Received: 12 February 2025
Accepted: 4 June 2025
Available online: 30 June 2025

Keywords

Deep learning, PSO, Bi_LSTM, Chatbot

Abstract

Artificial intelligence has quickly evolved and dramatically changed how humans interact with computers, particularly through chatbots that have become essential in areas like customer support, digital assistants, and web-based help services. However, even with these technological improvements, chatbots still face challenges in understanding and appropriately responding to subtle, complex conversations that depend heavily on context. This research presents a novel hybrid approach combining Particle Swarm Optimization (PSO) with Bidirectional Long Short-Term Memory (Bi-LSTM) networks to enhance chatbot performance in natural language understanding and response generation. The proposed model leverages the memory capabilities of Bi-LSTM for sequential pattern recognition while utilizing PSO for hyperparameter tuning and weight optimization. By integrating these methods, the model achieves significant improvements in training accuracy (0.9961), validation accuracy (0.9854), and testing accuracy (0.9882) compared to conventional Bi-LSTM networks. The evaluation, conducted on a real-world conversational dataset, demonstrates the model's effectiveness through various metrics, including BLEU, METEOR, Word Error Rate (WER), and F1-score, establishing its robustness and reliability for intelligent chatbot systems.

1. Introduction

Anyone can reach university websites and their information tools, no matter where they are or what time it is. To improve these services, institutions need strong infrastructure for their buildings, especially services like robots that let people access data and information 24 hours a day, 7 days a week [1]. People think that chatbots are one of the easiest ways to communicate because they use pattern matching rules to let users have conversations. Even though they are useful, these systems don't always have advanced language processing features. Instead, they are made up of modules that can be put together with other software to help with development. They need certain language traits to be able to chat and reply at the same time, which makes them good subjects for scientific studies [2]. Even though chatbot technology has come a long way, talking systems still have a hard time figuring out what users want and keeping up a useful conversation over long periods of time. Traditional deep learning methods, like recurrent neural networks (RNNs) and Long Short-Term Memory (LSTM) architectures, often have problems like decreasing gradients and choosing bad solutions because the parameters are not set up correctly. This makes chatbots less useful and negatively impacts the user experience [26][27]. Also, changing hyperparameters by hand and relying on traditional optimization methods often leads to operational inefficiencies and disappointing results in real-life talking situations [28][29]. Neural networks are very important for clever chatbots because they help them understand and respond to normal language. But how well these networks work depends a lot on how they are built and how their parameters are set [3]. Particle Swarm Optimization (PSO) is a methodical approach to improving these factors by making solutions better over time based on both individual and group performance measures. This method lowers the amount of work that needs to be done on the computer compared to trial-and-error methods while still making sure that the robot can have solid conversations [4, 5]. Learn from changing sets of data, which is better for chatbots that have been improved with PSO-optimized neural networks [6]. Hybrid techniques, like mixing PSO with other optimization methods like genetic algorithms, can also improve the quality of the solutions, which makes these chatbots more useful and flexible [7]. PSO is inspired by how organisms like fish and birds act when they live together: they share information to find the best answers. This method works especially well for optimizing neural networks because it can fine-tune weights, biases, and hyperparameters to make learning faster and more accurate [8]. When developers use PSO, they can fix problems that happen with standard backpropagation-based training methods, like overfitting and convergence to local minima [1, 9].

The study aims to fill in these gaps by suggesting a new method that combines (PSO) and (Bi-LSTM) networks. PSO was added because it has been shown to systematically and efficiently improve neural network parameters such as weights, biases, and hyperparameters. This helps reduce issues like overfitting and local minima convergence that come up with traditional training methods [30, 31]. Bi-LSTM networks, which are great at finding two-way relationships in sequential data, work well with PSO's optimization skills to make sure the chatbot can understand talks that are more complicated because of the context [32, 33]. This combined model offers big changes in conversational accuracy, response relevance, and user involvement, which will push the limits of what chatbots can do now. As research progresses, this approach is expected to further streamline chatbot development, ensuring scalability and adaptability in real-world applications [7][9]. As chatbots continue to evolve into indispensable tools in various sectors, the application of optimization methods and deep neural networks not only improves their efficiency but also pushes the boundaries of conversational AI, making interactions more human-like and contextually relevant [10][11].

2. Related Works

We have a number of practical studies that highlighted the implementation of a chatbot, the CNNs that were implemented as testers besides another feature in order to use more codes to construct the (Chatbot), the authors used a technique called “Word Embedding”, Specifically, “Word2Vec”, (AlexNet), [LeNet5], {ResNet} & (VGGNet CNN) codes, as well as testing the codes for the exactitude purposes, 1st record, practicing & conduction, so all that mentioned proved that the implemented codes were unlike in their tasks. [12] and the paper explores the BRNN [Bidirectional Recurrent neural Networks] that companied the notification features in order to deal with the codes of the entering words and phrases (20–40+ words), enabling more contextually appropriate responses. The model is trained on an English-to-English dataset sourced from Reddit, focusing on enhancing perplexity, learning rate, and BLEU score for same-language translation [13]. So, this research developed a chatbot) for Android to diagnose diseases of kids, simulating the role of a pediatrician. Using data from [IMCI], (chatbot) employs a Backpropagation Neural Network (BPNN) to analyze parental input. The best-performing BPNN model included directing correctness by 94%, authentication by less than 1.05 & confirmation correctness by 64% [14]. Developing a college-oriented chatbot to assist students and staff with campus-related queries and in-person information searches. The chatbot, utilizing LSTM (Long Short-Term Memory), a variant of RNNs, to process user inputs and generate appropriate responses. It stores user inputs in a database for future reference, ensuring improved functionality over time. This innovative approach aims to simplify campus navigation and enhance access to college resources [15]. The developed and generative chatbot uses a deep LSTM Sequence-to-Sequence model with an attention mechanism, it’s designed for open-domain conversations, allowing it to engage in meaningful interactions with humans. It is trained on Reddit conversation datasets and evaluated using the Turing test to assess its conversational quality. The performance of the proposed chatbot is compared with the Clever bot, with results highlighting its effectiveness in delivering a more dynamic and versatile conversational experience [16]. Figure (1) Explain the comparison of the related works results:

J. of Soft Computing and Data Mining Vol. 6 No. 1 (2025) p. 95-106

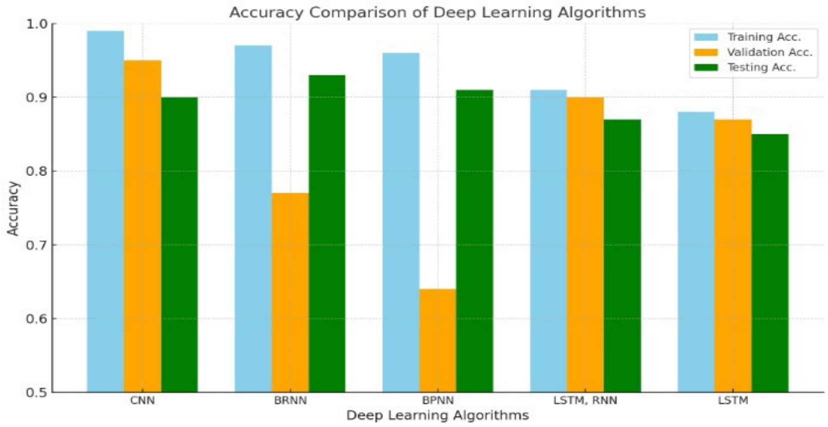

Fig.1 *The comparison between the results of the related work*

Table (1) shows the application domain of each related work:

Table 1 *Application domain of the related work*

Reference	Neural Network Technique	Application Domain
[12]	CNNs (AlexNet, LeNet5, ResNet, VGGNet)	General Chatbot Applications
[13]	Bidirectional RNN (BRNN)	General Conversational AI
[14]	Backpropagation Neural Network (BPNN)	Healthcare (Pediatric Diagnosis)
[15]	LSTM	Educational (Campus Assistance)
[16]	LSTM with Attention (Seq2Seq)	Open-Domain Conversational AI

3. Deep and Optimization Algorithms

LSTM the most developed if compared with (RNN) & consecutive system, is used to perform sequence predictions, the basic characteristic of LSTM that differs from traditional RNN is that it uses the gate mechanism to improve the problem of gradient vanishing [17], it has three gates that provide a boost for the type of performance, so the 1st one about the command if it was previously called or not, the 2nd one about if the command is new to implement new words and the last one is to link the new information of the specific date, as well as the mentioned Bi-LSTM were recalled via the gates, so the three gates as the following(Forget, Input & Output), so, the Bi-LSTM is controlled by these three components (forget, input and output), and they work together to manage the flow of information, enabling the network to "remember" or "forget" information over long sequences to store valuable information [18] below the equation of LSTM and the Figure (2) for LSTM gates [19] [20]

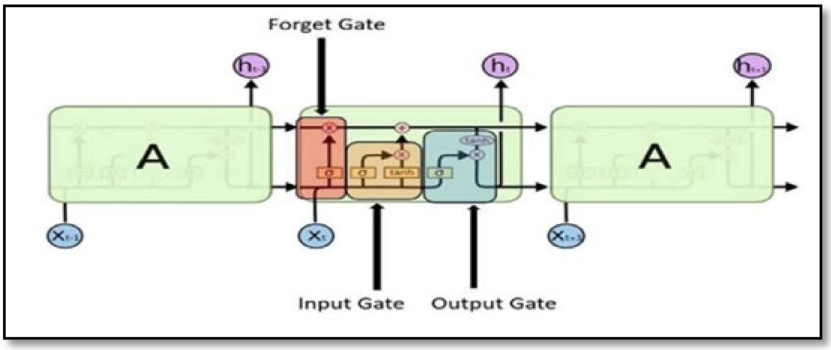

Fig. 2 *Gates of LSTM*

Equations of input gate: (process)

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{1}$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \quad (2)$$

Equation of Forget gate:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (3)$$

Equations of Output gate:

$$O_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (4)$$

$$h_t = o_t * \tanh(C_t) \quad (5)$$

Where:

- $\boldsymbol{x_t}$ represents the input at a time t.
- $\boldsymbol{h_t}$ is the output at time t.
- $\boldsymbol{C_t}$is the cell state at time t.
- $\boldsymbol{\sigma}$ is the sigmoid activation function.
- $\odot$ denotes element-wise multiplication.
- $\boldsymbol{W}$and $\boldsymbol{b}$ represent weights and biases respectively.

In Optimization process the Particle Swarm Optimization (PSO) is a stochastic population-based algorithm, all the mentioned results are to improve the issue that was solved as a (Bird), mostly known as element so the group of the elements called as flock so the transferring of the information via Dimension navigation for the error, so the location of each element will be moved according to the happening of the element besides the others nearby. Swarm techniques provide innovative and efficient ways to optimize and adjust weights in neural networks, enhancing their ability to accurately learn and adapt to complex problems like the image processing and the natural language processing, as well as the main benefits of (PSO) if we make a comparison with the improved plans, so the implementation of the (PSO) is not hard besides we have not that much of the correction classifiers, the following rules were reviews [21] [22]:

(Particle update rule): [*p* = *p* + *v]* (6)

as {$v = v + c_1 * rand * (pBest - p) + c_2 * rand * (gBest - p)$} (7)

The meaning of

- *[p*: position of particle]
- *[v*: direction of path]
- *[*c_1: local information's weight]
- *[*c_2: global information weight]
- *[pBest*: the particle of teh element's best position]
- *[gBest*: the swarm or folk's best position of]
- *[rand*: variable random]
- *[*C_1 personal best value's importance]
- *[*C_2 neighborhood best value's importance]

4. The Proposed Methodology

The proposed methodology aims on using deep neural network (Bi-LSTM) with one of the optimization algorithms (PSO) to optimize the process of the neural network that explored in chatbot using question and answering dataset after getting started with the dataset there are many Stages must be done to get ready in the classification process as explained below:

4.1 Dataset Preprocessing Stage

The dataset of research it is a dialogs dataset [23] which represent a real conversation and rich with quality. The data available in this dataset is in text format with (3725) items and are structured as questions and answers with punctuation and unnecessary characters so therefore must firstly carried out data preprocessing to filter the data. The data preprocessing requires eliminating the unlike data such as the (Abbreviations), the entire dataset's alphabets changed into lowercase besides deleting the entire symbols in order to prevent the variation, as well as enriching the superiority of the text via eliminating it like deleting the punctuation and switching an acronym [24] [25] as the Figure (3) that reviews the mentioned steps.

The steps of data preprocessing are:

- Input dataset.
- Assign each instance as code to an ID number.
- Build table for question and answers.
- Clean text data.
- Count the occurrence and mapping the table.
- Adding length and sort the question and answers according to it.

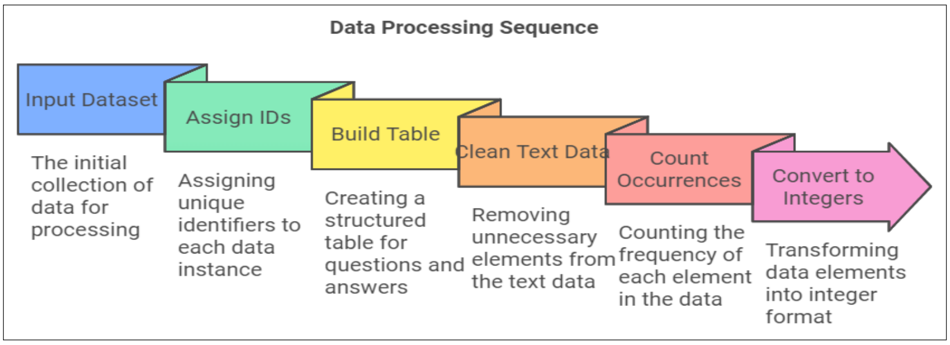

Fig. 3 *Steps of data preprocessing*

Figure (4) shows how the dataset before and after preprocessing.

Hello, how can I help you today?
Hi! I need help with my account.
What seems to be the issue with your account?
I forgot my password and can't log in!!!
No worries, I can help with that. Can you provide your email?
Sure, it's john_doe@example.com.

(a) Before Preprocesssing

ID	Length (words)	Question	Answer
1	13	hello how can i help you today	hi i need help with my account
2	17	what seems to be the issue with your account	i forgot my password and cant log in
3	18	no worries i can help with that can you provide your email	sure its johndoeexamplecom

(b) After Preprocessing

Fig. 4 *Example and dataset before and after preprocessing*

4.2 Feature Fusion Stage

Combining Word2Vec and TF-IDF features using a neural network layer can be an effective way to create a hybrid representation. Instead of directly concatenating the two feature sets, a neural network can learn an optimal combination by applying transformations and weighting the contributions of each feature type. This step is prepared by implementing the following steps:

4.2.1 Prepare Word2Vec and TF-IDF Features:

For each word in a sentence, obtain its corresponding Word2Vec embedding. Aggregate embedding's directly as sequences then compute the TF-IDF vector for the same sentence. After that to ensure compatibility between Word2Vec and TF-IDF, we implement padding for alignment.

4.2.2 Define the Neural Network Layer for Fusion

Input the features of Word2Vec embeddings and TF-IDF vectors as two separate inputs. Pass both inputs through separate dense (fully connected) layers to reduce their dimensions then combine the outputs: concatenation the two transformed feature vectors to train the model with the combined features as input and task-specific labels as output. Use an optimizer like Adam, and a suitable loss function (cross-entropy for classification), and the batch size (32), the suggested parameters as in table (2):

Table 2 *The evaluation parameters of neural network*

Parameters	Configuration
epochs	50
Learning rate	0.001
batch size	32
Optimizer	Adam Optimizer
Training Accuracy	97
Validation Accuracy	95

This step is important to rebuild anew dataset to be ready for classification process as we get the labeled features as an output for the next step.

4.3 Classification Process Stage

The proposed methodology utilizes a hybrid approach by integrating Particle Swarm Optimization (PSO) to fine-tune the hyperparameters and weights of the Bi-LSTM network. This optimization aims to enhance the classification performance of the model. Figure (5) illustrates the block diagram of the proposed hybrid model, which combines Particle Swarm Optimization (PSO) with the Bi-LSTM network.

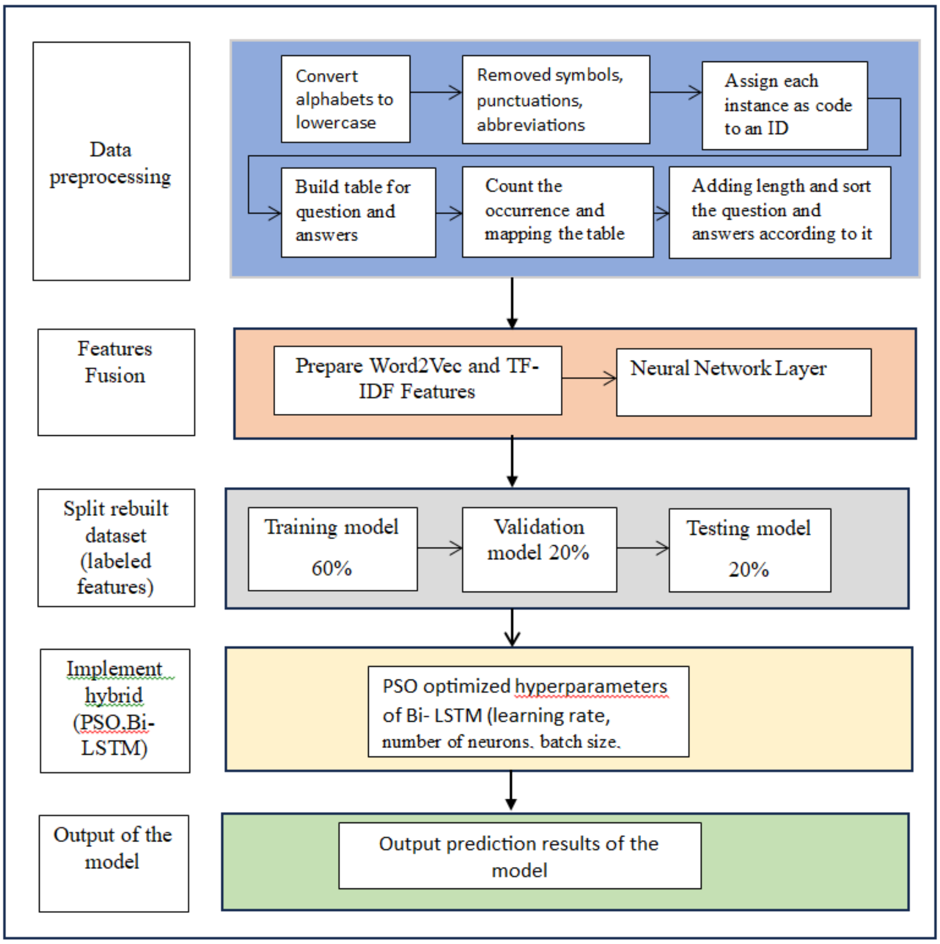

Fig. 5 *The block diagram of proposed hybrid (PSO,Bi-LSTM) model*

J. of Soft Computing and Data Mining Vol. 6 No. 1 (2025) p. 95-106

The proposed methodology hybrid the (PSO,Bi-LSTM) model using, it efficiently finds optimal solutions by iteratively improving candidate solutions. The hybrid (PSO,Bi-LSTM) combines the Bi-LSTM techniques with PSO to find the best hyper parameters (e.g., learning rate, number of neurons, batch size). This combination allows for faster convergence and higher accuracy compared to manually tuned Bi-LSTM models. The main steps of this hybrid model (PSO,Bi-LSTM) are:
For the proposed methodology we used (five layers) for Bi-LSTM with (three hidden layers) which will optimize their parameters (the gates) by using PSO as explained in the following algorithm 1.

Algorithm 1: Hybrid PSO, Bi-LSTM Model for intelligent Chatbot:

```
1.  Initialize Bi-LSTM Model:
    InputLayer(size)
    HiddenLayers (LSTM cells)
    OutputLayer(size)
2.  Define PSO Parameters:
    Set number_of_particles = N
    For i from 1 to N do:
        position[i] = Initialize randomly within feasible weight space of Bi-LSTM
        velocity[i] = Initialize randomly
        pBest[i] = position[i]
    Initialize gBest = position of best initial particle based on initial fitness
3.  Train PSO on LSTM Parameters:
    Repeat until convergence or maximum iterations reached:
      For i from 1 to N do:
          Set Bi-LSTM weights to position[i]
          Train Bi-LSTM on training data
          Evaluate Bi-LSTM on validation data to calculate fitness
          If fitness of position[i] > fitness of pBest[i] then:
              pBest[i] = position[i]
          If fitness of position[i] > fitness of gBest then:
              gBest = position[i]
4.  Update Particle Positions and Velocities:
     For i from 1 to N do:
          Update velocity[i] based on PSO velocity update formula
          Update position[i] based on updated velocity[i]
          Set Bi-LSTM weights to new position[i]
          Recalculate fitness for new position
          If new fitness > fitness of pBest[i] then:
              pBest[i] = new position[i]
          If new fitness > fitness of gBest then:
              gBest = new position[i]
5.  Final Model Evaluation:
    Set Bi-LSTM parameters to gBest
    Train Bi-LSTM on full training data with optimized parameters
    Evaluate Bi-LSTM accuracy on testing data
```

5. The Results

The result of the proposed methodology and the evaluation parameters used to enhance the accuracy for training the model to predict the answers of the intelligent chatbot. The set of parameters that used in the hybrid model shows in the Table (3) and we chose all the parameters that are used after many trial and error tries. The proposed model executes on python programming language and some libraries like tenserflow and pyswarm libraries, the number of Particles in PSO (30), Iterations (100), Bi-LSTM Configuration (hidden layers, 1024 LSTM units each).

Table 3 *The hyperparameter of the proposed methodology*

Parameters	Configuration
Batch size	(32)
Epoch	(100)
Number of Steps each epoch	(46)
Learning Rate	(0.001)
Units of Bi-LSTM	(1024)
Iteration	(100)
PSO	(30)

The dataset are splits into three parts 60% for training and 20% validation and 20% for testing. The results of Bi-LSTM with and without PSO optimization algorithm appeared in the Table (4):

Table 4 *The hyperparameter of the proposed methodology*

model	Training accuracy	Lose function	Validation accuracy	Lose function	Testing accuracy	Lose function
Bi-LSTM without PSO	0.8455	1.8737	0.8255	1.8737	0.8285	1.8667
Bi-LSTM with PSO	0.9961	0.0853	0.9854	0.0753	0.9882	0.053

Figure (6) shows the classification results accuracy after training the proposed model with and without using the PSO algorithm. There are other criteria used to measure the performance and the efficiency of the proposed model:

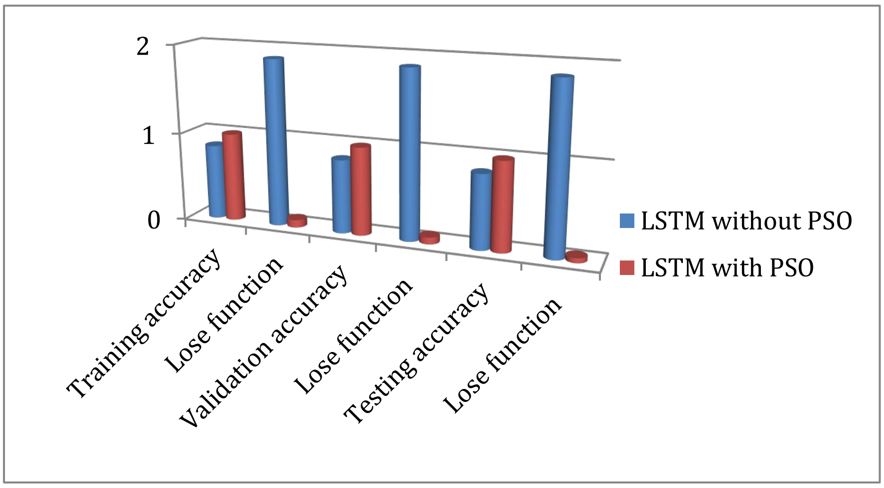

Fig. 6 *The results with and without using PSO*

BLEU (Bilingual Evaluation Understudy) score is used to evaluate the predicted which is considers the metric for evaluation the implemented of natural language processing to evaluate text that created automatically, so its measures how closely a model matches one or more reference texts. Figure (7) explain the Bleu score for the proposed model, and the Table (5) comparison of performance using Bleu score.

Table 5 *BLEU score*

The Proposed (Model)	Bleu (1) Gram	Bleu (2) Gram	Bleu (3) Gram	Bleu (4) Gram
LSTM without PSO	0.8432437	0.8375437	0.8274563	0.8136575
LSTM with PSO	0.8912653	0.8837152	0.8746456	0.8655674

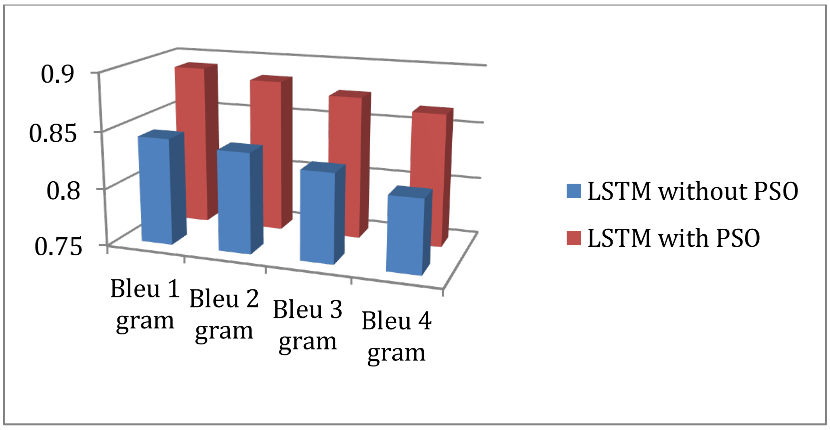

Fig. 7 *The results with and without using PSO*

And there are another metrics which are METEOR Score, Word Error Rate (WER), and finally the F1-Score. Figure (8) and table (6) shows the evaluation metrics with optimization.

Table 6 *The evaluation metrics with PSO*

Metric	Training Data	Metric	Training Data
METEOR Score	0.90	0.88	0.89
Word Error Rate (WER)	10%	12%	15%
F1-Score	0.95	0.93	0.94

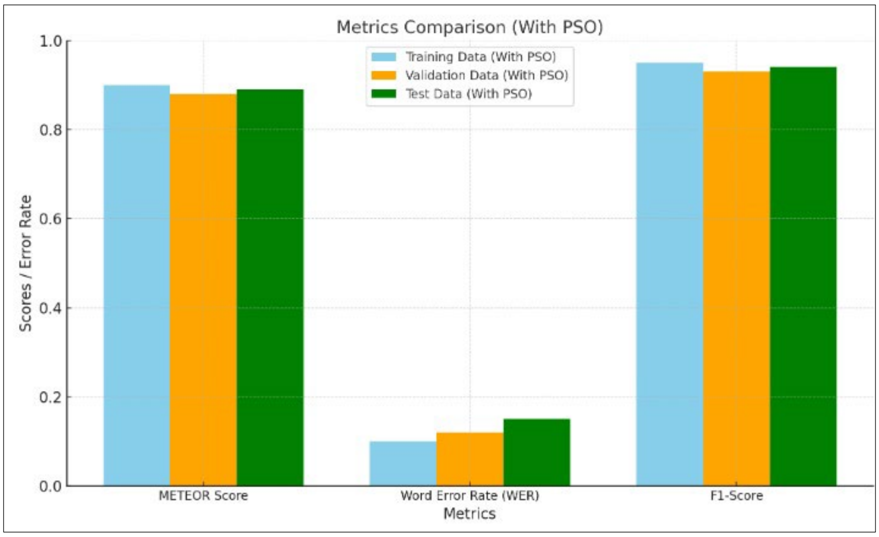

Fig. 8 *The evaluation metrics with optimization*

Figure (9) and table (7) below shows the evaluation metrics without optimization.

Table 7 *Evaluation metrics without optimization*

Metric	Training Data	Metric	Training Data
METEOR Score	0.70	0.68	0.67
Word Error Rate (WER)	18%	20%	22%
F1-Score	0.78	0.76	0.74

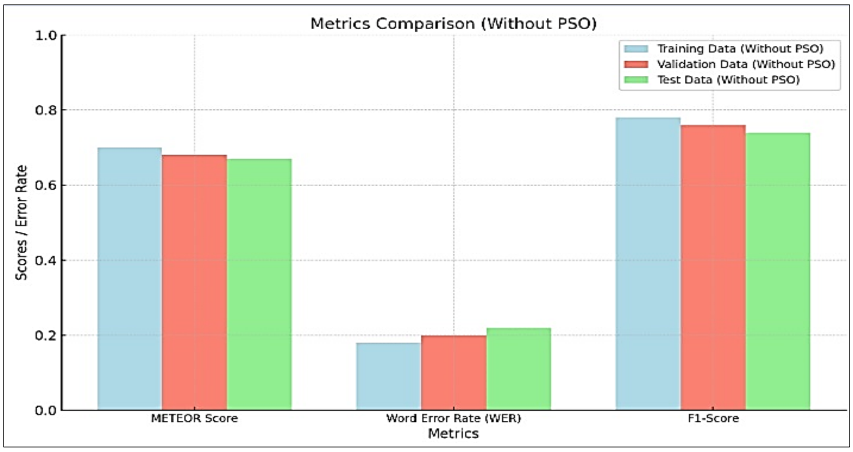

Fig. 9 *Evaluation metrics without optimization*

6. Conclusion

The contribution of this paper introduces a Particle Swarm Optimization to enhance the weight with Long Short-Term Memory (PSO-LSTM) model and demonstrates its success in offering a robust framework for hyper parameter optimization, reducing dependency on manual tuning and can yield higher accuracy by benefits of both sequential pattern recognition LSTM and the optimization PSO. In the proposed model (PSO-LSTM) help to achieve goals to have best replay with a 0.9961. The research evaluated & studied accuracy with planned PSO technique model and without PSO & reviewed the planned (PSO-LSTM) model with accuracy & execution.

Although the proposed hybrid PSO-LSTM model delivers substantial improvements in chatbot performance and accuracy, it encounters several constraints. A primary concern relates to scalability challenges. The model's intricate architecture, particularly the integration of particle swarm optimization with deep neural networks, may create obstacles when expanding the system to process extensive conversational datasets or support real-time applications that demand immediate response generation. Furthermore, computational overhead represents another significant constraint; incorporating PSO for parameter optimization considerably amplifies processing requirements, leading to extended training periods when compared to conventional, unoptimized neural network approaches. Overcoming these constraints would necessitate implementing complementary strategies, including distributed computing methods or advanced algorithmic refinements, to achieve an optimal balance between model sophistication, computational performance, and real-world applicability. The direction of future research includes looking into cutting-edge transformer architectures (like GPT) to improve conversational AI, testing chatbot performance in real-time, adding more cross-language features, using combined optimization methods, making models easier to understand and more open, and doing a full performance analysis across a wide range of application areas.

Acknowledgement

The authors have no financial support or academic advice for our research.

Conflict of Interest

The authors declare that there are no conflicts of interest regarding the publication of this manuscript.

Author Contribution

The authors confirm contribution to the paper as follows: ***Research problem proposal:*** *Huda W. Ahmed, Maysaa H. Abdulameer, Ruwaida M. Yas;* ***Theory development and computations:*** *Huda W. Ahmed, Maysaa H. Abdulameer;* ***Verification of analytical methods, investigation, and supervision:*** *Ruwaida M. Yas, Huda W. Ahmed, Maysaa H. Abdulameer; All authors discussed the results and contributed to the final manuscript. All authors reviewed the results and approved the final version of the manuscript.*

References

[1] Mihai, A. C., & Iancu, D. T. (2022, June). Optimizing a convolutional neural network using particle swarm optimization. In *2022 14th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)* (pp. 1-7). IEEE. https://doi.org/10.1109/ECAI54874.2022.9847314

[2] Dewantara, A. S. J. (2024). Implementation of a web-based chatbot using machine learning for question and answer services in universities. Advance Sustainable Science, Engineering and Technology, 6(1), 0240106.

[3] Setiabudi, D. H. B. G. S. A. R. (2024). Chatbot for complex questions in university admission using bidirectional long-short term memory and convolutional neural network. Proceedings of the 2024 2nd International Conference on Technology Innovation and Its Applications (ICTIIA).

[4] Kaminski, M. (2019). Neural network training using particle swarm optimization—A case study. Proceedings of the 2019 24th International Conference on Methods and Models in Automation and Robotics (MMAR).

[5] Rani, Y. A. B. A. S. M. R. N. S. A. R. P. K. A. (2024). AI-enhanced customer service chatbot. Proceedings of the 2024 International Conference on Science, Technology, Engineering, and Management (ICSTEM).

[6] Indu, M. S. N. S. C. (2023). Smart chatbot for college information enquiry using deep neural network. Proceedings of the 2023 9th International Conference on Advanced Computing and Communication Systems (ICACCS).

[7] Sageengrana, S. S. S. S. S. (2024). Optimized RB-RNN: Development of hybrid deep learning for analyzing students' behaviors in online learning using brain waves and chatbots. Expert Systems with Applications, 248, 123267. https://doi.org/[Insert DOI if available]

[8] Choudhary, P. C. S. (2023). An intelligent chatbot design and implementation model using long short-term memory with recurrent neural networks and attention mechanism. Decision Analytics Journal, 9, 100359.

[9] Cai, Z. (2024). Application of PSO algorithm in optimizing BP neural network. Proceedings of the 2024 IEEE 2nd International Conference on Control, Electronics, and Computer Technology (ICCECT), 853–859.

[10] Zheng, Y. J. (2024). A convolutional neural network model based on particle swarm optimization algorithm. Proceedings of the 2024 IEEE 6th Advanced Information Management, Communication, Electronic, and Automation Control Conference (IMCEC).

[11] Yas, R. M. H. S. (2021). Intelligent approaches for enhancing networked routing protocol. Iraqi Journal of Science, 2312-1637, 4121–4147.

[12] Tsakiris, G. P. C. P. G. K. K.-F. A. N. F. G. F. (2022). The development of a chatbot using convolutional neural networks. SHS Web of Conferences.

[13] Dhyani, M. K. R. (2021). An intelligent chatbot using deep learning with bidirectional RNN and attention mode. Materials Today: Proceedings, 34, 817–824.

[14] Muklason, A. R. F. R. E. M. F. T. R. V. R. A. B. N. (2022). Android-based chatbot application using backpropagation neural network to help the first treatment of children's diseases. Proceedings of the 7th International Conference on Sustainable Information Engineering and Technology.

[15] Chempavathy, B. P. S. N. V. D. L. Y. (2022). AI-based chatbots using deep neural networks in education. Proceedings of the 2022 Second International Conference on Artificial Intelligence and Smart Energy (ICAIS).

[16] Raj, V. P. M. (2022). A generative model-based chatbot using recurrent neural networks. Proceedings of the International Conference on Advanced Network Technologies and Intelligent Computing.

[17] Shu, X. Z. L. S. Y. T. J. (2020). Host–parasite: Graph LSTM-in-LSTM for group activity recognition. IEEE Transactions on Neural Networks and Learning Systems, 32(2), 663–674.

[18] Ruban, S. J. M. M. B. R. S. (2023). Improvisation of breast cancer detection using LSTM algorithm. Basti, Ram Shenoy.

[19] Muthamizarasan, M., & Ponnusamy, R. (2024). A comparative study of crime event forecasting using ARIMA versus LSTM model. Journal of Theoretical and Applied Information Technology, 102(5).

[20] Kadhim, N. K. A.-K. B. H. W. (2023). A proposed convolutional neural network for breast cancer diagnoses. Advances in Electrical and Electronic Engineering, 21(1), 9–18.

[21] Hong, J. T. W. (2023). Prediction in catalytic cracking process based on swarm intelligence algorithm optimization of LSTM. Processes, 11(5), 1454.

[22] Hameed, M. A. M. Z. A. K. A. K. M. M. M. (2024). A hybrid for analyzing text streaming using data mining and machine learning techniques. Journal of Engineering and Sustainable Development, 28(5), 675–680.

[23] Bio. (n.d.). Dialog.txt dataset for chatbot. [Available online]. Retrieved February 10, [https://www.kaggle.com/datasets/grafstor/simple-dialogs-for-chatbot].

[24] Assayed, S. K. A. M. S. K. (2023). Artificial intelligence-based chatbot for promoting equality in high school advising. Proceedings of the 2023 4th International Conference on Intelligent Engineering and Management (ICIEM).

[25] Ali, H. H. Y. R. M. M. H. (2023). Behavior analysis of machine learning algorithms for botnets detection. Proceedings of the 2023 International Conference on Information Technology, Applied Mathematics, and Statistics (ICITAMS).

[26] Shu, X., Zhang, L., Sun, Y., & Tang, J. (2020). Host–parasite: Graph LSTM-in-LSTM for group activity recognition. IEEE transactions on neural networks and learning systems, 32(2), 663-674.

[27] Ruban, S., Jabeer, M. M., & Basti, R. S. (2023, October). Improvisation of Breast Cancer Detection using LSTM Algorithm. In 6th International Conference on Intelligent Computing (ICIC-6 2023) (pp. 178-186). Atlantis Press.

[28] Hong, J., & Tian, W. (2023). Prediction in catalytic cracking process based on swarm intelligence algorithm optimization of LSTM. Processes, 11(5), 1454.

[29] Zheng, Y. J. (2024, May). A Convolutional Neural Network Model Based on Particle Swarm Optimization Algorithm. In 2024 IEEE 6th Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC) (Vol. 6, pp. 1096-1100). IEEE.

[30] Kaminski, M. (2019, August). Neural network training using particle swarm optimization-A case study. In 2019 24th International Conference on Methods and Models in Automation and Robotics (MMAR) (pp. 115-120). IEEE.

[31] Cai, Z. (2024, April). Application of PSO Algorithm in Optimizing BP Neural Network. In 2024 IEEE 2nd International Conference on Control, Electronics and Computer Technology (ICCECT) (pp. 853-859). IEEE.

[32] Dhyani, M., & Kumar, R. (2021). An intelligent Chatbot using deep learning with Bidirectional RNN and attention model. Materials today: proceedings, 34, 817-824.

[33] Choudhary, P., & Chauhan, S. (2023). An intelligent chatbot design and implementation model using long short-term memory with recurrent neural networks and attention mechanism. Decision Analytics Journal, 9, 100359.