A Context-Enhanced Model for Fake News Detection

Majdi Beseiso
Department of Computer Science, Al-Balqa Applied University, Al-Salt, Jordan
bsaiso@bau.edu.jo (corresponding author)

Saleh Al-Zahrani
Information Systems Department, Al-Imam Mohammad Ibn Saud Islamic University, Riyadh, Saudi Arabia
drsalehz@hotmail.com

Received: 7 October 2024 | Revised: 12 November 2024 | Accepted: 16 November 2024

 DOI: https://doi.org/10.48084/etasr.9192

ABSTRACT

News published on social networks has a notable impact on changing people's perceptions on various topics. However, all news available on social media may not be genuine and might come from unverified sources. The prevalence of fake news is an inevitable concern that needs to be addressed effectively. This study presents an ensemble algorithm to improve fake news detection tools. Long-Short-Term-Memory (LSTM) and an ensemble of LSTM and Convolutional Neural Networks (CNN) were used. The proposed model used bidirectional LSTM layers and CNN convolutional 2D layers with kernel sizes of 2, 3, and 4 for 2-gram, 3-gram, and 4-gram tokens. The results obtained show an accuracy of 96.7% and 97.3% on a fake news dataset using the LSTM model and CNN-LSTM model, respectively, significantly improved from the maximum accuracy of 94.88% reported in a previous study. Embedding layers yielded significant improvements when paired with extended word sequences and pre-trained embedding vectors. Diverse tokenization methods with and without pre-trained embedding layers were also considered. The ensemble model achieved a 10.03% improvement in predictive accuracy on the Liar dataset, compared to the 6.08% improvement reported in a previous study using the same dataset.

Keywords-convolutional neural networks; fake news; word2vec embedding; natural language processing; long short-term memory

I. INTRODUCTION

The rise of social networks has modernized the way people access news and information. Lately, applications such as Twitter and Facebook have not only remained a platform for socialization but also become a source of global news access for most of their users [1]. However, this widespread and rapid dissemination of information comes with the risk of quickly spreading false information. Several users read the news on social media and start believing it even without checking its credibility [2]. The news shared or read on social networks multiple times does not always guarantee its accuracy. Fake news, in particular, is used to manipulate public opinions, distort perceptions, and generate social unrest. Hence, users are encouraged to be more cautious about what they read to avoid getting manipulated easily. Nevertheless, humans, on average, can spot lies with only 54% accuracy [3]. Consequently, using Artificial Intelligence (AI) is believed to be a more reliable approach to precisely identify fake news [4].

Given the rapid dissemination of information on social networks, the challenge of identifying and mitigating fake news has become increasingly critical. Despite advances in fake news detection, existing algorithms face significant challenges, including limited accuracy, dataset bias, and the need for more sophisticated feature representation [5]. For instance, biases in the datasets used for training these models can lead to skewed predictions, whereas the models' reliance on simple features may overlook more complex patterns indicative of fake news.

Extensive research on the detection of fake news is imperative for providing essential tools to social networks or users to evaluate the legitimacy of the news published. A combined Natural Language Processing (NLP) and Machine Learning (ML)-based approach is a potential solution to detect fake news [6]. NLP can facilitate the computer in reading and decoding human language and extracting useful information from the text. ML can help detect and predict outcomes. Lately, Deep Learning (DL) has also gained significant research attention for multimodal fake news detection systems [7]. Some recent studies have explored innovative methods to detect fake news. In [8], a Kaggle dataset [9] was used to train classification models relying on a 1-gram approach that disregards previous tokens, a choice that may impact the model's effectiveness. Although an accuracy of 83.1% was reported for a Naive Bayes (NB) classifier with Lidstone smoothing, no additional performance metrics were provided. Consequently, it is challenging to assess the model's true effectiveness, as accuracy alone does not account for potential biases in the dataset or imbalances between real and fake news classes. These omissions leave uncertainty about the model's robustness and its capacity to generalize beyond the specific dataset. In [10], fake news detection was performed using n-grams of characters and words. This approach involved preprocessing and gradient boosting, achieving a maximum accuracy of 96% using character 3-grams and 4-grams with the term frequency-inverse document frequency (tf-idf) weighting. In [11], fake news detection was investigated using various feature extraction approaches and models. This study compared the tf, tf-idf, and Word2Vec feature extraction methods, where tf performed the best and word2vec showed the worst performance. In addition, the performance of classifiers and different feature extraction methods was compared using a 3-fold cross-validation. The tf features with the LSTM model showed an accuracy of 94.88% in a 3-fold cross-validation. All these studies, including [8], [10], and [11], report only accuracy as a performance metric, which limits their reliability for fake news detection.

In [12], traditional and DL/ML models were compared on two fake news datasets. One of the datasets used in this study was the same as in [8, 9] and the other as in [11]. This study implemented different classification models and LSTM as a DL model. An accuracy of 97.1% was achieved using the LSTM model with 1100 sequences of words. The study in [13] aimed to improve fake news detection accuracy on the Liar dataset, originally analyzed in [14]. By simplifying the multiclass problem into a binary classification (fake vs. real), a significant improvement was achieved over the results of [14]. Various traditional and DL models were compared, finding that a Convolutional Neural Network (CNN) and a Recurrent Neural Network (RNN) were the most effective, with a maximum accuracy of 59.82%. This level of performance indicates that further improvements are necessary. In [15], a new end-to-end design was proposed to detect fake news and mitigate its spread. This approach combined DL architectures, namely convolutional and bidirectional LSTM (BiLSTM), with a network-aware, real-time approach to identify fake news and immunize nodes spreading it. The efficacy of this solution was demonstrated on five real-world datasets, highlighting its practical applicability and innovative approach against misinformation. In [16], Arabic fake news classification was improved by addressing data imbalance across ML models. However, this study primarily used traditional algorithms, limiting its exploration of DL methods that could further enhance accuracy.

In [17], a hybrid model combining LSTM and Bidirectional Encoder Representations from Transformers (BERT) was presented for fake news classification. This model exhibited higher accuracy than the vanilla BERT model on the FakeNews dataset, demonstrating the effectiveness of the combined approach. Although this hybrid method holds significant promise, it is not without limitations. One of the key challenges lies in the potential for overfitting, particularly with the highly complex BERT model, which requires large, high-quality datasets for optimal performance. In [18], BiLSTM architectures with different sentence transformers were compared in detecting fake news in English and German. This study highlighted the use of sentence transformers and advanced BiLSTM architectures to effectively tackle the detection of fake news in several languages. This model performed better in detecting fake and true news but struggled with the partially false and "other" categories, highlighting the difficulty in distinguishing between certain and raising questions about its generalization. Additionally, in [19], a new method using document embeddings was proposed and evaluated on five large news corpora to efficiently detect fake news. The results showed that document encoding is more important than classification model complexity to obtain high accuracy.

In contrast, this study employs a novel approach by integrating multiple advanced techniques for fake news detection. Two DL models, CNN and LSTM, were used, combined with an ensemble learning approach and extended feature extraction methods. This study builds on the dataset used in [11] and introduces the dataset proposed in [14]. Advanced preprocessing and tokenization techniques were used, including pre-trained models, e.g. BERT [20], DistilBERT [21], RoBERTa [22], and XLM-RoBERTa [23]. A notable contribution of this study is the integration of n-gram features (2-grams, 3-grams, and 4-grams) into the CNN model. This approach, with CNN utilizing kernel sizes of 2, 3, and 4, allows for extracting diverse features from consecutive word sequences. These methods were tested using binary classes and compared to previous studies. Furthermore, a refined ensemble learning method is proposed and compared with the findings of previous studies. The FakeBERT model [24] complements this research by demonstrating the efficacy of combining BERT with CNN to handle text ambiguity, achieving an accuracy of 98.90%. This approach aligns with the use of advanced embeddings and bidirectional context to improve fake news detection.

II. METHOD

In the preprocessing step, the dataset used for classification was cleaned by erasing digits and punctuations, removing stop words and URLs, and converting all characters into lowercase. Moreover, punctuation marks, numbers, and URLs were removed from the news text to ensure a focused analysis. After data cleaning and NLP preprocessing, the tokenized text was converted into numerical representations using word embeddings. This approach captures semantic relationships between words, allowing for a more nuanced representation of the text. This results in a feature-rich matrix that encodes contextual meaning, making it well-suited for input into classification models.

Other techniques used for converting text to numerical representation include the Bag of Words (BoW), where each unique word in the text corpus is treated as a feature, and text documents are represented by the frequency or occurrence of each word in a sparse matrix format. A more sophisticated feature extraction method, tf-idf, normalizes word frequencies by their document frequencies to reduce the impact of common but less important words. This is particularly useful for news articles where common words might not be as informative.

Word2Vec is another feature extraction technique that represents words as dense vectors, capturing semantic relationships through word co-occurrences. Cosine similarity measures the similarity between words in these embeddings. Although theoretically unlimited, practical constraints require fixed-size representations [25]. Neural networks learn these embeddings, reducing dimensionality and improving efficiency. Unlike Word2Vec's sparse matrices, embedding layers use fixed-size vectors. Pretrained models such as BERT, DistilBERT, RoBERTa, and XLM-RoBERTa offer pre-trained embedding weights, accelerating training and providing valuable semantic knowledge.

An embedding layer was chosen over techniques such as Word2Vec, BoW, and tf-idf because it allows the model to learn task-specific, dense vector representations directly within the network. Unlike others that produce static embeddings, the embedding layers dynamically adapt during training, optimizing word representations specifically for the classification task. This approach provides richer contextual information and improves model performance.

This study used two DL models, LSTM and CNN-LSTM, and a tuned ensemble learning method with multiple preprocessing and tokenization approaches. Specifically, 3-gram and 4-gram features were used alongside traditional n-grams to capture a broader context and more nuanced patterns in the text data that are not captured by 1-gram or 2-gram features. This approach enhances the capability of the model to detect subtle differences between fake and real news. Figure 1 illustrates the workflow for the analysis implemented.

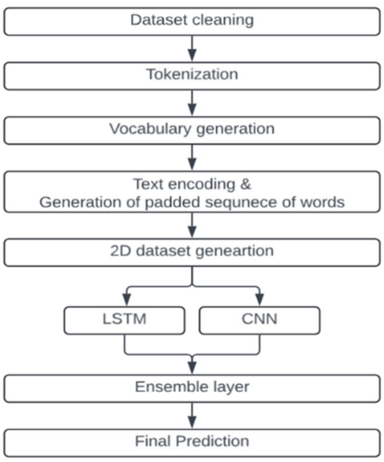

Fig. 1. Flowchart of the method.

A. Dataset Description

The dataset was taken from [26]. Data columns include title, author, text, and label (1 implies fake news, and 0 denotes real news). The dataset has some missing values for the title, author, and body of the news. Table I shows 20,203 news in the dataset, after removing news with missing values, of which 48.5% are fake and 51.5% are real. The proportion of both classes is approximately 50%, which implies that the chosen dataset is balanced.

TABLE I. REAL AND FAKE NEWS IN SELECTED DATASET

Description	# of Fake news	# of Real news
Total news	10387 (49.9%)	10413 (50.1%)
Missing title	558	0
Missing text	39	0
Non-missing news	9790 (48.5%)	10413 (51.5%)

The Liar dataset, sourced from [14], is a benchmark for fake news detection that contains several columns including label, context, mostly true counts, barely true counts, half true counts, pants on fire counts, false counts, party affiliation, state, speaker, speaker's job title, subject, and statement. The statement column provides the news claim to be classified, whereas the other columns offer additional context and metadata about the speaker and the statement. The label column classifies the statement as pants on fire, true, barely true, mostly true, half true, or false. After removing entries with missing values, the Liar dataset has 12,835 statements, including False (24.8%), Barely True (22.1%), Half True (19.4%), Mostly True (16.7%), True (10.9%), and Pants on Fire (6.1%). The distribution of these classes indicates a diverse range of truthfulness, making the dataset slightly imbalanced but providing rich contextual data for analyzing and predicting the veracity of statements.

B. LSTM Model

The LSTM model is a DL architecture that addresses the limitations of traditional n-gram models by considering the entire sequence of text using feedback connections to determine the state of a memory cell. It has input, output, and forget gates, and employs sigmoid and hyperbolic tangent activation functions [27]. The LSTM architecture uses a maximum of 160K words, a sequence length of 1000, and an embedding dimension of 256.

This study employed two bidirectional LSTM layers with 128 and 256 units. One of the layers retains the sequence information, while the other does not. BiLSTMs process sequences in both directions, doubling the output size compared to unidirectional LSTMs. Therefore, the output of a 256-unit BiLSTM layer for a 1000-sequence input is 1000×512, with 256 units for forward and backward sequences. After two LSTM layers, the output is flattened to be in one dimension. Subsequently, a dense layer with 32 neurons and a Rectified Linear activation function (ReLU) was used. The ReLU function keeps the positive outcomes and sets the negative outcomes to zero. Then, the last layer is the output, including 2 neurons. The values of the last layer are eventually mapped to be in the range [0, 1] using the sigmoid activation function. For recurrent layers, 30% dropout was considered, and for other layers, 10% dropout was used to avoid overfitting issues. The Adam optimizer was used with a learning rate of 0.001, whereas categorical cross-entropy was employed as a loss function. Each batch of data includes 32 news and the data are trained for 20 epochs.

C. CNN-LSTM Model

This study also used a CNN-LSTM [28] model that combines the LSTM structure with a CNN. The CNN generates n-gram tokens using kernels of varying sizes, considering both the embedding direction and the word sequence. The LSTM then processes the extracted features. The input data are embedded with a dimension of 256 and a max sequence length of 300 tokens. A kernel with a size of 3×256 is used to extract features from consecutive tokens in the sequence. The 2D convolutional layer requires a 3D input, so the sequence is expanded to 300×256×1. Using 10 filters, the output becomes 298×1×10, because the centroid of the kernels cannot be in the first or last token in the sequence, extracting 10 features. The filter values are learned during training, extracting features based on the size of the kernel and the number of filters.

In the CNN model, 3 convolutional 2D layers with kernel sizes of 2×256, 3×256, and 4×256 are used for training to obtain 2, 3, and 4 grams of tokens. Then, max-pooling layers are applied to reduce the dimensionality of the output. The concatenated output from the CNN and LSTM layers is then fed into a fully connected layer with 32 neurons, followed by a final layer with sigmoid activation and 2 neurons to predict whether the news is fake or real.

D. Ensemble Learning with Multiple-Tokenization

An ensemble learning approach was developed to classify fake and real news. This approach incorporates models trained on various preprocessing and tokenization techniques utilizing BERT, DistilBERT, RoBERTa, and XLM-RoBERTa. Two embedding strategies were employed: a normal embedding layer with a dimension of 256 and a maximum sequence length of 512, and pre-trained embedding layers from the backbone models. LSTM and CNN models were employed to process the embedding layer outputs. These model predictions were stacked and combined using their mode to enhance classification accuracy. In this approach, the hyperparameters of the LSTM-CNN model were tuned using grid search with for loops on the grids. For hyperparameter tuning, the Tesla T4 GPU with 16GB RAM was used in Google Colab. Algorithm 1 describes the proposed ensemble learning algorithm.

```
Algorithm 1 Proposed ensemble learning
algorithm
Input: {neurons, mdl, pretrained,
  Threshold}
Final_Prediction = empty list
Output: Final_Prediction.
Method:
forEach neurons do
  Prediction = empty list
  forEach mdl do
    If pretrained then
      Preprocessing and tokenizing with
      mdl
      M Model Fit with pretrained backbone
      mdl
    else
      tokenizing with mdl
      M Model Fit with normal embedding
      layer
    end if
    Pred = M.predict(test data) >
    Threshold
    Prediction append Pred
  end for
  Final_Prediction append Mode Prediction
end for
```

III. RESULTS AND DISCUSSION

The dataset was built by tokenizing the news documents and creating a sequence of words by padding the sequences to have equal shapes. The dataset has a size of 20,203 and a 3-fold cross-validation was employed. The categorical cross-entropy was employed as a loss function and the Adam optimizer was used with a 0.001 learning rate. Moreover, the accuracy metric was considered for model fitting.

The model was compiled for 100 epochs with a 32 batch size and early stopping, normalization, and dropout were employed to handle underfitting and overfitting issues. Validation accuracy was found to be 96.7% and 97.3% for the LSTM model and the CNN-LSTM model, respectively. In [11], an accuracy of 94.88% was achieved using an LSTM model on the same dataset with a 3-fold cross-validation. This implies that the proposed model has surpassed the results previously obtained. This study indicates that increasing the number of word sequences and fine-tuning parameters can enhance performance compared to prior methods. Moreover, although slightly higher accuracy (97.1%) was achieved in [12], it was based on a smaller dataset (3000 observations) compared to this study (6735 observations), demonstrating the robustness of this study's results.

Table II compares the results of this study and important relevant studies [11, 12]. The study in [11] favored count vectorizer whereas in [12], tf-idf was preferred as the better feature extraction method. This study showed that the word2vec embedding or embedding layer can perform as efficiently as the other feature extraction approaches by considering a proper sequence and a large enough vocabulary when computing the embedding layer. In addition, the Area Under the Curve (AUC) was 0.9970 for CNN-LSTM compared to 0.9960 in [12] for LSTM and 0.99 in [11] (which showed only two decimal points).

TABLE II. COMPARISON WITH THE PREVIOUS STUDIES

Model	[12]	[11]	This study
LSTM	97.1%	94.88%	96.7%
CNN-LSTM	not done	not done	97.3%
Information			
Support	3000	not mentioned	6735
Best features	tf-idf	countvector	Embedding
AUC	0.9960	0.99	0.9970

It was noticed that after 20 epochs, the model exhibited signs of convergence in both training and validation, thus, after 5 consecutive epochs with converged values, the model was early stopped. The model ran on a PC with a Core i7 CPU, 8 threads, and 16 GB RAM. The DL model was coded using the Python package of TensorFlow version 2. Table III shows the test results.

TABLE III. CONFUSION MATRIX - FAKE NEWS DATASET

Class label		Prediction	
		Real	**Fake**
Actual	Real	3344	119
	Fake	66	3206

From 6735 news in test data, 6550 (97.3%) were correctly predicted. Moreover, 3344 (96.6%) out of 3463 real news and 3206 (98%) out of 3272 fake news were correctly predicted. Table IV presents the classification report with accuracy measures of F1-score, recall, and precision in decimal points between 0 to 1, where 1 means 100%.

TABLE IV. CLASSIFICATION RESULTS FOR THE FAKENEWS DATASET

Label	Precision	Recall	F1-score	Support
Fake	0.96	0.98	0.97	3272
Real	0.98	0.97	0.97	3463
Accuracy			0.97	6735

Table V shows the confusion matrix for the Liar dataset, simplified to binary labels True and False, revealing a total of 4,000 observations. The model correctly identified 1,200 out of 1,380 actual True statements and 2,420 out of 2,620 actual False statements, indicating a robust performance. Table VI shows the classification results for the Liar dataset.

TABLE V. CONFUSION MATRIX FOR THE LIAR DATASET

Class label		Prediction	
		Real	**Fake**
Actual	Real	1200	180
	Fake	200	2420

TABLE VI. CLASSIFICATION RESULTS FOR THE LIAR DATASET

Label	Precision	Recall	F1-score	Support
Fake	0.93	0.92	0.93	2620
Real	0.86	0.87	0.87	1380
Accuracy			0.91	4000

These results show that the proposed DL model offered a promising performance in differentiating between fake and real news. The AUC-ROC (Figure 3) for the LSTM and CNN-LSTM models was 0.9943 and 0.9970 respectively, showing that CNN-LSTM was slightly better than the LSTM model.

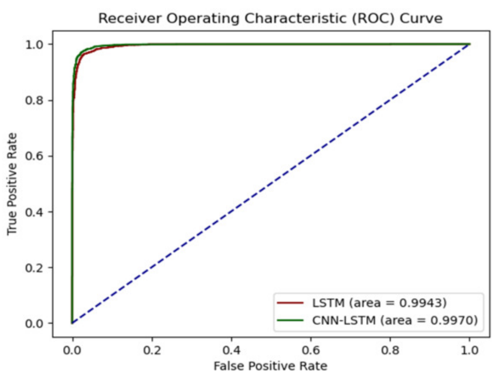

Fig. 2. ROC curve for LSTM and CNN-LSTM model with reported AUC.

The Liar dataset was then used and two CNN and LSTM models were fitted on it after making the labels binary (fake and real). The parameters of the number of neurons used in the models were tuned by grid search considering the validation accuracy as the evaluation metric. The batch size, number of epochs, dropout rate, and number of filters in the LSTM layer were tuned. The selected batch size was 32, the dropout rate was 0.1, and the total epochs were 20. The number of filters at 128 showed a better performance. The model was trained without pre-trained weights, and only the tokenization of BERT, DistilBER, RoBERTa, and XLM-RoBERTa was used in the first step.

In this instance, removing the LSTM component from the CNN-LSTM model significantly altered the ROC curve and its AUC, indicating a decrease in fake news classification accuracy, as shown in Figure 4. This analysis highlights the significance of the LSTM layer in improving the model's capability to differentiate between fake and real news.

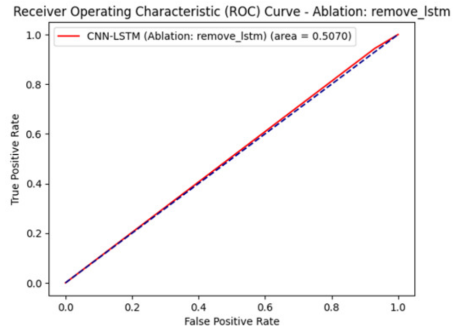

Fig. 3. ROC curve with the removal of LSTM.

The CNN layer is crucial for extracting spatial features from the input data. Removing the CNN layer from the CNN-LSTM model during ablation testing significantly reduced the model's ability to identify patterns and features, leading to a decline in accuracy and performance in classifying fake news, as shown in Figure 5. The LSTM component, without pre-extracted spatial features, struggled to process raw sequential data effectively. This ablation test highlighted the importance of the CNN layer in providing critical feature maps to enhance the model's overall effectiveness in differentiating between fake and real news.

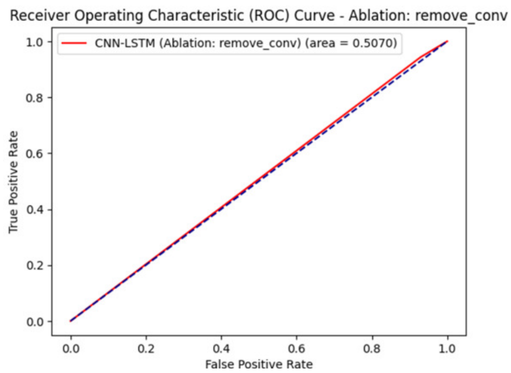

Fig. 4. ROC curve with the removal of CNN.

Hyperparameter tuning was performed for both CNN and LSTM models to optimize fake news detection. Key parameters such as learning rate, batch size, dropout rate, filters, LSTM units, kernel size, and pooling were adjusted through grid search. The best settings were 128 units and 0.1 dropout for the LSTM, and 256 filters with a kernel size of 3 for the CNN (Table VII).

TABLE VII. TUNING RESULTS FOR THE CNN-LSTM MODEL

Hyperparameter	Values tested	Best value	Validation accuracy	Test
Learning rate	0.0001, 0.001, 0.01	0.001	97.0	97.3
Batch size	16,32,64	96.8	97.3	
Dropout rate	0.1,0.3,0.5	96.9	97.3	
Number of CNN	128,256,512	256	97.1	97.3
Kernel size	2,3,5	97.3		
Number of LSTM	64,128,256	128	97.2	97.3

Hyperparameter tuning improved accuracy, reduced overfitting, and enhanced the generalization capability on the test data. Table VIII presents the test results for 64, 128, 192, and 256 filters.

TABLE VIII. LSTM RESULTS WITH VARIOUS TOKENIZATION

Filters	64	128	192	256
BRT	62.21%	62.63%	62.32%	61.12%
Distil	61.15%	**63.03%**	61.92%	62.24%
RoBERTa	62.24%	61.84%	59.46%	58.42%
xlmroberta	62.16%	60.25%	60.33%	60.81%

Note. These results are without pre-trained weights.

The model with 128 filters achieved the highest testing accuracy when using the Distil tokenizer. This model was then further tested with various pre-trained weights and backbone models (BERT, Distil, RoBERTa, and XLM-RoBERTa) using LSTM, and Table IX shows the results. The tuned threshold for the prediction was 0.5, 0.57, and 0.39 for RoBERTa, BERT, and Distil respectively. The xlmroberta was dropped from the ensemble model since its optimized threshold led to a prediction that was not better than the null or random model. The ensemble results of the LSTM model were better than the separate LSTM models.

TABLE IX. LSTM RESULTS WITH PRETRAINED WEIGHTS

Filter	BERT	Distil	RoBERTa	xlmrob
128	61.28%	63.51%	63.99%	60.33%
Ensemble		65.90%		

The Liar dataset was also checked by the CNN model. The number of filters used in the three CNN layers, which had kernel sizes of 2, 3, and 4, respectively, were 64, 128, 192, and 256. The number of epochs was 15, the batch size was 32, and the dropout rate was 0.5. Table X presents the results of the various tokenizers without pre-trained weights.

TABLE X. CNN RESULTS WITH VARIOUS TOKENIZATION

Filters	64	128	192	256
BERT	59.46%	60.10%	60.25%	58.03%
Distil	58.90%	60.49%	56.77%	**61.05%**
RoBERTa	58.28%	57.87%	58.90%	59.70%
xlmrob	58.50%	60.10%	58.11%	59.46%

Note. This result is without pre-trained weights.

The model with a filter size of 256 had the highest accuracy. The CNN models were run again using the pre-trained weights. The selected prediction threshold was 0.53, 0.48, 0.61, and 0.46 for the models with BERT, Distil, RoBERTa, and XLM-RoBERTa backbone pre-trained layers, respectively, and Table XI shows the results.

TABLE XI. CNN RESULTS WITH PRETRAINED WEIGHTS.

Filter	BERT	Distil	RoBERTa	xlmrob
128	63.59%	62.71%	63.67%	58.43%
256	63.04%	63.43%	63.20%	57.63%
Ensemble				66.38%

The performance of the proposed model was also compared to previous studies [13, 14] on the Liar dataset. In [14], multiclass labels were employed, whereas in [13], the focus was on binary classification. Table XII summarizes the results of these three studies for fake news detection.

TABLE XII. COMPARISON WITH PREVIOUS STUDIES.

Models	Accuracy	Balanced improvement compared to null results
[14]	27%	6.08%
[13]	59.82%	3.47%
This study	**66.38%**	**10.03%**

The test data for the multiclass label include 265 observations in the half-true class. The null model, which predicts all observations in one class, achieved an accuracy of 20.92%. In [14], for text only, a 6.08% improvement was achieved compared to the null model. In [13], for binary class labels with 714 observations in the real class (56.35% of the total data), a 3.47% improvement was shown over the null model. However, this study achieved an improvement of 10.03%, surpassing the previous two studies.

Figure 6 shows the precision-recall curves for the CNN ensemble model with changes in threshold values for BERT, RoBERTa, DistilBERT, and XLM-BERT. The precision-recall curves of each individual model are also shown for both classes of real and fake news. Figure 6 shows that the ensemble model performed consistently better than the individual models (CNN-XLM-Roberta, DistilBERT, BERT, and RoBERTa) in both fake and real news classification, especially for recall values above 0.5. Among individual models, CNN-XLM-Roberta performed the worst, followed by DistilBERT, BERT, and RoBERTa.

The performance of the model could be enhanced by using a computer with more memory and more GPUs. However, due to hardware limitations, the model could not analyze large text sequences, leading to potential performance issues. Even when using Google Colab GPU, the limited runtime duration made it difficult to run the entire process automatically, requiring manual step-by-step execution.

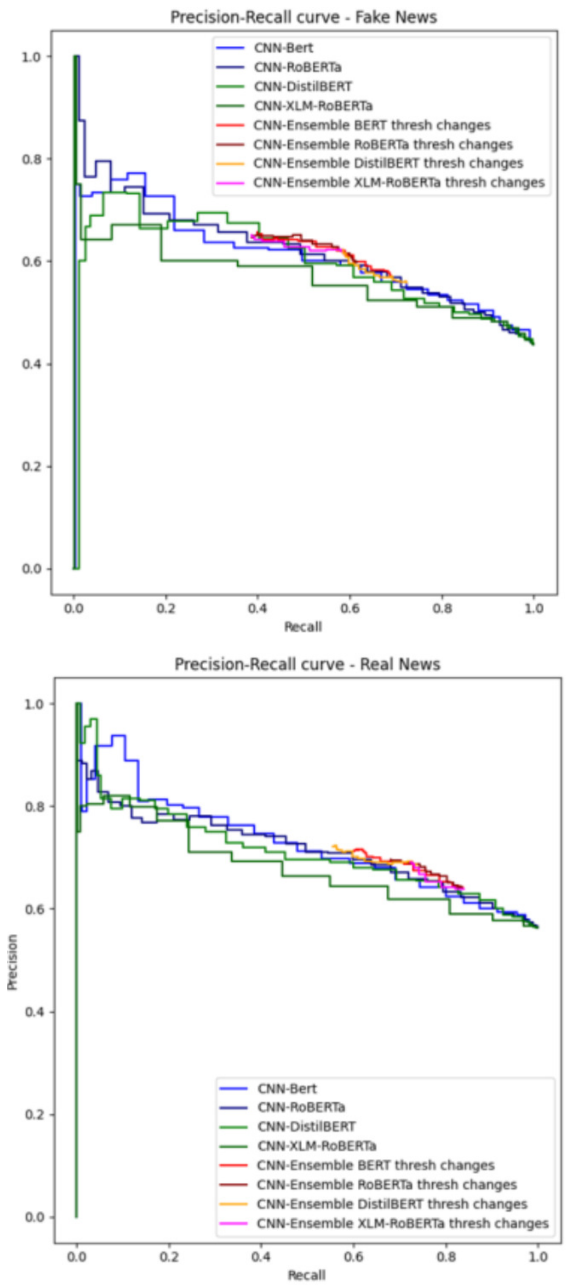

Fig. 5. Precision-recall for fake and real news.

IV. CONCLUSION

This study introduced an advanced deep learning framework that synergizes CNNs and LSTM networks to push the boundaries of fake news detection accuracy and robustness. By leveraging CNNs to capture important n-gram features and LSTMs to effectively model sequential dependencies, the proposed model achieved an impressive 97.3% accuracy, marking a significant advancement over existing techniques in the field. This approach not only sets a new benchmark but also underscores the potential of integrated architectures to address the challenges of misinformation detection. The proposed ensemble algorithm, which combines multiple tokenization techniques and pre-trained backbone models, demonstrated superior performance, achieving a 66.38% accuracy on the Liar dataset and exceeding benchmarks set by previous studies. This breakthrough showcases the potential for high adaptability and scalability in real-world misinformation detection tasks.

Despite these advances, it is important to recognize that the misinformation landscape remains complex and has aspects beyond the immediate scope of this study. Future research could build on the proposed ensemble model by exploring strategies that could deepen our understanding of fake news propagation and containment. Additionally, refining the architecture with larger sequence lengths, optimized filters, and various n-grams, along with using more sophisticated vectorization, could further enhance performance. By leveraging high-performance GPUs, these improvements could enable even greater system efficiency and accuracy, pushing the model closer to practical, real-world applications against misinformation.

ACKNOWLEDGMENT

The research reported in this publication was supported by the Deanship of Scientific Research and Innovation at Al-Balqa Applied University in Jordan.

REFERENCES

[1] R. K. Nielsen, "News media, search engines and social networking SITES as varieties of online gatekeepers," in *Rethinking Journalism Again*, Routledge, 2016.

[2] M. Faraon, A. Jaff, L. P. Nepomuceno, and V. Villavicencio, "Fake News and Aggregated Credibility: Conceptualizing a Co-Creative Medium for Evaluation of Sources Online," *International Journal of Ambient Computing and Intelligence (IJACI)*, vol. 11, no. 4, pp. 93–117, Oct. 2020, https://doi.org/10.4018/IJACI.20201001.oa1.

[3] E. Cueva, G. Ee, A. Iyer, A. Pereira, A. Roseman, and D. Martinez, "Detecting Fake News on Twitter Using Machine Learning Models," in *2020 IEEE MIT Undergraduate Research Technology Conference (URTC)*, Cambridge, MA, USA, Oct. 2020, pp. 1–5, https://doi.org/10.1109/URTC51696.2020.9668872.

[4] F. A. Ozbay and B. Alatas, "Fake news detection within online social media using supervised artificial intelligence algorithms," *Physica A: Statistical Mechanics and its Applications*, vol. 540, Feb. 2020, Art. no. 123174, https://doi.org/10.1016/j.physa.2019.123174.

[5] M. Nirav Shah and A. Ganatra, "A systematic literature review and existing challenges toward fake news detection models," *Social Network Analysis and Mining*, vol. 12, no. 1, Nov. 2022, Art. no. 168, https://doi.org/10.1007/s13278-022-00995-5.

[6] V. Kumar, A. Kumar, A. K. Singh, and A. Pachauri, "Fake News Detection using Machine Learning and Natural Language Processing," in *2021 International Conference on Technological Advancements and Innovations (ICTAI)*, Tashkent, Uzbekistan, Nov. 2021, pp. 547–552, https://doi.org/10.1109/ICTAI53825.2021.9673378.

[7] I. Q. Abduljaleel and I. H. Ali, "Deep Learning and Fusion Mechanism-based Multimodal Fake News Detection Methodologies: A Review," *Engineering, Technology & Applied Science Research*, vol. 14, no. 4, pp. 15665–15675, Aug. 2024, https://doi.org/10.48084/etasr.7907.

[8] K. Agarwalla, S. Nandan, V. A. Nair, and D. D. Hema, "Fake News Detection using Machine Learning and Natural Language Processing," *International Journal of Recent Technology and Engineering*, vol. 7, no. 6, pp. 844-847, 2019.

[9] "Fake News detection." https://www.kaggle.com/datasets/jruvika/fake-news-detection.

[10] H. E. Wynne and Z. Z. Wint, "Content Based Fake News Detection Using N-Gram Models," in *Proceedings of the 21st International*

Conference on Information Integration and Web-based Applications & Services, Munich, Germany, Dec. 2019, pp. 669–673, https://doi.org/10.1145/3366030.3366116.

[11] S. Vijayaraghavan *et al.*, "Fake News Detection with Different Models." arXiv, Feb. 15, 2020, https://doi.org/10.48550/arXiv.2003.04978.

[12] A. Shah, "Distinguishing fake and real news of twitter data with the help of machine learning techniques," M.S. Thesis, Laurential University, Sudbury, Canada, 2021.

[13] L. Singh, "Fake News Detection: a comparison between available Deep Learning techniques in vector space," in *2020 IEEE 4th Conference on Information & Communication Technology (CICT)*, Chennai, India, Dec. 2020, pp. 1–4, https://doi.org/10.1109/CICT51604.2020.9312099.

[14] W. Y. Wang, "'Liar, Liar Pants on Fire': A New Benchmark Dataset for Fake News Detection." arXiv, May 01, 2017, https://doi.org/10.48550/arXiv.1705.00648.

[15] C. O. Truică, E. S. Apostol, R. C. Nicolescu, and P. Karras, "MCWDST: A Minimum-Cost Weighted Directed Spanning Tree Algorithm for Real-Time Fake News Mitigation in Social Media," *IEEE Access*, vol. 11, pp. 125861–125873, 2023, https://doi.org/10.1109/ACCESS.2023.3331220.

[16] E. Aljohani, "Enhancing Arabic Fake News Detection: Evaluating Data Balancing Techniques Across Multiple Machine Learning Models," *Engineering, Technology & Applied Science Research*, vol. 14, no. 4, pp. 15947–15956, Aug. 2024, https://doi.org/10.48084/etasr.8019.

[17] N. Rai, D. Kumar, N. Kaushik, C. Raj, and A. Ali, "Fake News Classification using transformer based enhanced LSTM and BERT," *International Journal of Cognitive Computing in Engineering*, vol. 3, pp. 98–105, Jun. 2022, https://doi.org/10.1016/j.ijcce.2022.03.003.

[18] C. O. Truică, E. S. Apostol, and A. Paschke, "Awakened at CheckThat! 2022: Fake News Detection using BiLSTM and Sentence Transformer," presented at the CLEF 2022: Conference and Labs of the Evaluation Forum, Bologna, Italy, Sep. 2022.

[19] C. O. Truică and E. S. Apostol, "It's All in the Embedding! Fake News Detection Using Document Embeddings," *Mathematics*, vol. 11, no. 3, Jan. 2023, Art. no. 508, https://doi.org/10.3390/math11030508.

[20] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." arXiv, May 24, 2019, https://doi.org/10.48550/arXiv.1810.04805.

[21] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter." arXiv, 2019, https://doi.org/10.48550/ARXIV.1910.01108.

[22] Y. Liu *et al.*, "RoBERTa: A Robustly Optimized BERT Pretraining Approach." arXiv, Jul. 26, 2019, https://doi.org/10.48550/arXiv.1907.11692.

[23] A. Conneau *et al.*, "Unsupervised Cross-lingual Representation Learning at Scale." arXiv, Apr. 08, 2020, https://doi.org/10.48550/arXiv.1911.02116.

[24] R. K. Kaliyar, A. Goswami, and P. Narang, "FakeBERT: Fake news detection in social media with a BERT-based deep learning approach," *Multimedia Tools and Applications*, vol. 80, no. 8, pp. 11765–11788, Mar. 2021, https://doi.org/10.1007/s11042-020-10183-2.

[25] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed Representations of Words and Phrases and their Compositionality." arXiv, 2013, https://doi.org/10.48550/arXiv.1310.4546.

[26] W. Lifferth, "Fake News." https://kaggle.com/fake-news.

[27] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, Aug. 1997, https://doi.org/10.1162/neco.1997.9.8.1735.

[28] V. I. Ilie, C. O. Truica, E. S. Apostol, and A. Paschke, "Context-Aware Misinformation Detection: A Benchmark of Deep Learning Architectures Using Word Embeddings," *IEEE Access*, vol. 9, pp. 162122–162146, 2021, https://doi.org/10.1109/ACCESS.2021.3132502.