International Journal of Theoretical and Applied Research (IJTAR)

ISSN: 2812-5878

Homepage: https://ijtar.journals.ekb.eg

Review article

A Hybrid Approach for Fake News Detection on Cloud Computing

Abdullah A. Abdullah [1], Dina R Salem[1], AbdAllah A. AlHabshy[2], Kamal A. ElDahshan[2]

[1]*School of Computer Science, Canadian International College – CIC, Cairo, Egypt*
[2]*Department of Mathematics, Faculty of Science, Al-Azhar University, Cairo, Egypt*

ARTICLE INFO

Received 20/07/2025
Revised 08/08/2025
Accepted 04/09/2025

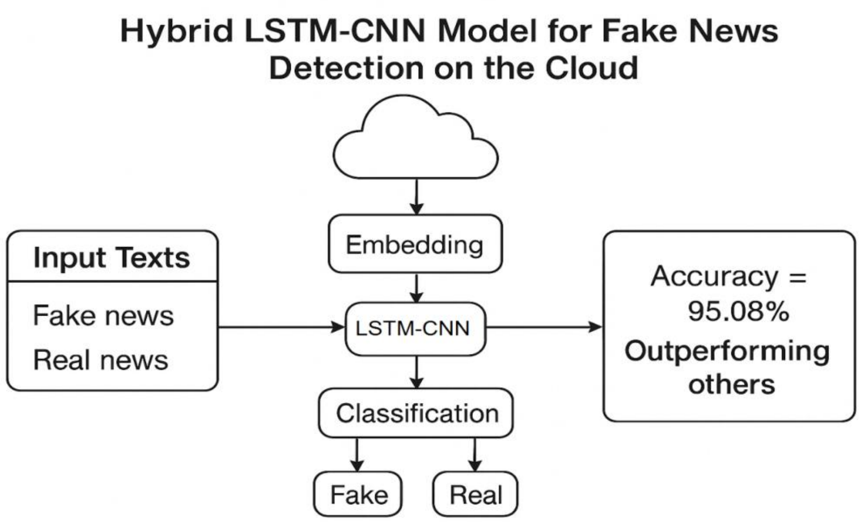
Keywords

Fake news
Machine learning
Deep learning
Cloud computing
CNN
LSTM

ABSTRACT

The rapid spread of fake news on digital platforms has become a global crisis that threatens public trust and decision-making. Some people make it up for attention or political gain. Machine learning and deep learning techniques have been developed to detect fake news. However, they tend to generate inaccurate reports. To detect fake news, this paper proposes a hybrid model that combines CNN and LSTM frameworks on Google Cloud. This model was able to categorize news with better accuracy than using each model individually. The model was tested and trained on a fake news classification dataset. We used different evaluation metrics (precision, recall, F1 metric, etc.) to measure the efficiency of the model. The democratization of content creation through social media platforms, blogs, and online news portals has enabled unprecedented access to real-time information. We believe that a hybrid classifier-based system has a higher level of reliability. The results and discussion section provides evidence for claims and establishes the objectives.

Graphical abstract



1. Introduction

In today's Connected World, the spread of fake news has become a dangerous issue, not just about incorrect information. This includes deliberate manipulation and spread of false or misleading information, often using social media platforms where information can go viral in minutes. Fake news is especially dangerous because it is not only wrong but also can distort meaning, disrupt political processes, and even increase health and economic crises. This problem is more complex in how individuals easily connect and follow emotional or sensational titles, often without verifying accuracy [1].

⁎ *Corresponding author*
E-mail address: abdallah_lovers12@yahoo.com

DOI: **10.21608/IJTAR.2025.402732.1136**

Platforms such as Facebook and Twitter produce part of millions of user-related materials daily, including news articles, opinions, and manipulated media [2]. This unnecessary and dynamically changed information creates serious challenges for traditional fact- and rule-based identity systems, which are not equipped to handle such scale or diversity in real time. This variety of high speed, volume, and fake news makes it a prominent example of big data.

Finding fake news is not just a technical challenge—it also reveals important moral dilemmas. There is a tension between ensuring the integrity of information at the heart of debate and preserving fundamental rights such as freedom of speech. The risk of wrong real material (false positivity) adds to another layer of complexity. Since these systems are used quickly on the scale, it becomes necessary to design the tracking that detects and respects the transparent, fair, and diverse approaches [3].

A scalable and effective solution is using cloud computing for the growing problem of identifying fake news. The distributed and adaptable architecture makes it possible to distribute real-time models that can effectively analyze the data in large quantities without giving up the speed [4, 5].

To lessen the impact of bogus news before it becomes viral, early detection is essential. Machine learning-backed proactive detection techniques can lessen the negative social and psychological effects of misleading narratives [6].

In recent years, deep learning has emerged as a powerful approach to dealing with complex problems in natural language processing (NLP) and text analysis. Unlike traditional machine learning methods, which depend too much on handmade functions, deep learning models can automatically learn abstract representations directly from raw text [7].

Convolutional Neural Networks (CNNs) are widely used in NLP tasks due to their effectiveness in extracting local features from text [8]. Long Short-Term Memory (LSTM) networks are a type of recurrent neural network designed to capture sequential dependencies in data [9].

While CNN and LSTM designs have proven effective in certain elements of text analysis, they do have inherent limits when utilized individually. CNNs excel at identifying localized features but may neglect larger contextual correlations. In contrast, LSTMs are well-suited for modeling long-term dependencies, but they may overlook strong, localized signals. This study fills these gaps by presenting a hybrid LSTM-CNN model that enhances the accuracy and dependability of fake news identification by combining the best features of both approaches. The cloud-based infrastructure in which the model is developed enables scalable deployment and quick, resource-efficient inference. Significantly, experimental findings show that the hybrid strategy routinely outperforms the separate CNN and LSTM models, underscoring its potential as a reliable and expandable solution for disinformation detection tasks in the real world.

The motivation behind this study stems from the pressing need to improve the reliability and scalability of fake news detection systems. Traditional techniques cannot capture deep linguistic features and contextual semantics efficiently. Given the evolving nature of misinformation, combining powerful deep learning models with scalable infrastructure can result in more adaptable and resilient detection frameworks.

The main contributions of this study are as follows:

- Present a hybrid LSTM-CNN model that enhances fake news detection by combining local text features (CNN) with long-term contextual patterns (LSTM), thus improving both accuracy and robustness..
- Implementation in a cloud computing environment, leveraging Google Colab to facilitate reproducibility, scalability, and resource efficiency, making it accessible even to researchers with limited hardware.
- Evaluation on a real-world dataset from Kaggle, with detailed performance analysis using standard metrics such as accuracy, precision, recall, and F1-score.
- Comparison with baseline models, including machine learning and individual deep learning architectures, to highlight the effectiveness of the hybrid approach.

The remainder of the paper is organized as follows: The "Related Work" in Section II examines previous research on Deep Learning Techniques, Hybrid CNN-LSTM Models, and Fake News Detection on Cloud Platforms. The "Proposed Methodology" in Section III provides detailed information about the proposed approach. In the "Results and Evaluation" in Section IV, it analysis and discusses the findings and their reasons. The "Comparison with other Studies" in Section V shows a comparison between the proposed model wither other previous works. Finally, the "Conclusion and future work" in Section VI summarizes the main points of the research.

2. Related work

This section reviews recent and relevant literature on fake news detection; this article discusses the most recent advances in the use of deep learning to detect fake news and the growing role of cloud computing in scalable deployment.

Alhindi, T., Kalita, J., & Sajjad, H. classified fake news using Support Vector Machines (SVM) and TF-IDF features on the LIAR dataset. While the approach provided reasonable accuracy, it struggled to understand context and did not scale well to real-world scenarios due to restricted semantic representation and dataset diversity [10].

Ehzaam *et al.* evaluated the usage of Random Forest classifiers on Kaggle's "Fake News Classification" dataset. Although the model provided baseline accuracy, it could not predict contextual and sequential connections, demonstrating the limitations of typical machine learning approaches in dealing with complex linguistic structures [11].

Ke, S., *et al.* suggested a neural network model that used TF-IDF features from the same Kaggle dataset. The method obtained an accuracy of 86.22%, but did not investigate more advanced deep learning architectures like CNNs or LSTMs, restricting its generalization [12].

Kesarwani et al. compared Naive Bayes, Decision Tree, and Random Forest classifiers on the "Fake News Classification" dataset. Their investigation revealed that Random Forest outperformed the others. However, the absence of hybrid or deep learning methodologies limited the model's capacity to learn high-level text representations [13].

Jain et al. compared CNN and LSTM architectures using the Fake News Net dataset. Their findings demonstrated that both methods had different advantages; nevertheless, the study did not assess real-time deployment or scalability, particularly in a cloud context [14].

Biradar and Patil developed a hybrid CNN-LSTM model trained on the ISOT dataset hosted on Kaggle. The hybrid approach improved accuracy by leveraging CNN for feature extraction and LSTM for temporal dependencies. Nonetheless, the system lacked cloud-based implementation and evaluation under real-time constraints [15].

Gupta explored the use of LSTM and RNN models implemented through GitHub using the Kaggle fake news dataset. Models achieved competitive performance, yet the absence of detailed evaluation metrics and lack of reproducibility limited their research applicability [16].

Hossain et al. developed a deep learning-based system for detecting bogus news in Bangla using CNN and LSTM. The study showed that deep models can be useful in non-English environments, but their scalability and cross-lingual adaptability are limited [17].

Wahab and Fong proposed an ensemble RoBERTa-based deep learning system that was evaluated on the ISOT dataset. The model was highly accurate, but it required significant computational resources and lacked integration with cloud platforms for distributed training and deployment [18].

Wang and Li built a CNN-LSTM model with dual attention that was trained using LIAR and custom data. The attention mechanism improved performance; however, the model was not evaluated for latency or cloud efficiency [19].

Utku, D., created a multichannel CNN-LSTM hybrid model using the FakeNewsNet dataset. The study showed improvements in accuracy and contextual understanding, but did not benchmark against transformer models or evaluate performance in cloud environments [20].

Xu et al. improved the CNN-LSTM hybrid architecture by including attention layers and tested it on the LIAR data set. The study underlined the significance of attention in capturing crucial linguistic signals, but it did not include any performance evaluation in cloud-based situations [21].

Hossain *et al.* presented a multichannel CNN-LSTM architecture trained on a Bangla false news dataset. Despite its good performance in that setting, it did not compare findings to transformer-based models, which limited insights about model competitiveness [22].

Zamani *et al.* aimed to develop deep learning-based fake news detection systems for cloud infrastructure. The work helped to understand the performance implications of distributed systems, but it did not incorporate hybrid models or assess content-level accuracy [23].

Finally, the literature identifies various shortcomings, including limited hybrid model integration, inconsistent use of scalable deployment platforms, and a lack of extensive testing across multiple datasets. These shortcomings highlight the need for a cloud-optimized hybrid LSTM-CNN model capable of robustly detecting fake news at scale.

To provide a comprehensive overview of recent advancements in fake news detection, Table I outlines major studies conducted between 2020 and 2025. The table summarizes models and methodologies utilized, the datasets used, and the major constraints identified in each study. This overview aids in identifying common difficulties, such as inadequate contextual knowledge, scalability, and the lack of cloud-based deployment, all of which the suggested hybrid CNN-LSTM architecture seeks to overcome.

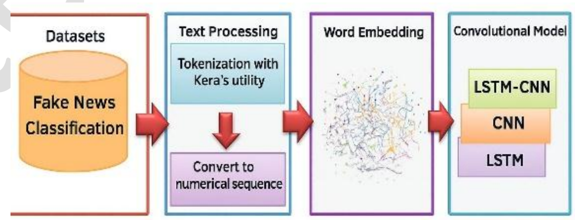


Fig. 1. Proposed Methodology

3. Proposed methodology

This section presents a comprehensive methodology for detecting fake news using deep learning algorithms, a structured pipeline implemented in a cloud environment to ensure scalability and efficiency. The suggested approach uses a systematic pipeline that includes data preprocessing, model development, and evaluation phases as shown in Fig. 1.

The process begins with the preparation of a dataset specifically designed to detect fake news. In the text processing phase, tokenization is performed using Keras tools to split the text into smaller units, which are then converted into numerical sequences for computational analysis. Word embedding techniques are applied to convert these tokens into dense vector representations, capturing the semantic relationships between words. The core of the model is based on a hybrid LSTM-CNN architecture, combining Long Short-term Memory (LSTM) networks to capture long-term dependencies and Convolutional Neural Networks (CNN) to extract local features from text. Alternatively, independent CNN or LSTM models can be deployed depending on task requirements. This integrated approach, implemented on a cloud platform, aims to enhance the accuracy and performance of fake news classification by leveraging the strengths of deep learning techniques in processing and analyzing textual data.

TABLE I. SUMMARY OF LITERATURE REVIEW ON FAKE NEWS DETECTION

Ref	Authors	Year	Models and Techniques Used	Datasets	Limitations
[10]	Alhindi et al.	2020	SVM with TF-IDF	LIAR	Poor contextual understanding, limited scalability
[11]	Ehzaam et al.	2021	Random Forest	Kaggle (Fake News Classification)	Limited to traditional ML; lacks deep contextual modeling
[13]	Kesarwani et al.	2022	Naive Bayes, Random Forest	Kaggle (Fake News Classification)	Limited accuracy; no hybrid or cloud deployment
[14]	Jain et al.	2022	Comparison of CNN and LSTM	FakeNewsNet	Focus on accuracy only, lacks deployment focus
[15]	Biradar & Patil	2022	Hybrid CNN + LSTM	Kaggle (ISOT)	No cloud integration, not tested on large-scale environments
[16]	Gupta (GitHub)	2022	LSTM, RNN	Kaggle (Fake News Classification)	Limited evaluation metrics and documentation
[17]	Hossain et al.	2022	Deep learning (CNN, LSTM)	Self-collected (Bangla news)	Language-specific, lacks real-time scalability
[18]	Wahab & Fong	2023	RoBERTa, Ensemble DL	ISOT	High computation, no cloud scalability
[19]	Wang & Li	2023	Dual Attention CNN-LSTM	LIAR, custom	Lacks cloud validation, not optimized for latency
[20]	Utku	2024	Hybrid multichannel CNN-LSTM	Kaggle (FakeNewsNet)	No transformer comparison, no scalability study
[21]	Xu et al.	2025	Attention-enhanced CNN-LSTM	LIAR	Lacks cloud benchmarking, limited dataset variety
[22]	Hossain et al.	2025	Multichannel Combined CNN-LSTM	Bangla fake news dataset	Language-specific, lacks transformer comparison
[23]	Zamani et al.	2025	Deep learning on the cloud	Social media streams	No hybrid model, focused on infrastructure, not accuracy

A. Preprocessing Data

In the preprocessing step, it is important to transform the data text into a clean corpus before feeding models [24]. To accomplish this, we load the datasets and combine the title and text columns into a single content column while dropping unnecessary columns. For model preparation, we use the TensorFlow tokenized to convert the text data into sequences of integers, which are then padded to ensure uniform length across all sequences. The preprocessing data workflow is shown in Fig. 2.

Following the completion of the preprocessing stages, the data is divided into testing and training sets in a ratio of 80 to 20. Following that, the features are suitably reshaped to satisfy the input specifications of the various deep learning architectures (LSTM, CNN, and hybrid models).

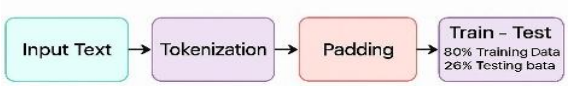


Fig. 2. Data preprocessing workflow

B. Model Training

Three deep learning architectures, LSTM, CNN, and hybrid (LSTM-CNN) models, are implemented in the suggested technique. Each architecture is intended to capture a distinct facet of the fake news detection patterns. The following sections explain LSTM, CNN, and LSTM-CNN details.

1) Long Short-Term Memory (LSTM)

Through the analysis of input feature sequences, the LSTM model is particularly made to detect fake news, as shown in Fig. 3. It begins with an embedding layer that maps tokenized words into dense vectors, transforming textual data into numerical representations suitable for neural networks. Next, a bidirectional LSTM layer processes these embeddings in both forward and backward directions, capturing long-range contextual relationships between words—critical for understanding nuanced misinformation patterns. To mitigate overfitting, a dropout layer randomly deactivates half the neurons during training, enhancing generalization. Finally, a Dense output layer with sigmoid activation produces probability scores for classification.

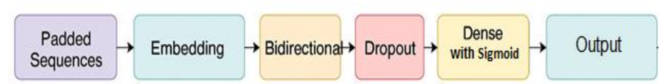


Fig. 3. Workflow of the LSTM Model

2) Convolutional Neural Network (CNN)

The CNN model uses a number of layers, each with a distinct function, to extract spatial characteristics from the input data, as shown in Fig. 4. The workflow begins with an embedding layer that converts the input data into meaningful numerical representations. Subsequently, a 1D Convolutional Layer (CNN Conv1D) extracts crucial features from these embeddings. Following this, a Global AveragePooling1D layer reduces the dimensionality of the extracted features. A Dropout layer is then applied to prevent overfitting, and finally, the data passes through a Dense layer with a sigmoid activation function to generate the final outputs, a probability score between 0 (fake) and 1 (real).

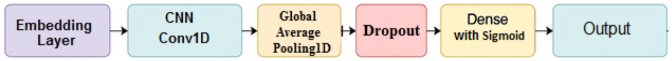


Fig. 4. Workflow of the CNN Model

3) A Hybrid (LSTM-CNN) Model

This hybrid model to detect fake news, as shown in Fig. 5. The process initiates with an embedding layer to transform the input text into meaningful numerical representations. Subsequently, a bidirectional LSTM is employed to capture contextual information by processing the input text sequence in both directions, generating a sequence of contextualized embeddings. These embeddings are then fed into a 1D Convolutional Neural Network (CNN) to extract local patterns within the contextualized sequences. Finally, Global Max Pooling aggregates the most salient features learned by both the LSTM and CNN, which are then passed through dense layers to produce a probability indicating whether the news is fake or real. This approach effectively combines the LSTM's capacity to understand sequential context with the CNN's strength in identifying local, indicative patterns.

Fig. 5. Workflow of the LSTM-CNN Model

C. Cloud-Based Implementation

The Google Colab platform is employed to implement the models. Google Colab is a cloud environment that hosts Jupyter notebooks that integrate GPUs (Graphical Processing Units) and TPUs (Tensor Processing Units), among other computing resources. It does not require any setup. Python was used to write the experiment's code. Google Colab works particularly effectively with ML.

4. Results and Evaluation

In this section, we present and analyze the results of the study, detailing the components that contribute to the performance of the proposed intrusion detection models for fake news. Firstly, a description is used for the Fake News Classification dataset for training and testing, followed by an outline of the experimental setup, including the specifics of the training process, the hyperparameters tuned for each model. Next, we discuss the performance measurement criteria used to evaluate model effectiveness and robustness. Finally, we report the results obtained from experiments and provide a comprehensive discussion, highlighting the comparative strengths, practical implications, and potential limitations of each approach within the context of real-world fake news detection.

A. Dataset Description

A specific dataset has been used in this work to address the fake news classification, which was proposed by Kaggle and is openly available; it was downloaded from [25]. The dataset contains just over 45,000 unique news articles specifically designed for binary classification of misinformation in digital media, which are labeled with 1 if the article is true (real) and 0 if it is false (fake), with the following attributes: id, title, author, text, and label. The dataset is split into three CSV files, as shown in Table II.

TABLE II. DATASET INFORMATION

Dataset	**Samples**	**Columns (Before)**	**Columns (After)**
Train Set	24,353	id, title, text, label	content, label
Validation	8,117	id, title, text, label	content, label
Test Set	8,117	id, title, text, label	content, label

TABLE III. Comparison between the hyperparameters of each model

Hyperparameter	**LSTM Model**	**CNN Model**	**Hybrid (LSTM-CNN) Model**
Layers	Embedding, BiLSTM, Dropout, Dense	Embedding_1, Conv1D, Global_Average_Pooling, Dropout_1, Dense_1	Embedding_2, BiLSTM_1, Conv1D_1, Max_Pooling1D, Global_Max_Pooling1D, Dense_2, Dense_3
Number of Layers	4	5	7
Activation Function	ReLU (hidden), Sigmoid (output)	ReLU (hidden), Sigmoid (output)	ReLU (hidden), Sigmoid (output)
Optimizer (Adam)	lr=0.001	lr=0.001	lr=0.001
Loss Function	Binary Cross-entropy	Binary Cross-entropy	Binary Cross-entropy
Batch Size	32	32	32
Epochs	25	25	25
Dropout	0.5	0.5	0.5
No. of Filters (CNN)	-	8	256
Kernel Size (CNN)	-	3	3
Pooling Layer (CNN)	-	GlobalAveragePooling1D	MaxPooling1D(pool_size=2)
LSTM Units	8 (bidirectional)	-	128 (bidirectional)
Input Shape	(None, max_length)	(None, max_length)	(None, max_length)
Output Layer	Dense(1, 'sigmoid')	Dense(1, 'sigmoid')	Dense(1, 'sigmoid')

This dataset stands out by its balanced representation of genuine and fabricated articles representing a variety of disciplines, including politics, health, and entertainment, providing researchers with a solid benchmark for creating and assessing cutting-edge fake news detection models.

B. Hyperparameters of each Model

As shown in Table III, we examine the key hyperparameters that govern their architecture and training process. These consist of the number and layer types, activation functions, optimizer settings, batch size, dropout rates, and the output layer configurations.

Accuracy (Acc.) = (TP+ TN) / (TP + TN + FP + FN) (1)

Precision (Prc.) = TP (TP + FP) (2)

Recall (Rc.) = TP (TP + FN) (3)

F1 Score (F1.) = 2∗ (Prc. ∗ Rc.) / (Prc. + Rc.) (4)

C. Evaluation Matrics

Various metrics are used to assess the performance of this work. These metrics are based on the confusion matrix, that matrix depends on four important parameters: TP, TN, FP, and FN, explained in Table IV. Moreover, the Accuracy (Acc.) was the main performance metric selected to measure the model. The key metrics for evaluating ML models include Accuracy (Acc.), Precision (Prc.), Recall (Rc.), and the F1 score (F1.). These metrics, calculated using specific equations [26]:

TABLE IV. CONFUSION MATRIX PARAMETERS

TP	When a real article is correctly predicted
FP	When a real article is wrongly predicted
TN	When a fake article is correctly predicted
FN	When a fake article is wrongly predicted

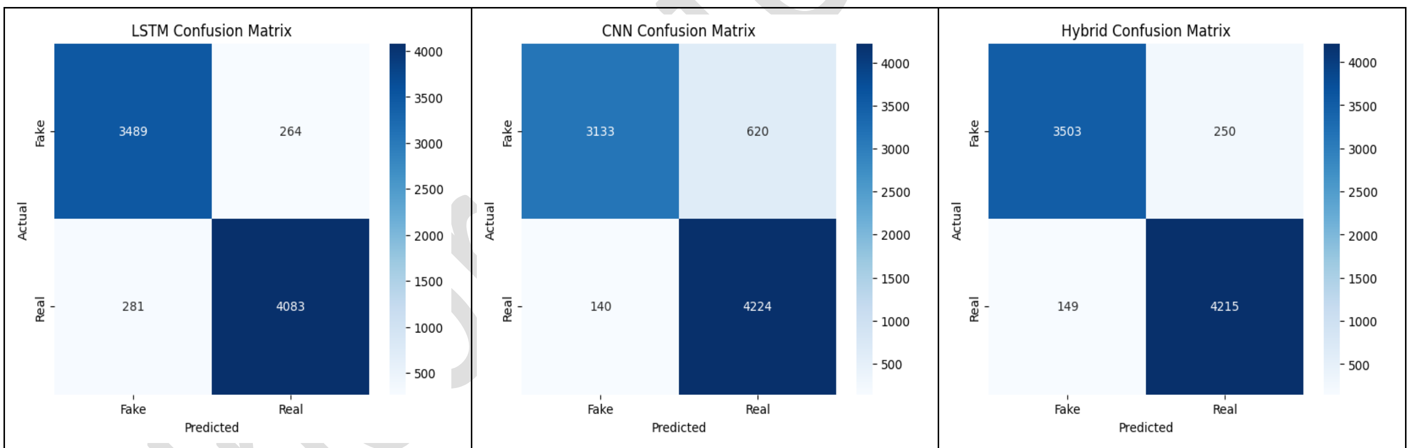
Where TP, FP, TN, and FN denote True Positive, False Positive, True Negative, and False Negative, respectively.

D. RESULTS AND DISCUSSION

Fig. 6 represents a confusion matrix with fake and real news using the LSTM in (a), CNN in (b), and LSTM-CNN in (c). As shown in Fig. 6(a), the results of the confusion matrix using the LSTM indicate that ٢٦٤ real articles are classified as fake, while ٢٨١ fabricated articles are classified as authentic. Additionally, we observe that ٣٤٨٩ fake articles are truly fake, while ٤٠٨٣ real articles are effectively legitimate.

According to the confusion matrix in Fig. ٦(b) generated by the CNN, ٦٢٠ real articles are classified as fake, while ١٤٠ fraudulent articles are classified as real. Additionally, we observe that 3133 fake articles are truly fake and 4224 real articles are effectively legitimate.

Also in Fig. 6(c), the results of the confusion matrix using the LSTM-CNN indicate that 250 real articles are classified as fake, while 149 fabricated articles are classified as authentic. Additionally, we observe that 3503 fake articles are truly fake, while 4215 real articles are effectively legitimate.



(a) (b) (c)

Fig. 6. Confusion matrix of fake and real news using (a) LSTM, (b) CNN, and (c) Hybrid LSTM-CNN

TABLE V. PERFORMANCE METRICS OF CNN, LSTM, AND HYBRID MODELS

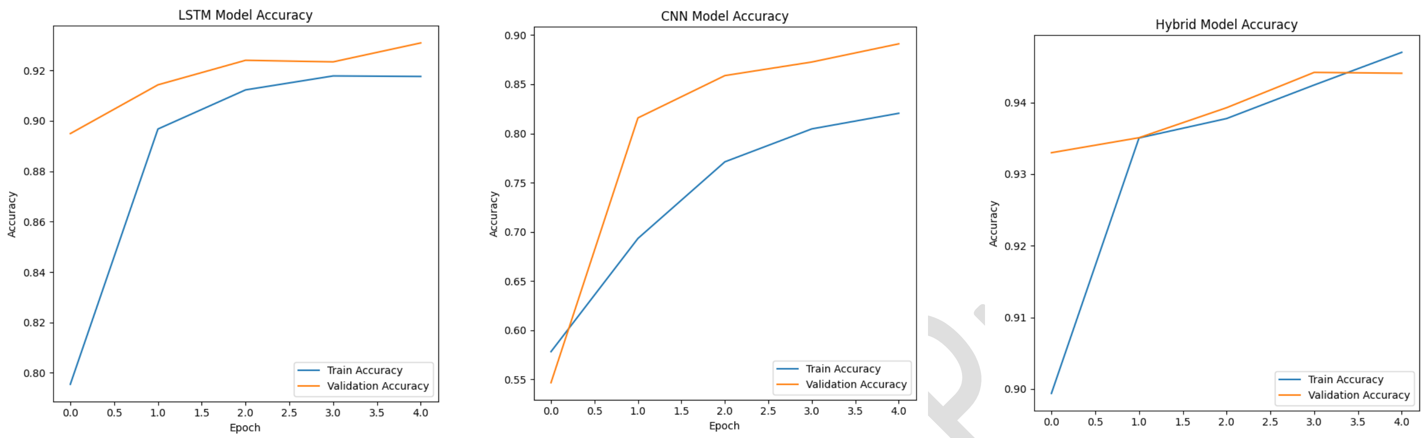
Metric	LSTM Model	CNN Model	Hybrid Model
Accuracy	93.3%	90.6%	95.1%
Precision (Fake News)	92.5%	95.7%	95.9%
Precision (Real News)	94.0%	87.2%	94.4%
Recall (Fake News)	93.0%	83.5%	93.3%
Recall (Real News)	93.5%	96.8%	96.5%

Table V presents, in general, that the hybrid model outperforms other models in most criteria. It has the highest accuracy, extremely good precision for both fake and real news, and outstanding recall for real news, with a considerable improvement in recall for fake news compared to the CNN model.

The LSTM model is more accurate than the CNN model and performs well across all criteria; the hybrid model exceeds it in overall accuracy and precision for fake news.

The CNN model is highly precise for fake news, but it falls below the other two models in terms of general accuracy and recall, particularly for actual news.

Fig. ٧ represents a model accuracy on training and validation data during training. The LSTM in (a), CNN in (b), and LSTM-CNN in (c). The graph shows how the model's accuracy changes on seen and unseen data as training progresses. During the subsequent epochs, the Hybrid model continuously performs better than the other two models in terms of validation accuracy. Although the LSTM model performs well at first, early stopping may be necessary to avoid overfitting. Despite demonstrating consistent progress, the CNN model had the lowest overall validation accuracy of the three during the observed training period, as also shown in Table VI.



(a) (b) (c)

Fig. 7. Model's Accuracy of fake and real news using (a) LSTM, (b) CNN, and (c) Hybrid LSTM-CNN

TABLE VI. COMPARISON OF MODEL PERFORMANCE BASED ON MODEL ACCURACY

Feature	LSTM	CNN	Hybrid
Initial Validation Accuracy	Relatively Higher	Relatively Low	Significantly Higher
Initial Learning Speed	Fast	Fast	Very Fast
Peak Validation Accuracy	Around 0.925	Around 0.89	Around 0.947
Signs of Overfitting	Potential in Later Stages	Less Pronounced Initially	Less Pronounced
Validation Stability Accuracy	Stable with Slight Fluctuation	Continuous Improvement	Stable at a High Level
Overall Performance Accuracy	Very Good	Lowest	Best

Table VI presents that the Hybrid model has the best overall performance, as seen by a strong start, rapid learning, the highest validation accuracy, and good stability with few indicators of overfitting. The LSTM Model likewise works well; however, it may be more prone to overfitting. The CNN Model improves steadily, but achieves the lowest validation accuracy compared to the other two models within the number of epochs shown.

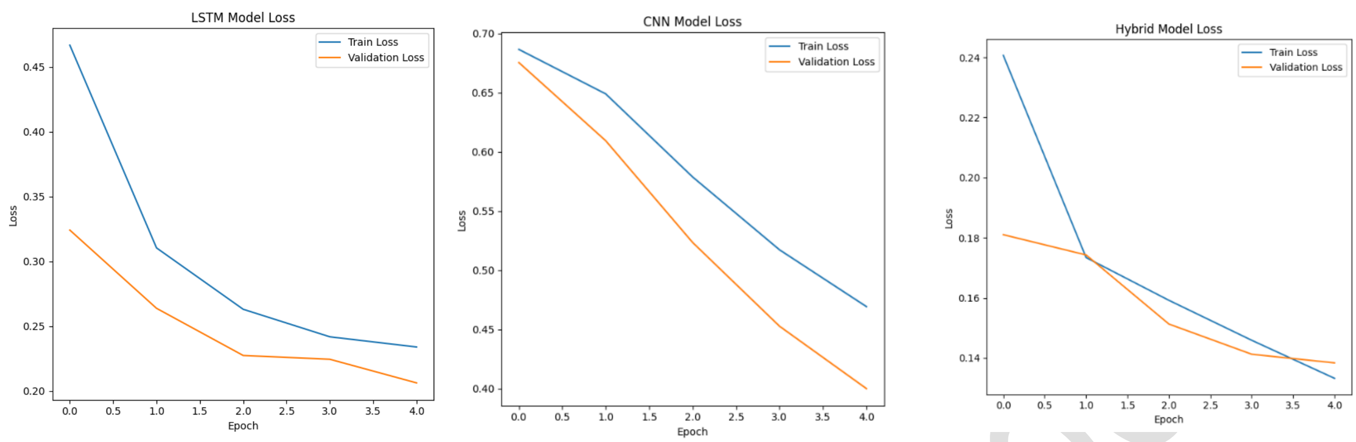
Fig. 8 represents a model loss on training and validation data during training. The LSTM in (a), CNN in (b), and LSTM-CNN in (c). The graph illustrates the training and testing loss across epochs. The decreasing loss curves reflect effective model learning, with the test loss closely following the training loss, suggesting good generalization with minimal overfitting, also shown in Table VII.

TABLE VII. COMPARISON OF MODEL PERFORMANCE BASED ON MODEL LOSS

Feature	LSTM Model	CNN Model	Hybrid Model
Initial Validation	Relatively Lower	Relatively High	Significantly Low
Validation Loss Trend	Decrease then Slight Increase	Consistent Decrease	Decrease then Slight Increase
Lowest Validation	Around 0.225	Around 0.40	Around 0.14
Signs of Overfitting	Potential after ~3 Epochs	Less Pronounced	Potential after ~3 Epochs
Learning Efficiency	Very Good	Good	Very High

Table VII presents that the Hybrid model has the lowest validation loss, implying that it is the most effective at decreasing prediction error on unknown data. However, it exhibits early signs of overfitting. The LSTM model produces a substantially smaller validation loss than the CNN model, but it also suggests potential overfitting. Within these epochs, the CNN model shows a continuous decrease in loss with no evident symptoms of overfitting, but it also has the largest validation loss of the three.

The suggested models were implemented on a cloud computing platform to guarantee scalability, flexibility, and effective resource use. Google Colab, as a Public Cloud that provides Software-as-a-Service (SaaS), was used to implement the research. There are many benefits to using Google Colab instead of local computers, as shown in Table VIII.



(a) (b) (c)

Fig. 8. Model's loss of fake and real news using (a) LSTM, (b) CNN, and (c) Hybrid LSTM-CNN

TABLE VIII. COMPARISON OF MODEL PERFORMANCE ON-DEVICE VS. CLOUD EXECUTION

Metric	LSTM Model	CNN Model	Hybrid Model
Validation Accuracy	93.29%	90.64%	95.00%
Training Time/Step (Cloud - T4 GPU)	15ms	7ms	10ms
Training Time/Step (Local - RTX 3060)	28ms	15ms	18ms
Training Time/Step (Local - CPU)	48ms	22ms	35ms
Inference Time/Sample (Cloud)	7ms	3ms	5ms
Inference Time/Sample (Local - GPU)	12ms	5ms	9ms
Inference Time/Sample (Local - CPU)	20ms	8ms	15ms
Hardware Efficiency	Needs cloud for real-time	Best for edge devices	Balanced for both

There are many benefits to using Google Colab instead of local computers, such as:

- Faster model training through the use of GPUs and TPUs.
- Avoiding local hardware limitations.
- No cost for any installation needed.

Because of these advantages, Google Colab was an appropriate and effective option for implementing and testing the suggested fake news detection models.

5. Comparison with Other Studies

The following table compares the performance of various machine learning and deep learning models that have been applied to the Fake News Classification dataset. The purpose of this comparison is to evaluate the effectiveness of the proposed hybrid LSTM-CNN model against other existing approaches based on the same dataset as indicated in Table IX.

TABLE IX. COMPARISON WITH OTHER STUDIES

Ref	Model Used	Acc. (%)	Key Features	Limitations
Ours	Hybrid LSTM-CNN	**95.08**	Combines CNN and LSTM; implemented on cloud	None
[27]	Random Forest	87.2	Simple, interpretable model	Limited contextual understanding
[28]	Neural Network with TF-IDF	88.6	TF-IDF + NN pipeline	No deep sequence learning
[16]	LSTM, RNN	90.1	Sequential modeling	Limited documentation
[١٣]	Naive Bayes, RF	81.0	Traditional ML approach	Low accuracy; lacks scalability
[29]	Recurrent CNN	92.3	Deep CNN with sequential input	No cloud integration
[30]	Logistic Regression + TF-IDF	85.4	Simple model, low complexity	High overfitting risk
[31]	CNN + LSTM	93.2	Hybrid DL model, limited generalization	Tested on the subset only

These findings show that the approach outperforms the majority of previous efforts in collecting both spatial and temporal characteristics. Although models are still computationally costly, they handle spatial and temporal patterns better than previous research.

6. Conclusion and future work

In order to detect fake news, this study suggested a hybrid deep learning model that combines the two models - Long Short-Term Memory (LSTM) networks with Convolutional Neural Networks (CNN). To take advantage of scalability and computational efficiency, the model was implemented in a cloud computing environment.

According to the experimental results, the hybrid model achieved a high performance of 95.08% and outperformed the results of running individually the CNN and LSTM models, in terms of accuracy, sensitivity, and specificity. This performance resulted from the successful extraction of both local features and long-term dependencies made possible by the combination of both models.

For future work, the model can be enhanced by integrating transformer-based architectures, such as BERT or RoBERTa, and optimizing deployment strategies for even lower latency and higher scalability. Also, it can be refined to include not only texts but also images that can also include fake information or be AI-generated.

References

1. M. Choudhary, S. Jha, Prashant, D. Saxena, and A. K. Singh, “A review of fake news detection methods using machine learning,” in 2021 2nd International Conference for Emerging Technology, INCET 2021, 2021. doi: 10.1109/INCET51464.2021.9456299.
2. X. Zhang and A. A. Ghorbani, “An overview of online fake news: characterization, detection, and discussion,” Information Processing and Management, vol. 57, no. 2, Mar. 2020, doi: 10.1016/j.ipm.2019.03.004.
3. T. Traylor, J. Straub, Gurmeet, and N. Snell, “Classifying Fake News Articles Using Natural Language Processing to Identify In-Article Attribution as a Supervised Learning Estimator,” in Proceedings - 13th IEEE International Conference on Semantic Computing, ICSC 2019, 2019. doi: 10.1109/ICOSC.2019.8665593.
4. N. Cavus, M. Goksu, and B. Oktekin, “Real‑time fake news detection in online social networks: FANDC cloud‑based system,” Scientific Reports, vol. 14, Art 25954, Oct. 2024, doi: 10.1038/s41598-024-76102-9.
5. Yang, C., Huang, Q., Li, Z., Liu, K., & Hu, F. (2016). Big Data and cloud computing: innovation opportunities and challenges. International Journal of Digital Earth, 10(1), 13–53. https://doi.org/10.1080/17538947.2016.1239771
6. Zhou, P., Li, Y., & Zhang, Y. Early detection of fake news on social media: An attention-based hybrid framework. Journal of Information Technology, 38(1), 44–61, 2023.
7. N. Aslam, I. Ullah Khan, F. S. Alotaibi, L. A. Aldaej, and A. K. Aldubaikil, “Fake detect: a deep learning ensemble model for fake news detection,” Complexity, vol. 2021, pp. 1–8, Apr. 2021, doi: 10.1155/2021/5557784.
8. Ghosh, S., & Kumar, R. CNN-based feature extraction for fake news classification. Applied Intelligence, 53(3), 4321–4336, 2023.
9. J. P. Singh, A. Kumar, N. P. Rana, and Y. K. Dwivedi, “Attention-based LSTM network for rumor veracity estimation of tweets,” Information Systems Frontiers, Aug. 2020, doi: 10.1007/s10796-020-10040-5.
10. Alhindi, T., Kalita, J., & Sajjad, H. "Detecting Fake News with Machine Learning Methods." Intelligent Systems Conference (IntelliSys), 1–6, 2020.
11. Ehzaam, A., Shankar, K., & Anand, S. "Fake News Detection Using Random Forest on Kaggle Dataset." 2021 International Conference on Advances in Computing and Communication Engineering (ICACCE), 76–80, 2021.
12. Ke, S., Yin, L., & Tan, J. "Neural Network Model for Fake News Detection Using TF-IDF." 2021 IEEE 10th Data Science and Engineering (ICDSE), 119–124, 2021.
13. Kesarwani, S., Sharma, R., & Verma, P. "Fake News Detection Using Naive Bayes and Random Forest." International Journal of Computer Applications, 184(2), 25–29, 2022.
14. Jain, R., Sharma, M., & Ghosh, P. "Comparative Study of CNN and LSTM in Fake News Detection." Procedia Computer Science, 199, 2332–2338, 2022.
15. Biradar, M. M., & Patil, A. D. "Fake News Detection Using Hybrid CNN-LSTM Model." International Conference on Computing, Communication, and Intelligent Systems (ICCCIS), 165–170, 2022.
16. Gupta, A. "Fake News Detection Using LSTM and RNN." GitHub Repository. https://github.com/AkshayGupta01/Fake-News-Detection, 2022.
17. Hossain, M. S., Rahman, M. A., & Ahmed, M. U. "A Deep Learning-Based Approach for Fake News Detection in Bangla." International Journal of Advanced Computer Science and Applications (IJACSA), 13(3), 123–130, 2022.
18. Wahab, A., & Fong, S. "Fake News Detection Using Ensemble Learning and RoBERTa Transformer." Journal of Information and Telecommunication, 7(1), 85–97, 2023.
19. Wang, L., & Li, J. "Fake News Detection Using Dual Attention-Based CNN-LSTM Model." IEEE Access, 11, 25241–25253, 2023.
20. Utku, D. "A Multichannel Deep Learning Approach Using CNN-LSTM for Fake News Detection." Journal of Information Security and Applications, 76, 103656, 2024.
21. Xu, Y., Zheng, L., & Zhao, H. "Attention-Based CNN-LSTM Model for Fake News Detection." Applied Sciences, 15(4), 923, 2025.
22. Hossain, M. A., Nizam, M. F., & Uddin, M. S. "Multichannel Combined CNN-LSTM Model for Fake

News Detection in Bangla Language." Neural Computing and Applications. https://doi.org/10.1007/s00521-025-08970-4, 2025.
23. Zamani, M., Hosseini, R., & Ahmadi, M. "Cloud-Based Deep Learning for Real-Time Fake News Detection." Future Generation Computer Systems, 152, 334–346, 2025.
24. M. Umer, Z. Imtiaz, S. Ullah, A. Mehmood, G. S. Choi, and B.-W. On "Fake news stance detection using deep learning architecture (CNN-LSTM)," IEEE Access, vol. 8, pp. 156695–156706, doi: 10.1109/ACCESS.2020.3019735, 2020.
25. The datasets used during the current study are available on Kaggle https://www.kaggle.com/datasets/bassamkasasbeh1/wsnds
26. E. Essa, K. Omar, and A. Alqahtani, "Fake News Detection Based on a Hybrid BERT and LightGBM Models," Complex & Intelligent Systems, vol. 9, no. 6, pp. 6581-6592, 2023.
27. Ehzaam, A., Al-Bahadili, H., & Abualkishik, A. M. Detecting Fake News Using Machine Learning Techniques. Journal of Information Security Research, 12(3), 45–52, 2021.
28. Ke, X., Zhang, H., & Liu, Y. "Fake News Detection Using TF-IDF and Neural Networks". Proceedings of the 5th International Conference on Computer and Communication Systems, 112–118, 2021.
29. Mavuduru, R. "Fake News Detection using Recurrent Convolutional Neural Networks". GitHub Repository. Retrieved from: https://github.com/rohithmavuduru/Fake-News-Detection, 2023.
30. Davis, B. "Fake News Classification with Logistic Regression and TF-IDF". GitHub Repository. Retrieved from: https://github.com/bdavis9/fake-news, 2023.
31. Zhou, X., Zafarani, R., Shu, K., & Liu, H. "Fake News Detection via NLP is Vulnerable to Adversarial Attacks". Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, 1805–1808, 2020.