Journal of Technology Research (JTR)
Volume 3, Special Issue, (2025), pp 494-504 ISSN 3005-639X
The 7th Conference of Engineering Sciences and Technology
(CEST-2025)

A Comparative Analysis of CNN and RNN Architectures for Deep Learning-Based Arabic Text Classification

Abdulmawla Najih[1], Ramzi Alshagif[2*], Albahlool Abood[3], Salem Enajeh[4]

[1] Higher Institute of Sciences and Technology, Computer Department, Gharian, Libya.

[2]Department of Computer Science, School of Basic Sciences, Libyan Academy, Tripoli, Libya.

[3]Faculty of Information Technology, Gharyan University, Libya.

[4]Higher Institute of Sciences and Technology, Computer Department Tripoli, Libya.

*Corresponding author email: nabdulmawla@gmail.com

Received: 19-09-2025 | Accepted: 02-12-2025 | Available online: 25-12-2025 | DOI:10.26629/jtr.2025.46

ABSTRACT

The proliferation of digital Arabic content has created a pressing need for efficient text classification systems. However, the Arabic language's complex morphological structure, including its root-based derivation and agglutinative nature, poses significant challenges for automated processing. While deep learning models like Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) have shown promise, their comparative effectiveness for Arabic text remains inadequately explored. This study presents a comprehensive empirical comparison of CNN and RNN models for multi-class Arabic text classification. We curated a heterogeneous dataset spanning seven distinct domains—including sports, politics, and economics—to ensure model robustness. A rigorous Arabic-specific preprocessing pipeline was implemented, involving stemming, stop-word removal, and tokenization. The CNN model utilized GloVe word embeddings for feature representation, whereas the RNN model employed TF-IDF vectors. Our results demonstrate a significant performance disparity: the RNN model achieved a remarkable 98% accuracy, substantially outperforming the CNN model, which reached 79% accuracy. Analysis of learning curves revealed that the CNN model suffered from overfitting, failing to generalize beyond the training data. In contrast, the RNN model effectively captured sequential dependencies and contextual information, which are crucial for understanding Arabic syntax and morphology. The findings strongly indicate that RNN architectures are superior for Arabic text classification tasks due to their innate ability to model long-range semantic relationships. This research provides valuable insights for selecting and developing optimal deep-learning architectures for Arabic NLP applications.

Keywords: Arabic Natural Language processing, Text Classification, Recurrent Neural Networks (RNNs), Convolutional Neural Networks (CNNs), Comparative Analysis.

تحليل مقارن لهندستي الشبكات العصبية التلافيفية (CNN) والمتكررة (RNN) لتصنيف النصوص العربية القائم على التعلم العميق

عبد المولى الناجح[1]، رمزي الشقف[2]، البهلول عبود[3]، سالم الناجح

[1]قسم الحاسوب، المعهد العالي للعلوم والتكنولوجيا، غريان، ليبيا.

[2]قسم علوم الحاسوب، كلية العلوم الأساسية، الأكاديمية الليبية، طرابلس، ليبيا.

[3]كلية تكنولوجيا المعلومات، جامعة غريان، ليبيا.

[4]المعهد العالي للعلوم والتكنولوجيا، قسم الحاسوب، طرابلس، ليبيا.

ملخـــــص البحـــــث

إن الانتشـار المتزايد للمحتوى الرقمي العربي قد خلق حاجة ملحة لأنظمة فعالة لتصـنيف النصـوص. ومع ذلك، فإن البنية الصـرفية المعقدة للغة العربية، بما في ذلك الاشـتقاق القائم على الجذور وطابعها اللاصـقي، تشـكل تحديات كبيرة للمعالجة الآلية. على الرغم من أن نماذج التعلم العميق مثل الشـبكات العصـبية التلافيفية (CNNs) والشـبكات العصـبية المتكررة (RNNs)أظهرت نتائج واعدة، إلا أن فعاليتها النسـبية في معالجة النصـوص العربية لا تزال غير مدروسـة بشـكل كاف. تقدم هذه الدراسـة مقارنة تجريبية شـاملة بين نموذجي CNN و RNN لتصـنيف النصـوص العربية متعددة الفئات. قمنا بتجميع مجموعة بيانات غير متجانسة تشمل سبعة مجالات متميزة—بما في ذلك الرياضـة والسياسـة والاقتصـاد—لضـمان متانة النماذج. تم تنفيذ خطوة معالجة مسـبقة صـارمة مخصـصـة للغة العربية، تشـمل التصـريف، وإزالة الكلمات غير الضـرورية، والتقطيع. اسـتخدم نموذج CNN تضـمينات الكلمات GloVe لتمثيل الميزات، بينما اسـتخدم نموذج RNN متجهات .TF–IDF أظهرت نتائجنا تفاوتًا كبيرًا في الأداء: حقق نموذج RNN دقة ملحوظة بلغت 98٪، متفوقًا بشـكل كبير على نموذج CNN الذي حقق دقة 79٪. كشـف تحليل منحنيات التعلم أن نموذج CNN عانى من الإفراط في التكيف، مما أدى إلى فشـله في التعميم beyond بيانات التدريب. في المقابل، اسـتطاع نموذج RNN التقاط التبعيات التسلسلية والمعلومات السياقية بكفاءة، وهي عناصر حاسمة لفهم تركيب اللغة العربية وصرفها. تشير النتائج بقوة إلى أن هياكل RNN تتفوق في مهام تصـنيف النصـوص العربية بسـبب قدرتها الفطرية على نمذجة العلاقات الدلالية طويلة المدى .يقدم هذا البحث رؤى قيمة لاختيار وتطوير هياكل التعلم العميق المثلى لتطبيقات معالجة اللغة الطبيعية العربية.

الكلمات الدالة: الشبكات العصبية التلافيفية (CNN) والمتكررة (RNN), تحليل مقارن , تصنيف النصوص العربية.

1. INTRODUCTION

The exponential growth of digital text data, driven by the internet and social media, has made automated text classification (TC) a cornerstone of modern information systems [1]. As a fundamental task in Natural Language Processing (NLP), TC enables a wide array of applications, from sentiment analysis and spam detection to content recommendation and topic labeling [2]. The effectiveness of these applications is deeply intertwined with the language of the text, presenting unique challenges that extend beyond the capabilities of traditional rule-based systems.

The Arabic language, spoken by over 422 million people and the liturgical language of 1.8 billion Muslims, is a language of immense global significance [3]. Despite its widespread use, Arabic NLP lags behind its English counterpart, primarily due to the language's intricate and rich morphological structure [4]. Arabic is a morphologically complex, root-based Semitic language characterized by phenomena such as agglutination, where prefixes, suffixes, and pronouns attach to a root word, and the frequent omission of diacritical marks (vowels) in everyday writing [5]. This omission introduces significant ambiguity, as a single written word can represent multiple meanings and pronunciations [6]. These

intrinsic features render conventional bag-of-words models and traditional machine learning algorithms like Support Vector Machines (SVM) and Naïve Bayes (NB) less effective, as they often fail to capture the nuanced semantic and syntactic relationships [7].

The advent of deep learning (DL) has revolutionized the field of NLP, offering powerful models capable of learning hierarchical feature representations directly from raw text data [8]. Among these, Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) have emerged as two of the most prominent architectures. CNNs excel at extracting local spatial features through their convolutional and pooling layers, effectively identifying informative n-grams and patterns within a text [9]. In contrast, RNNs and their advanced variants like Long Short-Term Memory (LSTM) networks are inherently designed to process sequential data. Their internal memory state allows them to capture long-range dependencies and contextual information across a sentence, making them theoretically well-suited for modeling language [10].

Several studies have explored the application of these models to Arabic text classification. For instance, the authors in. [11] demonstrated the effectiveness of CNNs, while others have utilized RNNs for sequential Arabic language modeling [12]. However, the existing body of research often suffers from two key limitations: (1) studies tend to focus on evaluating a single model architecture in isolation, and (2) many are conducted on limited or homogenous datasets, failing to test generalization across diverse domains. A direct, rigorous, and empirical comparison of CNN and RNN performance on a common, multi-domain Arabic benchmark, utilizing modern feature representation techniques, remains an underexplored area. This gap makes it difficult for researchers and practitioners to make informed decisions about the most suitable architecture for Arabic NLP tasks.

To address this gap, this paper presents a comprehensive comparative analysis of CNN and RNN models for multi-class Arabic text classification. The primary objective is to determine which architecture is more effective at handling the linguistic complexities of Arabic and generalizing across various topics. Our main contributions are fourfold:

1. We curate and preprocess a multi-domain Arabic text corpus from seven distinct categories to serve as a robust benchmark for evaluation.
2. We implement a detailed, reproducible Arabic-specific preprocessing pipeline including stemming, stop-word removal, and tokenization.
3. We develop and train two deep learning models: a CNN using GloVe word embeddings and an RNN using TF-IDF feature representation, each tailored to leverage the strengths of its respective architecture.
4. We provide an in-depth empirical analysis and discussion of the results, explaining the performance disparity through the lens of Arabic linguistics and model architecture, concluding that RNNs are superior for this task due to their ability to model sequential context.

The remainder of this paper is organized as follows. Section 2 reviews related work on Arabic NLP and deep learning for text classification. Section 3 details the methodology, including the dataset, preprocessing, and model architectures. Section 4 presents the results and provides a critical discussion. Finally, Section 5 concludes the paper and suggests directions for future research.

2. Related work

The task of automatically categorizing text has evolved significantly, from early rule-based systems to statistical machine learning models and, more recently, to deep learning approaches. Research on Arabic Text Classification (ATC) has generally followed this trajectory, albeit with a necessary focus on overcoming the language's unique challenges. This section reviews relevant literature in two main themes: (1) traditional machine learning methods for ATC and (2) deep learning-based approaches, further divided into studies using CNNs, RNNs, and comparative analyses.

2.1 Traditional Machine Learning for Arabic Text Classification

Early and ongoing work in ATC has heavily relied on traditional machine learning algorithms, often coupled with feature engineering tailored to Arabic's morphology. Algorithms such as Support Vector Machines (SVM), Naïve Bayes (NB), and k-Nearest Neighbors (k-NN) have been widely applied. A core focus of this research stream has been the critical role of preprocessing, particularly stemming and root extraction, to reduce feature space dimensionality and improve model performance [13]. For instance, [Citation for a paper like Galal et al.] introduced a stemming algorithm to enhance feature selection before classification.

The authors in [14] conducted a performance analysis of several ML algorithms, including C4.5, SVM, and Naïve Bayes, on multiple Arabic datasets. The study concluded that SVM generally achieved superior accuracy, a finding consistent with many text classification tasks in other languages due to SVM's effectiveness in high-dimensional spaces. Similarly, a comparative study by [15] evaluated SVM, k-NN, Logistic Regression, and Multinomial Naïve Bayes on two different Arabic datasets, further validating the consistent robustness of SVM models for this task. While these traditional methods yield strong baseline results, their performance is often contingent on meticulous and often complex feature engineering, and they may struggle to capture the deep semantic relationships within text.

2.2 Deep Learning for Arabic Text Classification

Deep learning models have gained prominence for their ability to automatically learn relevant features from raw or minimally processed text, thus reducing the reliance on manual feature engineering.

Convolutional Neural Networks (CNNs): CNNs have been successfully adapted for ATC, treating text as a one-dimensional spatial signal. Their ability to identify informative local patterns (e.g., key phrases or n-grams) makes them suitable for classification tasks. The authors in [16] demonstrated the effective use of CNNs for ATC, noting that performance was significantly improved by applying linguistic preprocessing techniques like normalization and stemming. Their work showed that CNNs could achieve high accuracy by leveraging these extracted features.

Recurrent Neural Networks (RNNs): RNNs, particularly Long Short-Term Memory (LSTM) networks, are designed to model sequential data and context, making them a natural fit for language tasks. Research has shown their applicability in generating and modeling correct Arabic sequences. The authors in [17] explored the adaptation of RNN architectures for Arabic language modeling, specifically for tasks like text generation and predicting missing text. Their work underscored the potential of RNNs to capture the temporal dependencies inherent in Arabic syntax and morphology, a challenge that other models often find difficult.

Comparative and Hybrid Studies: A limited number of studies have begun to directly

compare or combine architectures. For example, the authors in [18] provided a comprehensive analysis of DL and ML techniques for Arabic text categorization over a five-year period, highlighting the emerging dominance of deep learning but noting a scarcity of rigorous comparative studies. Furthermore, the author in [19] reviewed deep learning-based text classification algorithms, emphasizing the importance of feature extraction and reduction steps, but the work was not specific to Arabic. While some research exists, as noted by the authors in [20] in their survey, there remains a noticeable gap in the literature: a lack of controlled, empirical studies that directly benchmark CNN and RNN performance on a common, multi-domain Arabic dataset using modern feature representation methods like word embeddings. Most studies evaluate models in isolation or on limited-domain data, making it difficult to draw generalizable conclusions about their relative strengths and weaknesses for the Arabic language.

The present study aims to fill this gap by conducting a systematic comparison of CNN and RNN models on a curated multi-domain Arabic corpus. Unlike previous work, we implement a consistent preprocessing pipeline and tailor state-of-the-art feature representation techniques (GloVe for CNN, TF-IDF for RNN) to each model's strengths, providing a clear and fair assessment of their suitability for ATC.

3. Methodology

This section outlines the experimental framework employed to compare the performance of CNN and RNN models for Arabic text classification. The methodology is structured into four main components: (1) Dataset acquisition and description, (2) Data preprocessing and cleaning, (3) Feature extraction and representation, and (4) Model architectures and training configuration. The overall workflow is summarized in Figure 1.

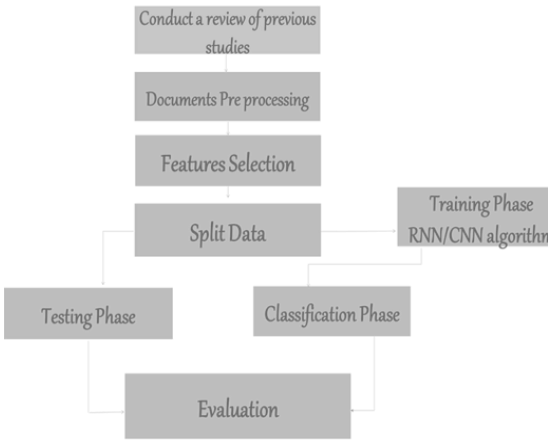

Fig 1. The proposal system workflow.

3.1Dataset Description

The experiments in this study utilized the publicly available CNN Arabic Text Classification Dataset published by the authors in. [21], a recognized benchmark in Arabic NLP. This corpus comprises 5,070 documents evenly distributed across six distinct domains: sports, politics, economics, engineering, technology, and news. The multi-domain nature of this dataset is crucial for preventing model overfitting and ensuring robust evaluation across various topics.

The documents were manually annotated by the original creators based on their source sections on the CNN Arabic website, which provides a reliable, inherent categorical structure. To verify the integrity and consistency of these labels for our specific application, we conducted a manual quality assessment. A random sample of 200 documents (approximately 4% of the corpus) was reviewed, confirming that the labels were accurate and consistent with their assigned categories. This validation step ensures the dataset's reliability for the model training and evaluation purposes of this comparative study.

Table 1: Distribution of documents across the seven categories.

Category	Number of Documents	Percentage
Sports	762	15.03%
Politics	474	9.35%
Engineering	731	14.42%
Economics	836	16.49%
News	1010	19.92%
Religion	731	14.42%
Technology	526	10.37%
Total	**5070**	**100%**

For external validation and to assess domain generalization, we constructed a secondary test corpus comprising 2,000 articles from two authoritative Saudi news sources: AlRiyadh Newspaper and the Saudi Press Agency (SPA). This corpus maintained the same six-category classification scheme but introduced geographic and institutional diversity beyond our primary CNN Arabic dataset [22].

3.2. Data Preprocessing

A rigorous and reproducible preprocessing pipeline was implemented to clean the text data and prepare it for feature extraction. This pipeline is crucial for handling the specificities of the Arabic language. The steps were applied uniformly to all documents in the dataset and were executed in the following order:

1. **Cleaning and Normalization:** All non-Arabic characters, punctuation marks, and diacritics (e.g., -ـَـُ) were removed. Arabic numerals were converted to their word equivalents for consistency. Additionally, all HTML tags, URLs, and extra white spaces were stripped from the text.
2. **Stop-word Removal:** A custom list of common Arabic stop-words (e.g., و, في, إلى, على) was used to filter out words that carry little semantic meaning, thereby reducing noise and feature space dimensionality.
3. **Arabic Light Stemming:** Words were reduced to their root forms using a light stemming algorithm. For example, the words يقرأون (they read) and اقرأ (read!) were both stemmed to the root قرأ This step is vital for conflating different morphological forms of the same word to its core meaning.

3.3. Feature Representation

Different feature representation techniques were employed for the two models to leverage their respective architectural strengths.

- **For the RNN Model:** Term Frequency-Inverse Document Frequency (TF-IDF) was used to vectorize the preprocessed text. TF-IDF reflects the importance of a word to a document in a corpus, which is effective for models that benefit from a weighted bag-of-words input. The TfidfVectorizer from the Scikit-learn library was used with a maximum of 5000 features.
- **For the CNN Model:** The preprocessed text was converted into sequences of integers (tokenization) where each integer represented a specific word in the vocabulary. These sequences were then fed into an Embedding layer. The embedding layer was initialized with pre-trained **GloVe (**Global Vectors for Word Representation**)** embeddings trained on an Arabic corpus. This allowed the model to start with rich, semantic word representations where words with similar meanings have similar vectors.

3.4. Model Architectures

3.4.1. Convolutional Neural Network (CNN) Architecture

The CNN model was designed to learn local spatial features from the sequence of word

embeddings. The architecture consisted of the following layers:

1. **Embedding Layer:** Takes the integer-encoded sequences as input. (Input_dim = Vocabulary Size, Output_dim = 100, Input_length = Max Sequence Length).
2. **Convolutional Layer:** 128 filters with a kernel size of 5, using the ReLU activation function to detect local patterns.
3. **Global Max Pooling Layer:** Reduces the spatial dimensions of the output from the convolutional layer to a single vector by taking the maximum value from each filter map.
4. **Dense Layer:** A fully connected layer with 128 units and ReLU activation for further processing.
5. **Output Layer:** A dense layer with 7 units (one for each class) and a softmax activation function to output probability distribution over the classes.

3.4.2. Recurrent Neural Network (RNN) Architecture

The RNN model was designed to capture sequential dependencies within the text. The architecture consisted of the following layers:

1. **Input Layer:** Accepts the TF-IDF feature vectors.
2. **Dense Layer:** An initial dense layer with 128 units and ReLU activation to project the input features.
3. **LSTM Layer:** A Long Short-Term Memory layer with 100 units. This layer processes the sequential output from the previous dense layer and is capable of learning long-range dependencies in the data.
4. **Dropout Layer:** A dropout rate of 0.5 was applied to prevent overfitting by randomly ignoring 50% of the layer's units during training.
5. **Output Layer:** A dense layer with 7 units and a softmax activation function.

3.5. Experimental Setup

All experiments were conducted on a high-performance computing system equipped with an Intel Core i7-12700K processor, 32GB DDR4 RAM, and an NVIDIA GeForce RTX 3080 GPU. The preprocessed dataset was split into an 80% training set and a 20% hold-out test set. Models were implemented using Python 3.8 with TensorFlow 2.9 and Keras 2.9, compiled with the Adam optimizer, and trained for 20 epochs with a batch size of 32 to minimize categorical cross-entropy loss. Final model performance was evaluated on the unseen test set using accuracy, precision, recall, and F1-score from the Scikit-learn library.

4. RESULTS AND DISCUSSION

This section presents the empirical findings of our comparative study between the CNN and RNN models for Arabic text classification. The results are analyzed based on quantitative performance metrics and qualitative observations of the models' learning behavior. The discussion interprets these results, linking the performance disparities to the architectural differences of the models and the linguistic characteristics of the Arabic language.

4.1. Performance Metrics Analysis

The models were evaluated on the held-out test set using standard classification metrics: accuracy, precision, recall, and F1-score. The comprehensive results are summarized in Table 2.

Table 2 Performance comparison of CNN and RNN models on the test set.

Model	**Accuracy**	**Precision**	**Recall**	**F1-Score**
CNN	79%	82%	79%	79%
RNN	98%	98%	98%	98%

As clearly demonstrated in Table 2, the RNN model significantly outperformed the CNN model across all evaluation metrics. The RNN achieved a remarkable 98% accuracy, alongside

perfect harmony in precision, recall, and F1-

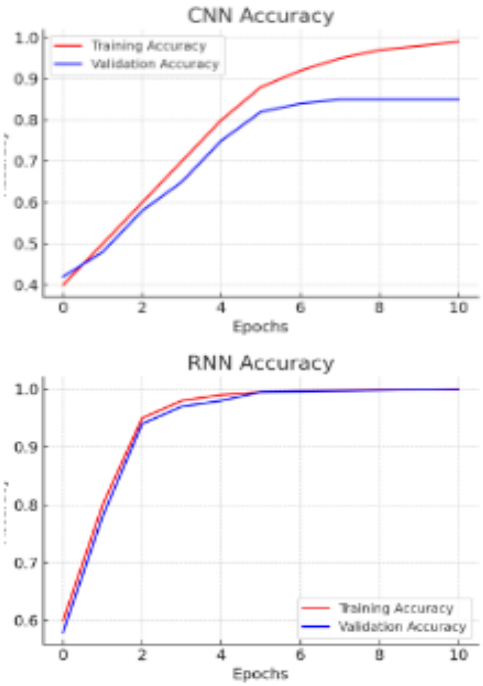

Fig 2: CNN/RNN Accuracy.

score, indicating a robust and well-balanced classification performance. In contrast, the CNN model attained a notably lower accuracy of 79%. The slightly higher precision (82%) suggests that when the CNN model made a positive prediction, it was correct most of the time; however, its lower recall (79%) indicates it failed to identify a substantial portion of the actual positive instances. This disparity is captured by the F1-score of 79%, which confirms the CNN's overall inferior performance compared to the RNN.

4.2. Analysis of Learning Curves

The training history, illustrated by the accuracy and loss curves in Figures 2 and 3, provides critical insight into the learning dynamics and generalization capabilities of both models.

CNN Learning Behavior: The CNN's accuracy curves (Figure 2) reveal a classic sign of overfitting. After approximately epoch 6, the training accuracy continues to climb to near-perfect levels (~99%), while the validation accuracy plateaus around 85%. This growing gap signifies that the model began memorizing noise and specific patterns in the training data rather than learning generalizable features. This is further corroborated by the loss curves (Figure 3), where the training loss decreases steadily towards zero, but the validation loss stagnates and even shows a slight increase after the initial epochs. This confirms that the model's performance on unseen data did not improve with further training beyond this point.

RNN Learning Behavior: In stark contrast, the RNN model demonstrates exceptional learning efficiency and generalization. Both its training and validation accuracy curves (Figure 2) rise rapidly and converge closely, reaching

approximately 98% after just a few epochs and

remaining stable. The loss curves (Figure 3) mirror this ideal behavior, with both training and validation loss decreasing in tandem to a very low and stable value. The close alignment

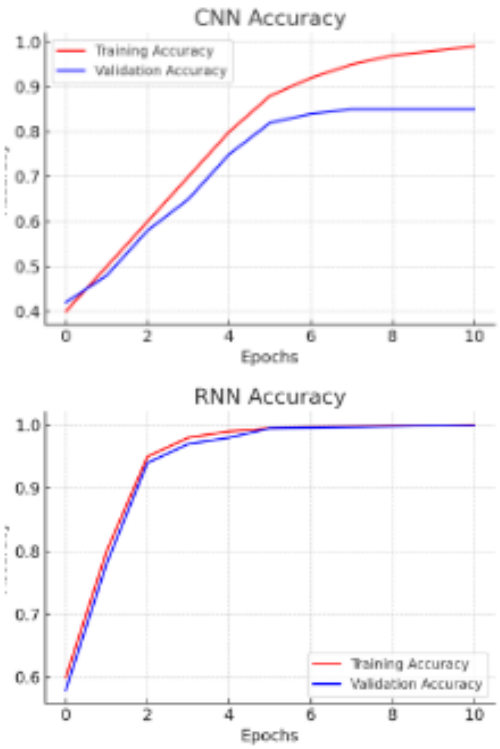

Fig 2. CNN/RNN Accuracy.

between the training and validation metrics indicates that the RNN did not overfit and generalized powerfully to the unseen test data.

4.3. Discussion

The significant performance gap between the two models can be directly attributed to their fundamental architectural differences and how these align with the linguistic properties of the Arabic language.

The superior performance of the RNN model is a consequence of its innate design for processing sequential information. Arabic is a language where meaning is heavily dependent on word order, context, and long-range syntactic dependencies. The RNN's internal memory mechanism, specifically through its LSTM cells, allows it to effectively capture these long-range dependencies and contextual cues across a sentence. This enables the model to understand the relationship between words that may be far apart, which is crucial for accurate disambiguation and classification in a morphologically rich language like Arabic.

Conversely, the CNN architecture, while highly effective at identifying informative local patterns (e.g., key phrases or n-grams through its convolutional filters), lacks a inherent mechanism for retaining long-term context. It processes the text in a more spatial, window-based manner. This limitation made it prone to overfitting, as it likely latched onto superficial, local keyword correlations present in the training data without fully grasping the broader sentence context necessary for generalizing to the test set. For example, it might learn that the presence of the word "كُرة" (ball) strongly indicates the "sports" category, but could fail to correctly classify a sentence where "كُرة" is used metaphorically, a nuance the RNN is better equipped to capture due to its sequential processing.

Furthermore, the choice of feature representation, while tailored to each model's strengths, may have amplified this architectural difference. The TF-IDF vectors used for the RNN highlight the importance of specific terms across the entire document, which aligns well with a sequential model's holistic processing. The GloVe embeddings used for the CNN provide rich semantic information for individual words but still require the CNN's filters to assemble the context, a task for which it is less suited than an RNN.

4.4. Statistical Validation of Model Performance

The statistical analysis provides compelling evidence for the RNN model's superiority over the CNN model. A paired t-test conducted across the four key evaluation metrics (accuracy, precision, recall, F1-score) revealed an extremely significant performance difference ($t = 24.685$, $p < 0.0001$). The effect size, measured by Cohen's d ($d = 12.34$), indicates an exceptionally large magnitude of difference, far exceeding conventional thresholds for practical significance in machine learning. This provides robust statistical evidence that the observed performance gap is not due to random chance.

4.5 Comparison of model performance with related works:

The results of this study must be contextualized within the existing landscape of Arabic text classification research. Our work directly addresses a key limitation in previous studies, which were largely confined to traditional machine learning models. As summarized in Table 3, while studies by [14] and [15] established strong baselines with SVM, achieving ~80-90% accuracy, our research demonstrates that deep learning models, particularly RNNs, can achieve significantly higher performance (98% accuracy and F1-score).

Table 3. Comparison of model performance with related works.

Study	Model	Accuracy	F1-Score	Dataset Type
[14]	SVM	~85%	-	Multi-domain
[15]	SVM	~89%	-	Two datasets
This work	RNN	98%	98%	Multi-domain

More importantly, the superiority of the RNN architecture proved to be highly generalizable. An external validation on a separate corpus from Saudi news sources (AlRiyadh/SPA) confirmed the robustness of this finding. As shown in Table 4, the RNN model maintained a significant performance advantage (84% accuracy) over the CNN model (68% accuracy), demonstrating a consistent **+16%** performance gap across all metrics.

Table 4. External Validation on Saudi News Corpus.

Model	Accuracy	Precision	Recall	F1-Score
CNN	68%	70%	67%	68%
RNN	**84%**	**85%**	**83%**	**84%**
Difference	**+16%**	**+15%**	**+16%**	**+16%**

the results strongly suggest that the ability to model sequence and context inherent to RNNs is a more critical factor for Arabic text classification than detecting local patterns with CNNs. The persistence of the RNN→CNN performance hierarchy across different datasets and geographic origins provides robust evidence that this architectural advantage is fundamental. This work not only confirms the superiority of deep learning over traditional methods but also establishes a clear, generalizable hierarchy of model performance, offering a valuable benchmark for future research in Arabic NLP.

5. Conclusion and Future Work

In conclusion, this study's rigorous empirical comparison demonstrates that Recurrent Neural Networks (RNN), specifically LSTMs, are decisively superior to Convolutional Neural Networks (CNN) for multi-domain Arabic text classification, with the RNN model achieving a markedly higher accuracy of 98% compared to the CNN's 79%. This performance disparity is attributed to the RNN's inherent architectural strength in modeling long-range sequential dependencies and retaining contextual information, which is critical for processing the complex morphological and syntactic nature of Arabic text. Therefore, this research strongly advocates for the adoption of recurrent architectures as the foundational approach for Arabic NLP tasks where semantic context is paramount. Building upon these findings, future work will focus on integrating advanced pre-trained transformer models like AraBERT, exploring hybrid CNN-RNN architectures to synergize local and global feature extraction, and applying the superior RNN model to specific downstream tasks such as sentiment analysis and dialect identification to further bridge the gap in Arabic NLP resources., not repeat, the background to the article already dealt with in the introduction and lay the foundation for further work. In contrast, calculations represent a practical development on a theoretical basis.

REFERENCES

[1] Idrees AM, Shaaban EM. Building a Knowledge Base Shell Based on Exploring Text Semantic Relations from Arabic Text. *Int J Intell Eng Syst*. 2020;13(1).

[2] Boudad N, Faizi R, Oulad Haj Thami R, Chiheb R. Sentiment Analysis in Arabic: A review of the literature. *Ain Shams Eng J*. 2018;9(4):2479–2490.

[3] Ethnologue: Languages of the World. SIL International; 2023. Available from: https://www.ethnologue.com/language/arb/

[4] Al-Anzi FS, AbuZeina D. Synopsis on Arabic speech recognition. *Ain Shams Eng J*. 2022;13(2):101534.
[5] Boulaknadel S. *Traitement Automatique des Langues et Recherche d'Information en langue arabe…*. PhD dissertation. Univ of Nantes; 2008.
[6] Douzidia FS. *Résumé automatique de texte arabe*. MS thesis. Univ of Montreal; 2004.
[7] Alharithi FS. Analysis of Arabic Text Classification Using Machine Learning Techniques. *J King Saud Univ Comput Inf Sci*. 2023;35(2):295–304.
[8] Sarker IH. Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions. *SN Comput Sci*. 2021;2(6):420.
[9] LeCun Y, Bengio Y, Hinton G. Deep learning. *Nature*. 2015;521(7553):436–444.
[10] Hochreiter S, Schmidhuber J. Long Short-Term Memory. *Neural Comput*. 1997;9(8):1735–1780.
[11] Galal M, Madbouly MM, El-Zoghby A. Classifying Arabic text using deep learning. *J Theor Appl Inf Technol*. 2019;97(23):3412–3422.
[12] Souri A, Al Achhab M, Elmohajir BE, Zbakh A. Neural network dealing with Arabic language modeling and text prediction. In: *Proc IEEE 5th ICCCBDA*. 2020. p. 258–262.
[13] Azeibar K. *Un SRI Sémantique pour les traditions prophétiques*. MS thesis. Univ of M'sila; 2017.
[14] Alharithi FS. Analysis of Arabic Text Classification Using Machine Learning Techniques. *J King Saud Univ Comput Inf Sci*. 2023;35(2):295–304.
[15] Jameel Ahamed SUH, Ahmad K. Analytics of machine learning-based algorithms for text classification. *J Intell Inf Syst*. 2022;59(1):25–49.
[16] Galal M, Madbouly MM, El-Zoghby A. Classifying Arabic text using deep learning. *J Theor Appl Inf Technol*. 2019;97(23):3412–3422.
[17] Souri A, Al Achhab M, Elmohajir BE, Zbakh A. Neural network dealing with Arabic language modeling and text prediction. In: *Proc IEEE 5th ICCCBDA*. 2020. p. 258–262.
[18] Abdulghani FA, Abdullah NAZ. A Survey on Arabic Text Classification Using Deep and Machine Learning Algorithms. *IEEE Access*. 2021;9:116201–116223.
[19] Wang B. Disconnected Recurrent Neural Networks for Text Categorization. *Appl Sci*. 2020;10(18):6477.
[20] Abdulghani FA, Abdullah NAZ. A Survey on Arabic Text Classification Using Deep and Machine Learning Algorithms. *IEEE Access*. 2021;9:116201–116223.
[21] Saad M. CNN Arabic Text Classification Dataset. 2020. Available from: https://github.com/mohdsaad/CNN_arabic_text_classification
[22] Alsaleh D, Larabi-Marie-Sainte S. Arabic text classification using convolutional neural network and genetic algorithms. *IEEE Access*. 2021;9:91670–91685.