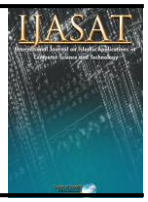

# An Intelligent Model for Analyzing the (I‘rāb) Syntax Parsing of the First Three Parts (Juz’) of the Holy Quran

**Teba A. Jasim [1], Prof. Dr. AbdulSattar Mohammed Khidhir [2], Safa Akram Altutunji [3]**

[1,] Technical Engineering College of Computer and Artificial Intelligence, Northern Technical University, Mosul, Iraq

[,2] Polytechnic College-Mosul, Northern Technical University, Northern Technical University, Mosul, Iraq

[,3] Engineering Technical College, Northern Technical University, Mosul, Iraq

[1] mti.lec74.teba@ntu.edu.iq, [2]abdulsattarmk@ntu.edu.iq, [3] safa.altutunji@ntu.edu.iq

**Abstract**

This research paper aims to design and develop an intelligent model based on deep neural networks and long-term memory (LTM) for the automated analysis of Qur'anic texts. This facilitates a more accurate understanding of the grammatical structures of the Qur'anic text. The first three parts of the Holy Qur'an were used as the data set, and a long-term memory network (LSTM) was employed after converting the Qur'anic words from their textual form into numerical representations that could be trained and processed by the proposed model. The results showed the model's ability to predict and analyze the grammatical structure of the words with 96% accuracy, demonstrating the effectiveness of long-term memory networks in analyzing words within the complex grammatical and morphological structure of the Arabic language.

**Keywords:** LSTM, Intelligent model, Quran analysis, Deep learning.

## 1. Introduction

Syntactic analysis (morphology) is the most important feature of the Arabic language(Hochreiter & Schmidhuber, 1997), enabling us to understand meanings accurately(دراوشة, 2023) and distinguish between different grammatical structures(Thaib & Shamsuddin, 2023). Therefore, Muslim scholars paid great attention to the syntactic analysis of the Holy Quran due to its impact on interpreting meaning and ensuring correct recitation(Maalej et al., 2021)(حمودي1, 2025). Despite numerous human efforts in syntactic analysis, the automated syntactic analysis of Arabic texts in general(مدني et al., 2025)( & سامي دعاء, 2024), and Quranic texts in particular, remains a significant challenge for researchers in the field of Natural Language Processing (NLP) due to the complexity of morphological and syntactic rules and the richness of the Arabic vocabulary(Al-Fadhli et al., 2023). With the great advancements in artificial intelligence and deep learning techniques, long-term memory networks have proven their efficiency in handling sequential data such as language and text(S. F. Ahmed et al., 2023)(Jiang et al., 2024)(Saeed et al., 2024). This research aims to design a deep learning model for a long-term memory network to classify the grammatical case of each word in the first three parts of the Holy Quran(Jiang et al., 2024). This is done

by applying a long-term memory network model that learns all grammatical patterns from previously translated texts, with the aim of predicting the correct grammatical case of Qur'anic words.

Numerous studies have explored the analysis of Arabic using artificial intelligence techniques(2025 ,البلادي & الرشيدي)(عبدالوهاب et al., 2023). Some research has employed traditional neural networks for grammatical structure analysis(Rashidi et al., 2024)( ,زينهم 2025), such as recurrent neural networks. These networks rely on their ability to process sequential data(Fadel et al., 2021), making them suitable for handling word sequences in Arabic texts(H. A. Ahmed & Mohammed, 2022). More recent studies have turned to more advanced models, such as short-term long memory (LSTM), a type of recurrent neural network (RNN), for grammatical structure analysis due to their ability to handle longer linguistic contexts(Maalej et al., 2021). In the field of Qur'anic analysis, most previous research has focused on discrimination, basic lexical analysis, (Al-Nima et al., 2023), root extraction, and occasionally the grammatical correctness of sentences. However, no studies have been specifically dedicated to the automated analysis of (I'rāb) the Qur'an using LSTM, and those that exist are very limited or virtually nonexistent. Given the emergence of this research gap, particularly in the field of automated Quranic analysis using LSTM memory networks, this research aims to bridge this gap by constructing a model for analyzing Quranic text based on LSTM memory networks. These networks are distinguished by their ability to process long-term sequential data, as they can retain information for extended periods within a sequence. Furthermore, they are capable of handling sequences of varying lengths without loss of context. These characteristics make them suitable for automated Quranic analysis, thus justifying this research.

## 2. Research Background

### 1.2 Long Short-Term Memory (LSTM)

It is a type of recurrent neural network (Mbow et al., 2021)designed to process sequential data such as text(Rehman et al., 2024)(Saeed et al., 2024), time series, and audio signal(Huang et al., 2022). LSTM is characterized by its ability to learn long-term relationships and overcome the vanishing/exploding gradient problem that plagues recurrent networks(Al-Dulaimi & Kurnaz, 2024)(Ghislieri et al., 2021). This algorithm relies on a cell structure containing three gates that control the flow of information over time(Aldabbagh, 2025)(YOUNIS & AL AZZO, 2024).

1.Forget Gate: which determines which part of the previous memory should be forgotten.

$$f_t = \sigma(W\_f \cdot [h_{t-1}, x_t] + b\_f) \quad (1)$$

2.Input Gate: This determines the new information that will be added to memory.

$$i_t = \sigma(W\_i \cdot [h_{t-1}, x_t] + b\_i) \quad (2)$$

$$\tilde{C}_t = \tanh(W\_C \cdot [h_{t-1}, x_t] + b\_C) \quad (3)$$

3.Output Gate: which controls the current output that will move to the next step.

$$o_t = \sigma(W\_o \cdot [h_{t-1}, x_t] + b\_o) \quad (4)$$

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

The memory state is updated according to the equation:

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad (6)$$

Thanks to this structure, LSTM memory can retain important information for extended periods and discard unnecessary data, making it suitable for classification and temporal data analysis tasks.

### 2.1 Data Preparation and Processing

The data was downloaded from the official website of the Qur'anic Corpus Project, which provides detailed analytical files of the Qur'anic text in a text format (quranic-corpus-morphology-0.4.txt). The text file contains standardized linguistic analysis lines, each line indicating the word's position in the text, its type, and its morphological characteristics.
For example, the following format represents a sample of the input data:

| (114:6:3:3) | n~aAsi | N | STEM \|POS:N \| LEM:n~aAs \| ROOT:nws \| MP \| GEN |
|---|---|---|---|

Each part of this line indicates the following:

- (114:6:3:3): The word's position in the text (chapter number, verse number, word number, sub-part)
- n~aAsi : The word is spelled in Latin letters according to the Buckwalter system and represents the Arabic word "النَّاسِ" (people).
- N: Word type: noun
- STEM, POS, LEM, ROOT, CASE, GENDER, NUMBER: Morphological properties such as root, case, gender, and number.

#### 2.1.1 Word Analysis in Arabic

Surah 114, Verse 6, Third Word, Part 3 → Word: "النَّاسِ" (al-nas), Type: Noun, Plural: النَّاس (al-nas), Root: نوس (nous), Masculine Plural, Genitive Case.
The data was processed using Python because it contains reliable libraries for data organization and text analysis. The pandas library was used to organize the data and convert it into Excel spreadsheets, facilitating review of results and subsequent statistical analysis. The os library organizes paths and manages files.

The processing stages included the following:

1. Read all lines of data in the text file and ignore blank lines.
2. Analyze the lines of the file and extract all information pertaining to each word, such as root, prefix, case, number, and gender.
3. Organize the properties (attributes) into a single dictionary containing the following standard columns: STEM, POS, LEM, ROOT, PREFIX, GENDER, NUMBER, CASE.
4. Compile the results into a Data Frame using the pandas library. Then, export them to an Excel file containing all the Quranic words with their morphological properties, organized into the following columns:

| Location | Form | Tag | Stem | Pos | Lem | Root | Prefix | Gender | Number | Case | Extra_Features |
|---|---|---|---|---|---|---|---|---|---|---|---|

Each column in the row above represents a linguistic or morphological property of the original data, as shown in Table 1, which illustrates the linguistic properties of the data from the Holy Quran.

Table 1: Patterns for NP extraction

| Column Name | Description |
|---|---|
| **LOCATION** | Indicates the word's position in the Quranic text (Surah: Verse: Word:Part). |

| | |
|---|---|
| **FORM** | The transliterated form of the Arabic word (Buck walter representation). |
| **TAG** | The general part-of-speech tag (noun, verb, particle, etc.). |
| **STEM** | The stem or core morphological base of the word. |
| **POS** | Part of speech; identifies the grammatical category of the word. |
| **LEM** | Lemma; the canonical dictionary form of the word. |
| **ROOT** | The triliteral or quadriliteral root of the word. |
| **PREFIX** | Any prefixes or additional letters attached to the word root. |
| **GENDER** | Indicates grammatical gender (masculine or feminine). |
| **NUMBER** | Indicates grammatical number (singular, dual, or plural). |
| **CASE** | Indicates grammatical case (nominative, accusative, genitive, etc.). |
| **EXTRA_FEATURES** | Contains any additional linguistic features not covered by the standard columns. |

- The text columns (FORM, ROOT, POS, PREFIX) were encoded independently
- using the Tokenizer tool from the Keras library, which converts words into numbers representing their dictionary keys.
- The length of the sequences was balanced using the pad_sequences function so that the length of each sequence remained constant (MAX_SEQ_LEN =1) because each row represented only one word.
- The labels were encoded by converting the text values in the target TAG column to numeric values using the Label Encoder tool from the scikit-learn library, where each label was assigned a unique number.
- These numbers were then converted to One-Hot vectors using the to_categorical() function to suit multi-category classification tasks.
- Finally, the data were divided into two groups: training (80%) and testing (20%)
- A multi-input deep neural network model was designed as follows:
- The model was designed to pass each input from the four columns representing
- linguistic features (FORM, ROOT, POS, PREFIX), each represented as
- a numerical sequence, through an Embedding layer independently to transform
- each word or symbol into
- a 64-dimensional dense vector representation. This transformation allows the model
- to understand the semantic relationships between vocabulary at the feature level.
- Using a Concatenate layer, we combined all the vector representations resulting from
- the four output Embedding layers to form a unified representation with dimensions
- of (1 × 256).
- This was then passed to a 256-unit LSTM layer to extract the relationships between features, i.e., to extract the temporal patterns and contextual structures between different linguistic features.
- •Next, we added a Dropout layer with a 0.5 index to prevent overfitting.
- Then, we added a Dense layer with 128 nodes and enabled ReLU to process the previous outputs non-linearly.
- After that, we added a second Dropout layer with a 0.4 index to achieve better model generalization.

- In the final layer, we added a final Dense layer with a number of nodes equal to the (45) classes in the designed model and used the Softmax function to generate multiple membership possibilities for each class for multiclass classification.
- This is shown in Figure (1), which illustrates the structure of the proposed multi-input LSTM network model used in the research.

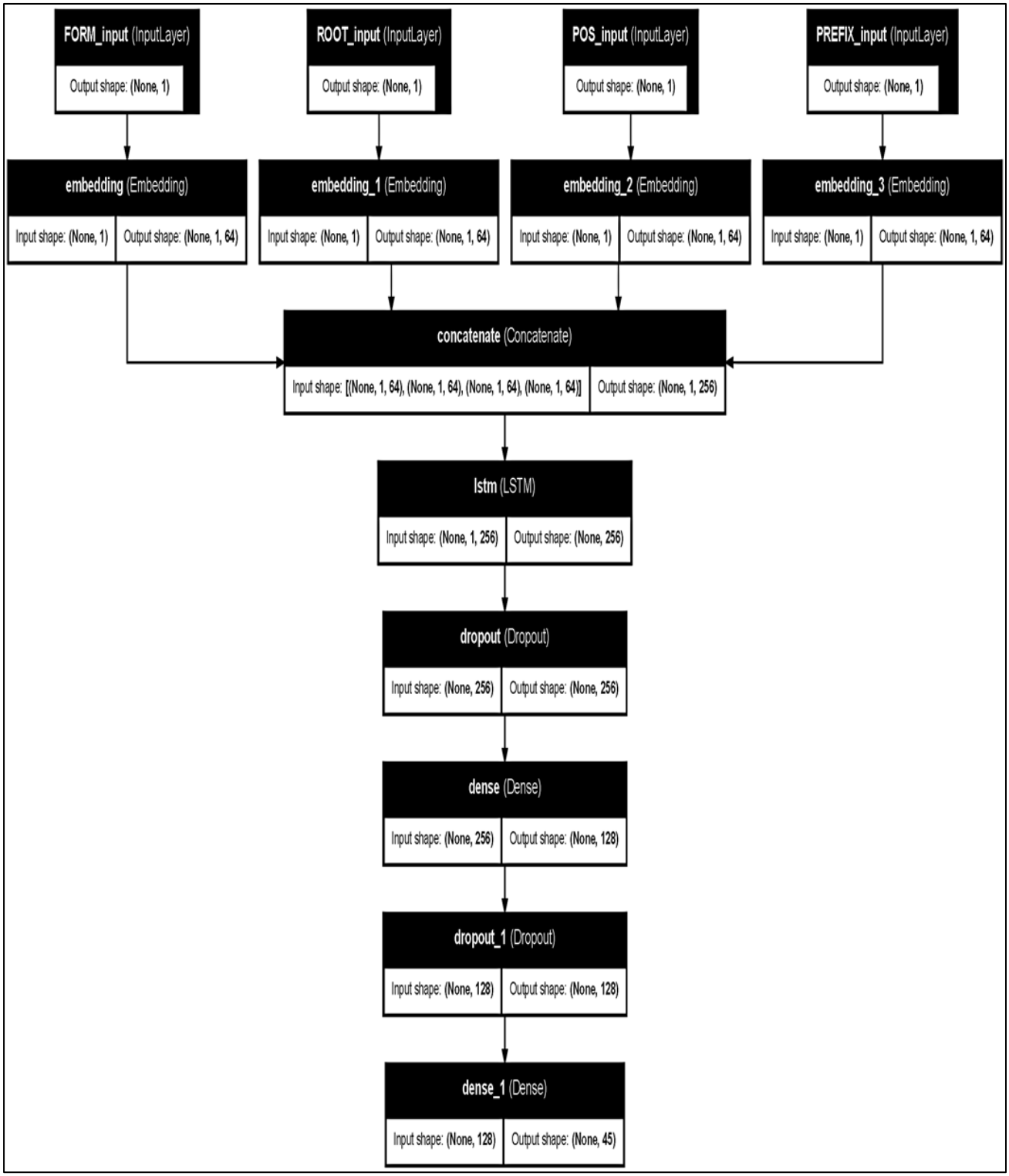


Figure 1: Structure of a multi-input (LSTM) network model.

This structure enables the model to learn the morphological and syntactic properties of the word, which enhances the accuracy of the linguistic tag classification (TAG classification). The test results showed high accuracy in performance with good generalizability.

### 2.1.2 Model Training Phase

• The model was initialized using the Adam algorithm as the optimizer, with a categorical crossentropy error function, as it is suitable for multi-category classification.

• The model was trained with((15) epochs and a batch size of (32), with (20%) of the training data allocated for validation. Model Evaluation Phase

After completing the training process, the model was evaluated on test data to calculate its accuracy and loss.

Accuracy was used to express the percentage of correctly classified samples.

### 2.1.3 Model Saving Phase

• The trained model was saved in h5 format for future use in prediction or reloading.

• All tokenizers and label encoders were saved using the Pickle library, which allows for the re-encoding process in the future.

### 2.1.4 Prediction process

A custom function called predict_tag_multi()` was ultimately built. This function predicts the syntactic and morphological tag of a word based on its properties: form (FORM), root (ROOT), word type (POS), and prefix (PREFIX).

The function encodes the new values in the same way as the training data and then uses the model to provide the most likely classification.

def predict_tag_multi(word, root='UNK', pos='UNK', prefix='UNK'): pred = model.predict(seqs)

Figure (2) illustrates the detailed steps of the entire model design and implementation process, from data entry to obtaining the final results.

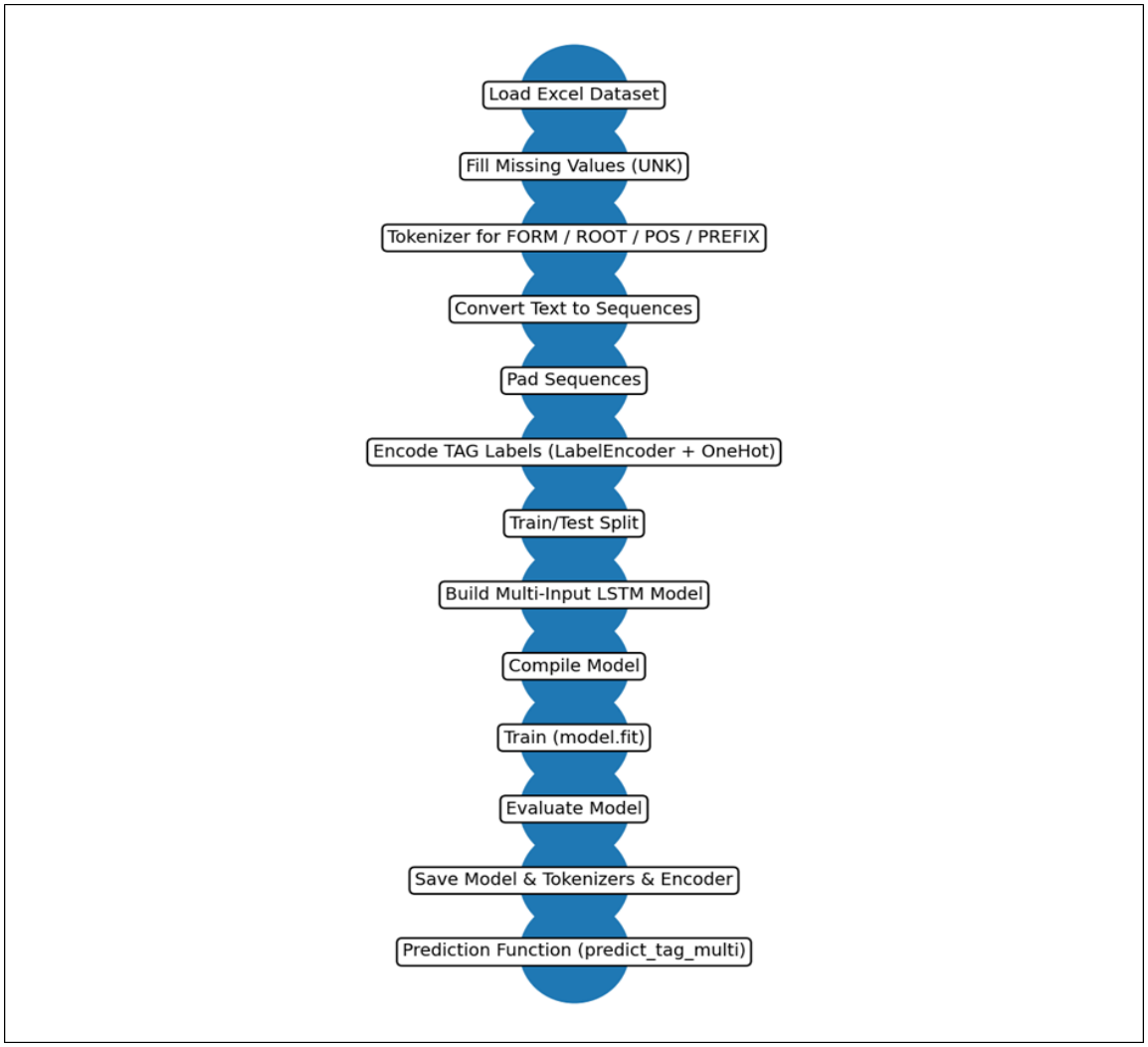


Figure 2: Flowchart for the design and implementation of the proposed model.

## 3. Results

The model was trained on a dataset of 128,219 samples of grammatical analysis from the first three parts of the Quran, distributed across 45 different grammatical categories. Despite the significant variation in the number of samples among the categories, as illustrated in Figure( 3) (Data Distribution Diagram), the model achieved excellent results on the test dataset. The poor performance, or even complete lack, of some categories is attributed to a clear imbalance in the distribution of syntactic features within the data. Some dominant features (such as N, PRO, and V) constituted a large proportion of the samples, while rare or underrepresented features were underrepresented. This resulted in poor model learning for the few or rare categories, which explains the observed lack of performance in some classification results.

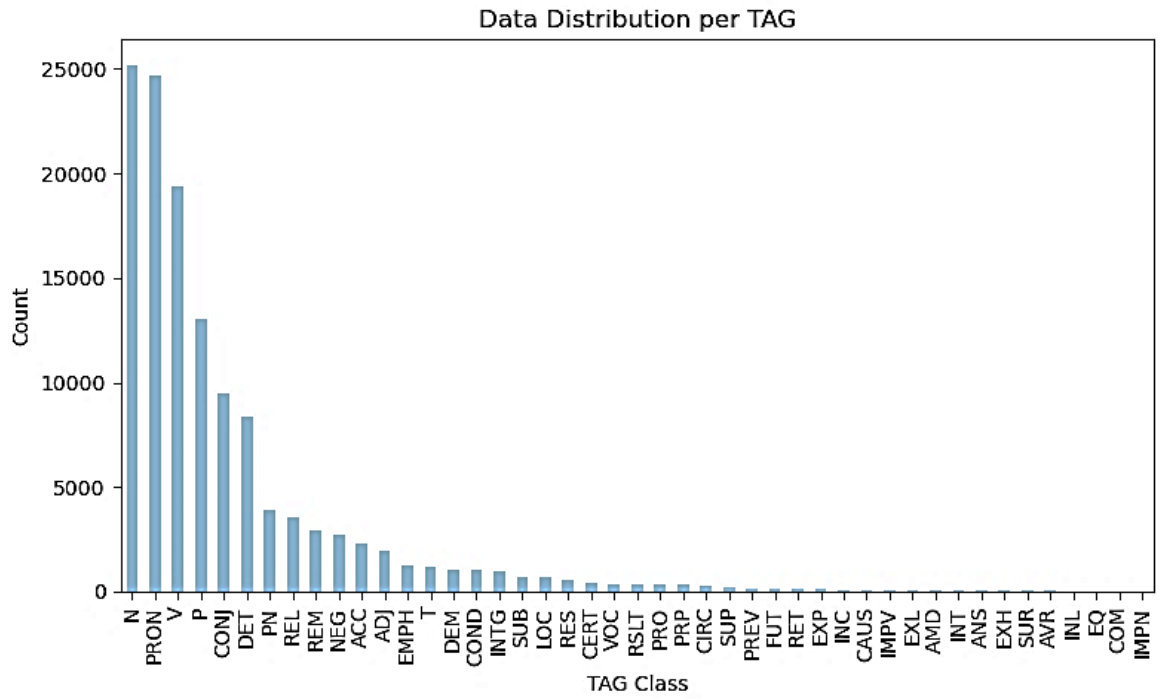


Figure 3: data distribution diagram

The model achieved an accuracy of (96.45%) on the test data, with a Precision value of (0.9584), a Recall value of 0.9645, and an F1-Score value of (0.9555). As shown in Table (2), which details the performance metrics for each category on the test data, these high and similar values indicate that the model has a good balance between accuracy and recall, reflecting efficiency in generalization and lack of bias towards one category at the expense of another.

Table 2: Performance scale for each category (Precision, Recall, F1-score)

| N | Class | Precision | Recall | F1-Score |
|---|---|---|---|---|
| **1** | ACC | 1.0000 | 1.0000 | 1.0000 |
| **2** | ADJ | 1.0000 | 1.0000 | 1.0000 |
| **3** | AMD | 1.0000 | 1.0000 | 1.0000 |
| **4** | ANS | 1.0000 | 0.6667 | 0.8000 |
| **5** | AVR | 1.0000 | 1.0000 | 1.0000 |
| **6** | CAUS | 0.0000 | 0.0000 | 0.0000 |
| **7** | CERT | 1.0000 | 1.0000 | 1.0000 |
| **8** | CIRC | 0.0000 | 0.0000 | 0.0000 |
| **9** | COM | 0.0000 | 0.0000 | 0.0000 |
| **10** | COND | 0.9875 | 1.0000 | 0.9937 |
| **11** | CONJ | 0.8625 | 0.9379 | 0.8986 |
| **12** | DEM | 1.0000 | 1.0000 | 1.0000 |
| **13** | DET | 0.9916 | 0.9912 | 0.9914 |
| **14** | EMPH | 1.0000 | 0.0027 | 0.0053 |
| **15** | EQ | 0.0000 | 0.0000 | 0.0000 |
| **16** | EXH | 1.0000 | 1.0000 | 1.0000 |
| **17** | EXL | 1.0000 | 1.0000 | 1.0000 |

| | | | | |
|---|---|---|---|---|
| **18** | EXP | 1.0000 | 1.0000 | 1.0000 |
| **19** | FUT | 1.0000 | 1.0000 | 1.0000 |
| **20** | IMPN | 1.0000 | 1.0000 | 1.0000 |
| **21** | IMPV | 0.0000 | 0.0000 | 0.0000 |
| **22** | INC | 1.0000 | 1.0000 | 1.0000 |
| **23** | INL | 1.0000 | 1.0000 | 1.0000 |
| **24** | INT | 1.0000 | 1.0000 | 1.0000 |
| **25** | INTG | 0.7136 | 1.0000 | 0.8328 |
| **26** | LOC | 1.0000 | 1.0000 | 1.0000 |
| **27** | N | 1.0000 | 1.0000 | 1.0000 |
| **28** | NEG | 1.0000 | 1.0000 | 1.0000 |
| **29** | P | 0.9101 | 0.9813 | 0.9444 |
| **30** | PN | 1.0000 | 1.0000 | 1.0000 |
| **31** | PREV | 1.0000 | 1.0000 | 1.0000 |
| **32** | PRO | 1.0000 | 1.0000 | 1.0000 |
| **33** | PRON | 0.9897 | 1.0000 | 0.9948 |
| **34** | PRP | 0.0000 | 0.0000 | 0.0000 |
| **35** | REL | 1.0000 | 1.0000 | 1.0000 |
| **36** | REM | 0.6166 | 0.6420 | 0.6291 |
| **37** | RES | 1.0000 | 1.0000 | 1.0000 |
| **38** | RET | 1.0000 | 1.0000 | 1.0000 |
| **39** | RSLT | 0.0000 | 0.0000 | 0.0000 |
| **40** | SUB | 1.0000 | 1.0000 | 1.0000 |
| **41** | SUP | 1.0000 | 0.0857 | 0.1579 |
| **42** | SUR | 1.0000 | 1.0000 | 1.0000 |
| **43** | T | 1.0000 | 1.0000 | 1.0000 |
| **44** | V | 1.0000 | 1.0000 | 1.0000 |
| **45** | VOC | 1.0000 | 0.9823 | 0.9911 |

The training and validation curves also showed that the model, as illustrated in Figure (4), which shows the model's accuracy and loss, reached a steady state after a limited number of epochs. At this point, there was no difference between the training and validation accuracy, indicating the absence of overfitting and the model's ability to generalize well despite the presence of unbalanced and rare data in some categories.

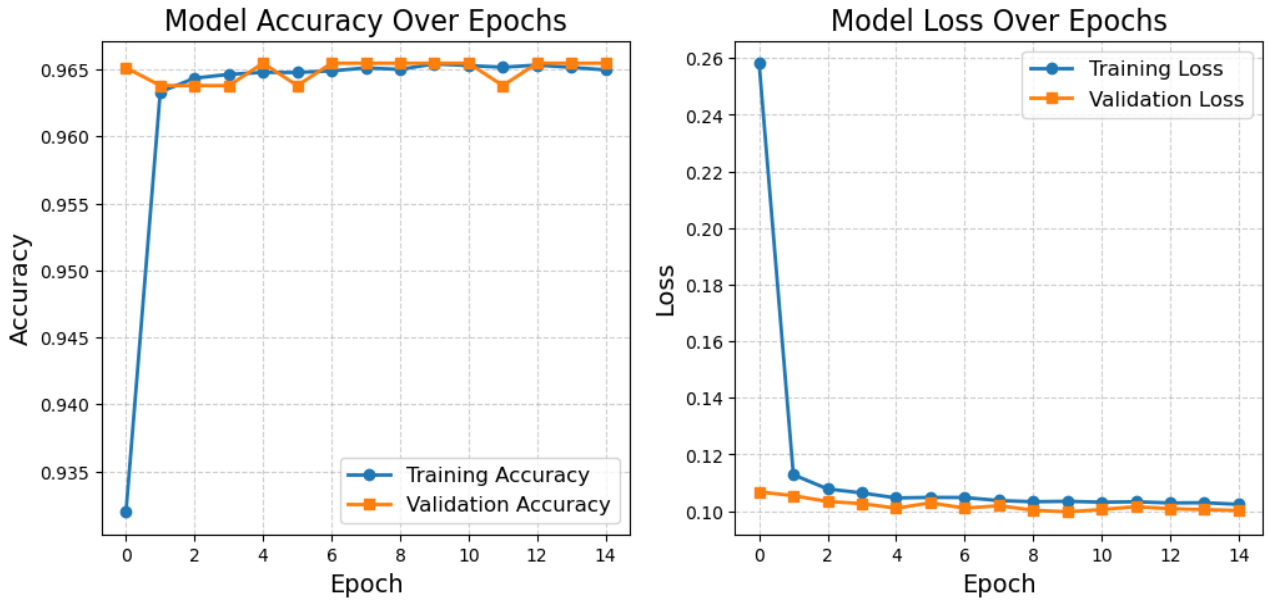


Figure 4: Model accuracy and loss

**The model's high performance was achieved for the following reasons:**

1. The LSTM network architecture was designed to handle sequential data in Arabic.
2. The preprocessing and cleanup of the data or text before model training.
3. Effective word embedding (Tokenization Word Embedding) was used for linguistic representation.
4. Precisely defined parameters suited to the model accelerated the learning process and improved stability.

Overall, the results demonstrate that the LSTM model is an effective solution for the automated parsing of the Holy Quran, highlighting the potential of employing these advanced capabilities, particularly deep learning techniques in general and the LSTM network in particular, in complex and sensitive Arabic language applications.

## 4. Conclusion

After data preparation and processing, designing the LSTM model, and training and testing the data, it demonstrated its high capability in analyzing the Quranic text (I'rāb) (the grammatical analysis of the first three parts(Juz')) with excellent accuracy. It was also able to learn complex Arabic word patterns within the Quranic text sequence. This research represents a significant step towards building a comprehensive automated system for parsing the Holy Quran using deep learning techniques. Future work includes expanding the study to encompass the complete parsing of (I'rāb) the Holy Quran using the Transformer algorithm to improve performance, along with designing an interactive interface that displays the parsing with grammatical interpretation.

## References

Ahmed, H. A., & Mohammed, E. A. (2022). Using Artificial Intelligence to classify osteoarthritis in the knee joint. *NTU Journal of Engineering and Technology*, *1*(3).

Ahmed, S. F., Alam, M. S. Bin, Hassan, M., Rozbu, M. R., Ishtiak, T., Rafa, N., Mofijur, M., Shawkat Ali, A. B. M., & Gandomi, A. H. (2023). Deep learning modelling techniques: current progress, applications, advantages, and challenges. *Artificial Intelligence Review*, *56*(11), 13521–13617.

Al-Dulaimi, O. A. H. H., & Kurnaz, S. (2024). Deep fake image detection based on deep learning using a hybrid CNN-LSTM with machine learning architectures as classifier. *2024 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*, 1–7.

Al-Fadhli, S., Al-Harbi, H., & Cherif, A. (2023). Speech Recognition Models for Holy Quran Recitation Based on Modern Approaches and Tajweed Rules: A Comprehensive Overview. *International Journal of Advanced Computer Science & Applications*, *14*(12).

Al-Nima, R. R. O., Al-Hatab, M. M. M., & Qasim, M. A. (2023). An artificial intelligence approach for verifying persons by employing the deoxyribonucleic acid (DNA) nucleotides. *Journal of Electrical and Computer Engineering*, *2023*(1), 6678837.

Aldabbagh, H. (2025). A Speech Act Analysis of Trumps's Second Inaugural Speech. *رؤى في الآداب والعلوم الإنسانية*, *4*(2), 85–97.

Fadel, A., Tuffaha, I., & Al-Ayyoub, M. (2021). Neural Arabic text diacritization: State-of-the-art results and a novel approach for Arabic NLP downstream tasks. *Transactions on Asian and Low-Resource Language Information Processing*, *21*(1), 1–25.

Ghislieri, M., Cerone, G. L., Knaflitz, M., & Agostini, V. (2021). Long short-term memory (LSTM) recurrent neural network for muscle activity detection. *Journal of NeuroEngineering and Rehabilitation*, *18*, 1–15.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735–1780.

Huang, R., Wei, C., Wang, B., Yang, J., Xu, X., Wu, S., & Huang, S. (2022). Well performance prediction based on Long Short-Term Memory (LSTM) neural network. *Journal of Petroleum Science and Engineering*, *208*, 109686.

Jiang, X., Li, F., Zhao, H., Qiu, J., Wang, J., Shao, J., Xu, S., Zhang, S., Chen, W., & Tang, X. (2024). Long term memory: The foundation of ai self-evolution. *ArXiv Preprint ArXiv:2410.15665*.

Maalej, R., Khoufi, N., & Aloulou, C. (2021). Parsing Arabic using deep learning technology. *TACC*, 74–80.

Mbow, M., Koide, H., & Sakurai, K. (2021). An Intrusion Detection System for Imbalanced Dataset Based on Deep Learning. In *Proceedings - 2021 9th International Symposium on Computing and Networking, CANDAR 2021* (pp. 38–47). https://doi.org/10.1109/CANDAR53791.2021.00013

Rashidi, A. N. A., Seman, H. M., & Hussin, M. (2024). Students' Needs towards Learning Arabic Language Via the Electronic Network: Saraf Unit as a Model: Language Learning. *Al-Ḍād Journal*, *8*(2), 1–22.

Rehman, A., Noor, F., Janjua, J. I., Ihsan, A., Saeed, A. Q., & Abbas, T. (2024). Classification of Lung Diseases Using Machine Learning Technique. *2024 International Conference on Decision Aid Sciences and Applications (DASA)*, 1–7.

Saeed, A. Q., Aldulaimi, M. H., Ismail, I. A., Ahmed, I. M., Yahya, Y. A., Kharma, Q. M., & Ghazal, T. M. (2024). Integrating Three Machine Learning Algorithms in Ensemble Learning Model for Improving Content-based Spam Email Recognition. *Journal of Soft Computing and Data Mining*, *5*(2), 188–196.

Thaib, Z. H., & Shamsuddin, S. M. (2023). Arabic Language Rules in Forming a Scholarly Personality of Exegete: Language Acquisition. *Al-Ḍād Journal*, *7*(2), 1–16.

YOUNIS, Y. B., & AL AZZO, F. (2024). INTEGRATIVE DETECTION OF RETINAL DETACHMENT WITH ALEXNET AND HEAT MAPS VISUALIZATION. *Mathematics for Application*, *13*(1).

الرشيدي, ب. ع., & البلادي, س. س. (2025). تقييم فعالية أداة الذكاء الاصطناعي ChatGPT-4 في تصميم منهجية علمية للأبحاث العلمية: دراسة تجريبية. *Journal of Information Studies and Technology*, *2025*(1), 5.

حمودي1, آ. (2025). دور دلالة الأصل اللغوية في فهم القرآن الكريم وتفسيره-دراسة تطبيقية دلالية. *مجلة العلوم الإنسانية والطبيعية*, *6*(2), 182–205.

دراوشة, ص. ا. أ. (2023). *تعليمية اللغة العربية: دراسات لسانية تطبيقية بحوث المؤتمر الدولي الرابع للسانيات التطبيقية وتعليم اللغات 3-5 مارس، 2022*. Zayed University Press.

زينهم, ه. (2025). رؤى لسانيّة تطبيقيّة في الذكاء الاصطناعيّ واللغة العربيّة. *Applied Linguistics Journal*, *2*(1), 134–139.

سامي, & دعاء. (2024). صيغة الوجوب في النصوص التشريعية باللغتين العربية والإسبانية: أداة حاسوبية للكشف الآلي والتحليل التقابلي. *مجلة كلية الآداب-جامعة القاهرة*, *84*(4), 1–36.

عبدالوهاب, ع., أحمد, محمود, م., عبدالرازق, رشوان, م. ع., & أحمد. (2023). تطبيقات الذكاء الاصطناعي وأثرها في تنمية الذات اللغوية الإبداعية لدى الطلاب الفائقين بالمرحلة الثانوية. *مجلة كلية التربية (أسيوط)*, *39*(1), 109–135.

مدني, خديجة, زنبوط, & إيمان. (2025). الذكاء الاصطناعي في التحليل الآلي للمشاعر واكتشاف العواطف في النصوص العربية. *مجلة الآداب و العلوم الإجتماعية*, *22*(1), 58–73.

**Biodata**

| | |
|---|---|
|  | Teba Ali Jasim: Assistant Lecturer at NTU, teaching courses in software engineering, artificial intelligence, and network security in the Department of Cloud Computing and Internet of Things.<br>Education: Master's degree in Software Science – Specialization in Intelligent Technologies and Network Security.<br>Email: Mti.lec74.teba@ntu.edu.iq<br>https://www.researchgate.net/profile/Teba-A-Jasim<br>https://orcid.org/0009-0009-9747-5865<br>https://scholar.google.com/citations?user=BbVWr9wAAAAJ&hl=ar<br>https://www.scopus.com/authid/detail.uri?authorId=58121472000 |
|  | AbdulSattar M. Khidhir (Born 1959) is a professor at Networks and Computer Software Techniques Department - Mosul Polytechnic College - Northern Technical University in Iraq. He obtained his B.Sc. (1981) and M.Sc. (1989) both in Electronics and Communications Engineering from University of Mosul. His Ph.D. (2000) was obtained in Communications Engineering from University of Mosul too. He supervised many Ph.D. and M.Sc. theses in different scientific and engineering areas. He was a member of scientific committees for many Ph.D. and M.Sc. students. He published many researches in various fields of science and engineering (see google scholar). He reviewed many scientific papers for journals and conferences.<br>https://scholar.google.com/citations?user=wPXOA9cAAAAJ&hl=en<br>https://www.scopus.com/authid/detail.uri?authorId=57225144859<br>https://orcid.org/0000-0001-6710-0987<br>https://publons.com/researcher/1732991/abdulsattar-khidhir/<br>Email: abdulsattarmk@ntu.edu.iq, abdulsattarmk@gmail.com |
|  | Name: Safa Akram younis<br>Academic Title: Assistant Lecturer<br>Affiliation: Technical engineering college / mosul<br>Email: safa.altutunji@ntu.edu.iq<br>Education: Master's degree in computer engineering<br>Specialization in Intelligent Technologies<br>Experience: Assistant Lecturer at NTU, Teaching courses in computer and artificial intelligence. |