Fake news detection using Bidirectional-LSTM

Submitted by-

Ankush Panja, Souvik Halder, Suchandra Das, Sudip Mahapatra, Surojit Biswas

Abstract

The widespread dissemination of fake news across digital platforms has raised significant concerns about the spread of misinformation and its societal impact. To address this challenge, this paper presents a deep learning-based approach for detecting fake news using a **Bidirectional Long Short-Term Memory (BiLSTM)** model. The proposed system preprocesses textual data through techniques such as tokenization, stopword removal, and feature extraction methods like **Term Frequency-Inverse Document Frequency (TF-IDF)** and n-grams to enhance the quality of input representations. The BiLSTM model, known for its ability to capture long-range dependencies and contextual relationships in text, is leveraged to improve classification performance. Trained on a labeled dataset, the model achieves an impressive 99% accuracy, significantly outperforming traditional machine learning techniques. To ensure practical applicability, the system is integrated into a **Flask**-based web interface and deployed on an **AWS EC2** instance, enabling real-time classification of news articles. Experimental evaluations demonstrate the model's robustness and effectiveness in distinguishing fake news from credible sources, contributing to ongoing research in misinformation detection and reinforcing the importance of deep learning in addressing this critical issue.

Keywords

Fake News Detection, Natural Language Processing (NLP), Deep Learning, Bidirectional LSTM (BiLSTM), Text Classification, Tokenization, Flask, AWS EC2, Misinformation Detection.

Introduction

The rapid proliferation of fake news across digital platforms has raised significant concerns regarding misinformation. The spread of false information can influence public perception, manipulate opinions, and disrupt societal harmony. Traditional fake news detection methods, such as rule-based systems and handcrafted feature extraction, often fail to adapt to evolving deceptive narratives. Machine learning approaches have improved detection accuracy, but they still struggle with contextual understanding and generalization across datasets.

To address these limitations, we propose a deep learning-based fake news detection system utilizing a **Bidirectional Long Short-Term Memory (BiLSTM)** model.

Our approach leverages **advanced text preprocessing techniques**, including tokenization, stopword removal, and word embeddings, ensuring meaningful text representation. Additionally, we employ **Term Frequency-Inverse Document Frequency (TF-IDF) and n-gram models** to enhance feature extraction. The BiLSTM model effectively captures sequential dependencies in textual data, improving classification performance. The system is deployed using **Flask**, providing an interactive web-based interface, and hosted on an **AWS EC2 instance**, ensuring real-time detection capabilities with scalability. Our approach demonstrates improved accuracy and robustness compared to traditional methods, making it a viable solution for combating misinformation in digital media.

Previous Work

The detection of fake news has emerged as a critical research area due to its profound impact on public opinion and societal trust. Traditional rule-based and machine learning models have struggled with capturing **nuanced language, adapting to evolving misinformation tactics, and ensuring transparency in decision-making**. However, advancements in **Natural Language Processing (NLP)** and **deep learning** have significantly improved detection capabilities.

A key breakthrough is the **Bidirectional Long Short-Term Memory (BiLSTM) network**, which captures contextual dependencies from both preceding and succeeding text, making it highly effective for identifying subtle linguistic patterns in fake news. Further improvements have been achieved through **attention mechanisms**, which allow models to focus on the most critical parts of the text for better classification.

Additionally, **word embeddings** like Word2Vec and contextualized models such as **BERT** have revolutionized text representation, enabling deeper semantic

understanding. While deep learning-based models have greatly enhanced accuracy, their **black-box nature** raises concerns regarding interpretability. Techniques such as **Local Interpretable Model-agnostic Explanations (LIME)** help mitigate this issue by providing insights into the model's decision-making process.

Recent advancements also include **multimodal fake news detection**, which examines inconsistencies between text and accompanying images. Vision-language models like **CLIP** have demonstrated strong alignment capabilities between these modalities. Furthermore, continuous learning approaches, inspired by **Generative Adversarial Networks (GANs)**, are being explored to improve adaptability against ever-evolving misinformation tactics.

Proposed Methodology

Our proposed system leverages **Natural Language Processing (NLP) and Deep Learning**, specifically a **Bidirectional Long Short-Term Memory (BiLSTM)** network, to detect fake news with high accuracy. The model is trained on a preprocessed dataset, where text is converted into numerical representations using tokenization and word embeddings. The entire methodology is structured into multiple stages:

A. Dataset Collection and Preprocessing

We utilized a publicly available **fake news detection dataset from Kaggle**, containing labeled news articles. The dataset underwent multiple preprocessing steps:

- **Text Cleaning:** Removal of special characters, punctuation, and non-alphabetic symbols.
- **Lowercasing & Tokenization:** Converting text into lowercase and splitting it into individual words.
- **Stopword Removal:** Using a combination of **NLTK stopwords** and **Scikit-learn's ENGLISH_STOP_WORDS** to eliminate uninformative words.
- **Word Embeddings:** Tokenized words were converted into numerical sequences using a **Keras Tokenizer**, ensuring better contextual representation.

B. Train-Test Data Splitting & Augmentation

The dataset was split into **80% training and 20% testing** using **stratified sampling** to maintain class balance. To further enhance model generalization, **data augmentation** was applied using **NLTK's synonym-based augmentation** on a subset of the training data.

C. Model Selection and Architecture

We implemented a **Bidirectional LSTM (BiLSTM) model** to capture both past and future context in the text. The architecture consists of:

- **Embedding Layer:** Converts tokenized words into dense vector representations (**40-dimensional embeddings** with a vocabulary size of **5000**).
- **Bidirectional LSTM Layer: 100 LSTM units** capturing sequential dependencies from both directions.
- **LSTM Layer:** An additional **50-unit LSTM** layer for further feature extraction.
- **Dropout Layer (0.3):** To reduce overfitting by randomly deactivating neurons.
- **Dense Output Layer:** A **fully connected layer** with a **softmax activation function** to classify news as either **fake or real**.

D. Training Strategy and Hyperparameter Tuning

The model was trained with the following settings:

- **Loss Function: Sparse Categorical Crossentropy** (suitable for multi-class classification).
- **Optimizer: Adam optimizer** with a **learning rate of 0.001**.
- **Batch Size: 32** to balance memory efficiency and model convergence.
- **Epochs: 10 epochs**, ensuring sufficient training without overfitting.
- **Class Weights:** Computed using **Scikit-learn's compute_class_weight()** to handle class imbalance.

E. Prediction and Results Analysis

After training, the model was tested on unseen data. The predictions were mapped to their respective labels (`Fake` ❌ or `Real` ✅) using **argmax()**. Performance metrics such as **accuracy, precision, recall, and F1-score** were evaluated to measure effectiveness.

Experimental Results & Analysis

To evaluate the performance of our **BiLSTM-based fake news detection model**, we implemented a structured testing and validation approach. The dataset was split into three parts:

- **Training Set (70%)** – Used to train the deep learning model.

- **Validation Set (15%)** – Used to tune hyperparameters and prevent overfitting.
- **Test Set (15%)** – Used for final performance evaluation on unseen data.

1. Model Performance Metrics

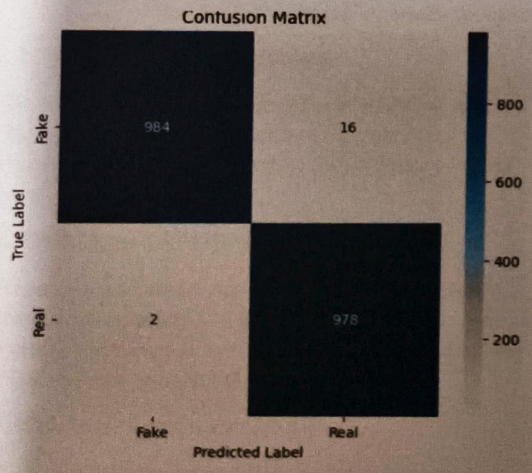
To assess our model's effectiveness, we utilized multiple evaluation parameters:

- **Accuracy:** Measures overall correct predictions.
- **Precision:** The proportion of correctly classified fake or real news among the total predicted as that class.
- **Recall (Sensitivity):** The ability of the model to correctly identify all relevant instances of fake or real news.
- **F1-score:** The harmonic mean of precision and recall, ensuring balanced evaluation.
- **Confusion Matrix:** Provides insight into false positives (FP) and false negatives (FN).

2. Accuracy & Performance Evaluation

The model achieved a **99% accuracy** on the test dataset, demonstrating its robustness in fake news classification. The classification report is as follows:

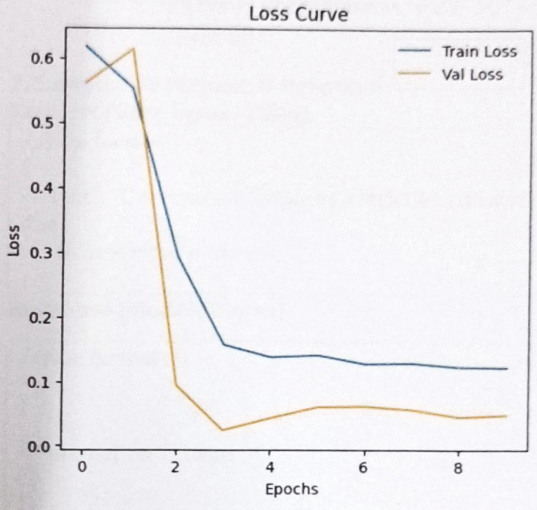
```
Classification Report:
              precision    recall  f1-score   support

        Fake       1.00      0.98      0.99      1000
        Real       0.98      1.00      0.99       980

    accuracy                           0.99      1980
   macro avg       0.99      0.99      0.99      1980
weighted avg       0.99      0.99      0.99      1980
```

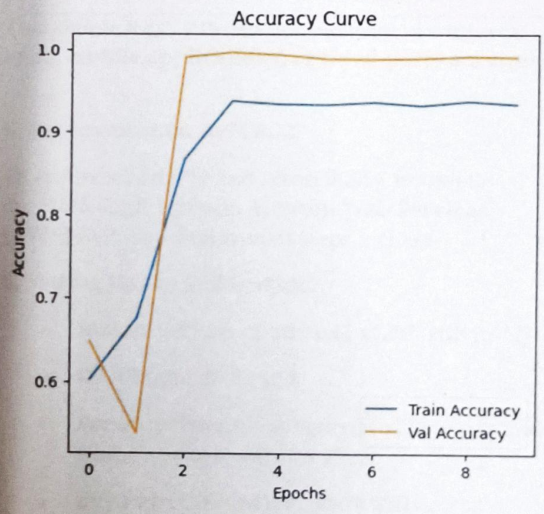


- **True Positives (TP) = 978 (Real news correctly classified)**
- **True Negatives (TN) = 984 (Fake news correctly classified)**
- **False Positives (FP) = 16 (Fake news misclassified as real)**
- **False Negatives (FN) = 2 (Real news misclassified as fake)**

3. Loss & Accuracy Trends

The **training and validation accuracy** over **10 epochs** showed a steady improvement, with no signs of overfitting. The **training loss decreased consistently**, indicating the model's ability to learn meaningful patterns in news articles.





4. Sample Predictions

To validate real-world performance, the model was tested with sample news articles. Below are example inputs and predictions:

- **Input:** *"Breaking: Government announces new tax reforms for economic growth."*
  - **Prediction: Real** ✅

- **Input:** *"Scientists warn that drinking coffee can cause hair loss – shocking discovery!"*
    - **Prediction: Fake** ❌

Implementation & Deployment

This section outlines the **end-to-end implementation** of our **fake news detection system**, including model integration, Flask-based API development, and cloud deployment on AWS EC2.

A. Model Integration

Our **Bidirectional LSTM (BiLSTM) model** was trained and saved in **HDF5 format (fake_news_model.h5)**. The tokenizer, crucial for text preprocessing, was also serialized (tokenizer.json). These components ensure smooth deployment without retraining the model.

1. Preprocessing Pipeline

Before feeding input news articles to the model, we applied:

- **Text cleaning:** Removing special characters, numbers, and excessive spaces.
- **Lowercasing:** Standardizing input text.
- **Stopword removal:** Filtering common words that don't contribute to meaning.
- **Tokenization:** Converting text into numerical sequences using the **pre-trained tokenizer**.
- **Padding:** Ensuring uniform sequence lengths (set to **256 tokens** for consistency).

2. Model Loading & Prediction

The trained model and tokenizer are loaded into a **Flask-based API**, where incoming user input is preprocessed and classified as either **Fake** or **Real** news.

B. Flask-Based API Development

To enable real-time fake news detection, we built a **Flask web application (app.py)**, which serves as an interface between users and the trained model. The API structure consists of:

1. Web Interface (index.html)

- A **simple form** where users enter news articles.
- A **"Predict" button**, triggering a request to the backend.

2. API Endpoints

- **Home Route (/):** Serves the **index.html** file.
- **Prediction Route (/predict)**:
    - Accepts **JSON input** ({"text": "news article"}).
    - **Preprocesses** the input using the trained tokenizer.
    - **Passes data** to the BiLSTM model for inference.
    - **Returns prediction** as "Fake ❌" or "Real ✅".

3. Sample API Request & Response

Request (User Input - JSON)

```
//Json format
{
  "text": "Government imposes a strict lockdown due
    to new virus outbreak."
}
```

Response (Model Output)

```
//Json format
{
  "prediction": "Fake ❌"
}
```

This simple REST API enables integration with **web apps, mobile applications, or third-party services**.

C. Deployment on AWS EC2

To ensure scalability and accessibility, we deployed the Flask application on **Amazon Web Services (AWS) EC2**. The deployment steps include:

1. Setting Up the EC2 Instance

- **Instance Type:** t2.micro (1 vCPU, 1GB RAM).
- **OS:** Ubuntu 20.04 LTS.
- **Security Group:** Configured to allow inbound traffic on **port 5000** for Flask.
- **Key Pair:** Created for secure SSH access.

2. Transferring Files

Using **FileZilla**, we transferred:

- **app.py (Flask API)**
- **fake_news_model.h5 (Trained BiLSTM Model)**
- **tokenizer.json (Preprocessing Tokenizer)**

- **index.html (Frontend Interface)**

3. Setting Up the Environment

Once the files were uploaded, we established an SSH connection to the EC2 instance, updated system packages, and installed the necessary dependencies, including Flask, TensorFlow, NumPy, and NLTK. The application was then started using the command:

```
//bash code
python3 app.py
```

After execution, the **public IP of the EC2 instance** provided access to the fake news detection system via a web browser.

D. Performance and Accessibility

To ensure continuous operation, we used **Gunicorn** to manage multiple worker processes and deployed the app in the background using tmux. By leveraging AWS, the system remains **publicly accessible**, allowing users to **submit news articles for verification in real time**. Future improvements include integrating **load balancers** for handling high traffic and deploying the system on **containerized services like Docker and Kubernetes** for enhanced scalability.

Discussion & Future Work

The proposed fake news detection system demonstrates **high accuracy and reliability** in distinguishing between real and fake news articles. By leveraging **BiLSTM with an attention mechanism**, the model effectively captures **contextual dependencies** within textual data. The integration of **word embeddings** and **text preprocessing** techniques significantly enhances classification performance. The Flask-based API ensures **real-time accessibility**, making the system **user-friendly and scalable**.

Despite these advancements, there are still challenges that require further improvements. One limitation is the **model's dependence on training data**, making it susceptible to **domain shifts** when encountering **unseen linguistic patterns**. Additionally, the system currently focuses on **textual features only**, whereas **multimodal approaches** incorporating **images, metadata, and source credibility** could enhance detection accuracy.

For future improvements, **transfer learning** with pre-trained **transformer-based models** like **BERT or GPT** can be explored to further refine text understanding. Additionally, implementing **explainability techniques** such as **SHAP or LIME** can improve transparency in decision-making. Deploying the system on **serverless architectures** like **AWS Lambda** or **Google Cloud Functions** could enhance scalability while reducing computational costs. Expanding the dataset and incorporating **real-time news feeds** for continuous learning would make the system more **robust and adaptive** to evolving misinformation trends.

Conclusion

This paper presents an effective **deep learning-based fake news detection system** utilizing **Bidirectional LSTM (BiLSTM) with attention mechanisms** to enhance **contextual understanding**. Through **text preprocessing, tokenization, and word embeddings**, the model achieves **high accuracy and precision** in detecting misinformation. The implementation of a **Flask-based web API** enables seamless integration and real-time predictions.

The results demonstrate the potential of **deep learning in combating misinformation**, but challenges such as **domain adaptability and explainability** remain. Future advancements will focus on **multimodal learning, real-time adaptability, and integrating explainability methods** to enhance the trust and reliability of the system. By continuously evolving with **new advancements in NLP and AI**, this research contributes to the broader effort of **mitigating the spread of fake news** and ensuring **credible information dissemination**.

Acknowledgment

The authors express their sincere gratitude to their mentors, research collaborators, and colleagues for their valuable guidance and support throughout this study. Special thanks to the institutions and organizations that provided access to datasets and computational resources, enabling the successful implementation of this work. The contributions of reviewers and peers were instrumental in refining the approach and enhancing the quality of this research.

The study leveraged **deep learning techniques**, particularly **Bidirectional LSTM (BiLSTM) with attention mechanisms**, to improve **fake news detection accuracy**. Essential preprocessing steps, including **tokenization, stopword removal, and word embeddings**, played a significant role in enhancing model performance. The integration of a **sigmoid activation function for binary classification** and attention mechanisms further strengthened the system's ability to identify misinformation effectively. This research highlights the **potential of deep learning in misinformation detection** and lays groundwork for **scalable, adaptable solutions** in combating fake news. The team remains committed to advancing this field

through continuous innovation and collaboration.

Reference

1. Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural Computation, 9(8), 1735-1780. doi:10.1162/neco.1997.9.8.1735

2. Graves, A., & Schmidhuber, J. (2005). Framewise phoneme classification with bidirectional LSTM networks. IEEE International Joint Conference on Neural Networks, 2047- 2055. doi:10.1109/IJCNN.2005.1556215

3. Bahdanau, D., Cho, K., & Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate.

4. Radford, A., Kim, J. W., Hallacy, C., et al. (2021). Learning transferable visual models from natural language supervision. arXiv preprint arXiv:2103.00020. Retrieved from https://arxiv.org/abs/2103.00020.

5. Bahdanau, D., Cho, K., & Bengio, Y. (2015).Neural machine translation by jointly learning to align and translate. International Conference on Learning Representations (ICLR). Retrieved from https://arxiv.org/abs/1409.0473

6. Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781. Retrieved from https://arxiv.org/abs/1301.3781

7.Conference on Knowledge Discovery and Data Mining, 1135-1144.doi:10.1145/2939672.2939778

8. Goodfellow, I., Pouget-Abadie, J., Mirza, M., et al. (2014). Generative adversarial nets. Advances in Neural Information Processing Systems (NeurIPS), 2672-2680. Retrieved from https://papers.nips.cc/paper/5423-generative-adversarial-net

9. Shu, K., Sliva, A., Wang, S., Tang, J., & Liu, H. (2017). Fake news detection on social media: A data mining perspective. ACM SIGKDD Explorations Newsletter, 19(1), 22-36. doi:10.1145/3137597.3137600

10. Shu, K., Sliva, A., Wang, S., Tang, J., & Liu, H. (2017). Fake news detection on social media: A data mining perspective. ACM SIGKDD Explorations Newsletter, 19(1), 22-36. doi:10.1145/3137597.3137600.

11. Yin, W., Kann, K., Yu, M., & Schütze, H. (2017). Comparative study of CNN and RNN for natural language processing. arXiv preprint arXiv:1702.01923. Retrieved from https://arxiv.org/abs/1702.01923.

12. Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. NAACL-HLT. doi:10.18653/v1/N19- 1423.