NLP-DRIVEN FAKE NEWS DETECTION USING MACHINE LEARNING AND DEEP LEARNING MODELS

Riddhi Sharma

Delhi Technological University

ABSTRACT

The spread of false news in digital media has emerged as a major concern, contributing to disinformation and societal divide. This project aims to create a robust system for detecting false news using machine learning (ML) and deep learning (DL) approaches. We used the ISOT dataset, which included articles tagged as true or false, and performed rigorous preprocessing procedures to improve model performance. Multiple ML models, such as Support Vector Machines (SVM), Random Forest, Logistic Regression (LR) and Multinomial Naïve Bayes (NB) were trained in our study, as well as DL models such Long Short-Term Memory networks (LSTM), Convolutional Neural Networks (CNN), Bidirectional LSTMs (BiLSTM), Simple Neural Networks (NN) and Gated Recurrent Unit (GRU). In addition to preprocessing, TF-IDF and GloVe word embeddings were used to extract useful features from the textual data.

This efforst provides a comprehensive framework for detecting false news, using the benefits of both ML and DL models to address a serious societal issue. The suggested method provides a scalable and effective strategy for combating disinformation in digital media, with practical implications for increasing information integrity and encouraging informed public conversation.

INTRODUCTION

The spread of false news in digital media has serious effects for countries throughout the world, affecting public opinion, confusing people, and adding to social divide. Detecting and countering false news has consequently become an urgent priority. This project will explore several machine learning (ML) and deep learning (DL) models to create an effective system for detecting false news.

The major purpose of this study is to determine the best-performing model based on a variety of assessment parameters. We trained and evaluated our models using the ISOT dataset, which comprises articles tagged as real or bogus. Fake news detection is the issue of separating fake news stories from authentic ones, which necessitates algorithms that can recognize complicated language patterns and contextual clues.

To do this, we used preprocessing techniques to clean and prepare the data, such as TF-IDF and GloVe word embeddings, to extract relevant characteristics from textual data. Our research includes training a number of ML models, including Support Vector Machines (SVM), Random Forest, Logistic Regression (LR) and Multinomial Naïve Bayes (NB) which are well-known for their effectiveness in text categorization tasks. We also looked at deep learning models such as Long Short-Term Memory networks (LSTM), Convolutional Neural Networks (CNN), Bidirectional LSTMs (BiLSTM), Simple Neural Networks (NN) and Gated Recurrent Unit (GRU), which are noted for their capacity to capture detailed patterns in sequential data.

These models were evaluated using a variety of measures, including accuracy, precision, recall, and F1-score, to determine their ability to discern between false and authentic news items. Our findings illustrate the benefits and shortcomings of each strategy, providing insights into the most successful strategies for countering false news in digital media.

Dataset Description

The ISOT dataset utilized in this study contains a fair mix of factual and false news stories taken from credible news providers and identified fake news sites. The real news stories are sourced from Reuters.com, a well-known news website noted for its accurate reporting. The false news stories, on the other hand, were gathered from untrustworthy sources identified by Politifact and Wikipedia, with a primary focus on political and global news subjects.

The dataset consists of two CSV files:

- **True.csv**: Contains over 12,600 articles from Reuters.com.
- **Fake.csv**: Contains over 12,600 articles from various unreliable sources.

News	**Size (Number of articles)**	***Subjects***	
Real-News	21417	**Type**	**Articles size**
		World-News	*10145*
		Politics-News	*11272*
Fake-News	23481	**Type**	**Articles size**
		Government-News	*1570*
		Middle-east	*778*
		US News	*783*
		left-news	*4459*
		politics	*6841*
		News	*9050*

Referenced from [Fake News Detection Datasets (kaggle.com)]

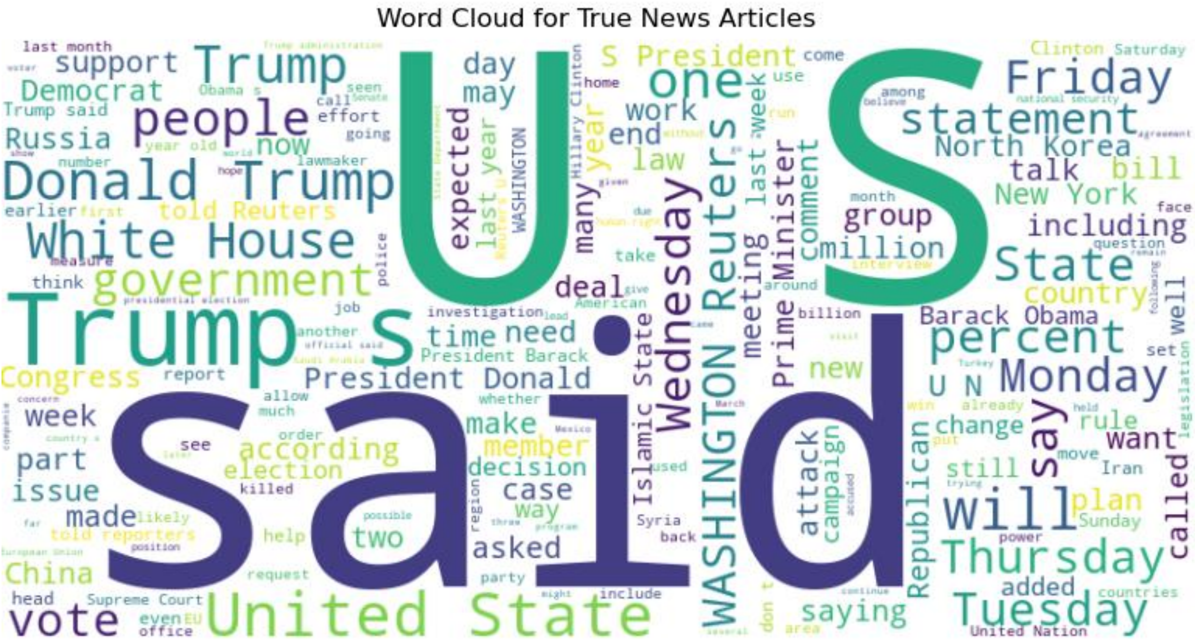
Each article in the dataset includes the following information:

- Article title
- Article text
- Subject
- Date published

The data collected spans articles primarily from 2016 to 2017, with an emphasis on maintaining the original punctuation and errors present in the fake news texts, ensuring the dataset's authenticity for analysis.

Word Clouds

Word clouds were created to graphically show the most frequently used terms in both real and fraudulent news items. These word clouds aid in recognizing the primary themes and prevalent phrases used in each category, offering insights into the language and issues discussed.



METHODOLOGY

The ISOT dataset underwent several preprocessing steps to prepare it for model training and evaluation. These steps ensured that the text data was clean, standardized, and ready for feature extraction and modelling.

1. **Loading and Labelling:**

   - The dataset was loaded from two CSV files: **True.csv** and **Fake.csv**.
   - Articles from the **True.csv** file were labelled as '0' (real news), while articles from the **Fake.csv** file were labelled as '1' (fake news).
   - After labelling, the datasets were combined (concatenated) into a single dataset and shuffled to ensure randomization.

2. **Removing Unnecessary Columns:**

   - Columns such as 'title', 'subject', and 'date' were dropped as they were not necessary for the fake news detection task. Only the article text and the corresponding labels were retained for further processing.

3. **Text Preprocessing:**

   - **Lowercasing:** All text was converted to lowercase to ensure uniformity and to prevent the model from treating words with different cases as different features.
   - **Removing Punctuation:** Punctuation marks were removed from the text using Python's string operations.
   - **Tokenization:** The text was tokenized into words or tokens. Tokenization is the process of splitting the text into smaller units (tokens) such as words or phrases.
   - **Removing Stopwords:** Stopwords (commonly used words like 'and', 'the', 'is') were removed from the text using the NLTK library. Stopwords typically do not contribute to the meaning of the text and can be safely ignored.

4. **Feature Extraction:**

   - **TF-IDF Vectorization (for ML models):** TF-IDF (Term Frequency-Inverse Document Frequency) vectorization is a technique for converting textual input into numerical characteristics suitable for ML models. In this study, TF-IDF was used to discriminate between false and genuine news items based on the frequency of phrases in each. The TF-IDF vectorization procedure begins with creating a lexicon from the complete corpus of articles. Each article is then converted into a numerical vector, with each dimension representing the relevance of a phrase in the article in comparison to the corpus. For our study, the TF-IDF vectorizer was configured to limit the number of features to the top 5000 most frequent terms. This step ensures that only the most relevant terms are considered, reducing noise and improving model performance.

- **Tokenization and Padding (for DL models):** To properly capture the semantic content of the articles, deep learning algorithms require distinct preprocessing of text input. To transform textual data into numerical sequences, tokenization was used, which is the act of breaking text into smaller parts known as tokens. Following tokenization, sequences were padded to guarantee consistent length across all articles. This phase is critical for deep learning models that need fixed input sizes. By padding sequences to a maximum length of 100 tokens, we ensure that each piece is uniformly displayed, regardless of its original length. In addition, pre-trained GloVe embeddings were employed to represent words as dense vectors in the deep learning models. GloVe embeddings capture semantic associations between words by analyzing their co-occurrence in a huge corpus of text. This allowed the model to learn from the context of words, which improved its capacity to recognize nuanced patterns in false news items.

MACHINE LEARNING MODELS

1. **Logistic Regression:** Logistic Regression is a linear model used for binary classification. It models the probability of a certain class or event existing.
2. **Support Vector Machine:** SVM creates a hyperplane in a high-dimensional space to divide classes. It is effective in high-dimensional areas when the number of dimensions exceeds the number of samples.
3. **Random Forest:** Random Forest creates many decision trees and then combines them to get a more accurate and reliable prediction. It is effective at handling huge datasets with high dimensionality.
4. **Multinomial Naïve Bayes:** Naive Bayes classifiers are a type of basic probabilistic classifier that uses Bayes' theorem with strong (naive) independence assumptions across features.

DEEP LEARNING MODELS

1. **Simple Neural Network:** A basic neural network architecture with an embedding layer, dense layers, and dropout for regularization.
2. **LSTM (Long Short-Term Memory):** A type of recurrent neural network (RNN) architecture well-suited for sequence prediction tasks, capable of learning long-term dependencies.
3. **Bidirectional LSTM:** A variant of LSTM that enhances model performance by processing the input sequence both forward and backward.
4. **CNN-LSTM:** Combines convolutional neural network (CNN) layers for feature extraction with LSTM layers for sequence modelling.
5. **GRU (Gated Recurrent Unit):** A type of RNN architecture similar to LSTM but with fewer parameters, making it faster to train.

EVALUATION METRICS:

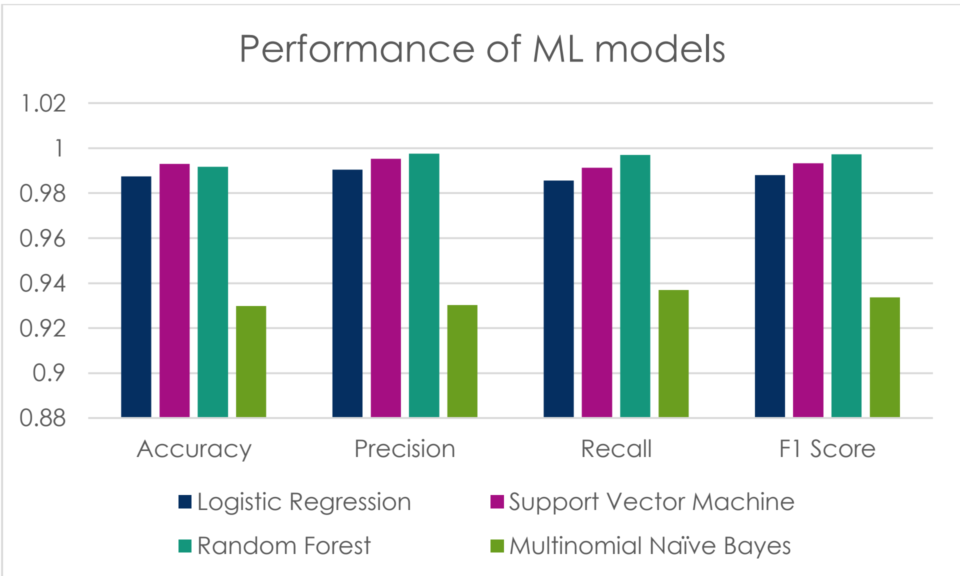
1. **Accuracy:** Overall accuracy of predictions.
2. **Precision:** Proportion of true positive predictions out of all positive predictions.
3. **Recall:** Proportion of true positive predictions out of all actual positives.

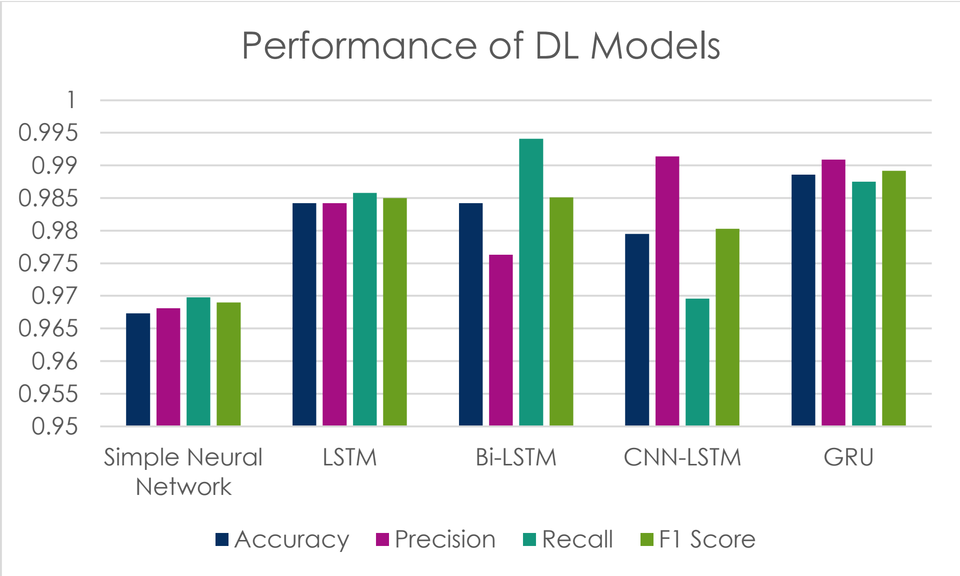
4. **F1-score:** Harmonic mean of precision and recall.

EVALUATION AND RESULTS

Algorithms	Accuracy	Precision	Recall	F1 Score
Logistic Regression	0.9874	0.9904	0.9856	0.9880
Support Vector Machine	0.9930	0.9953	0.9913	0.9933
Random Forest	0.9917	0.9975	0.9970	0.9973
Multinomial Naïve Bayes	0.9298	0.9303	0.9370	0.9337
Simple Neural Network	0.9673	0.9681	0.9698	0.9690
LSTM	0.9842	0.9842	0.9858	0.9850
Bi-LSTM	0.9842	0.9763	0.9941	0.9851
CNN-LSTM	0.9795	0.9914	0.9696	0.9803
GRU	0.9886	0.9909	0.9875	0.9892

Support Vector Machine performs the best among the ML models with an accuracy of 99.30% while Random Forest has the highest F1 Score of 0.9973. GRU has the highest accuracy of 98.86% and F1 Score of 0.9892.





CONCLUSION

The goal of this project was to create and test several models for distinguishing between legitimate and false news stories. The study tested many machine learning and deep learning models, including Logistic Regression, SVM, Random Forest, Multinomial Naive Bayes, Simple Neural Network, LSTM, Bidirectional LSTM, CNN-LSTM, and GRU. These models were trained and evaluated using a dataset containing both actual and false news stories. The performance of each model was measured using criteria like as accuracy, precision, recall, and F1-score. The findings of this experiment are critical for the development of automated systems to prevent the spread of disinformation and false news. Such systems have the ability to improve media literacy, promote accurate information, and reduce the social consequences of disinformation.

FUTURE WORK

Despite the project's accomplishments, various areas for additional investigation and improvement remain. Further hyperparameter adjustment and optimization of models such as Bidirectional LSTM and CNN-LSTM may result in even greater performance. Exploring other designs or pre-trained embeddings such as BERT may be advantageous. Incorporating more varied and broad datasets encompassing many themes and languages may improve model generalization and resilience. Creating a real-time system that watches news stories as they are published and uses the trained model to detect possibly fraudulent news in real time. Using characteristics from various modalities, such as photographs and videos that accompany news stories, to increase the accuracy of false news identification. Investigating possible biases in the dataset and models, and using fairness-aware strategies to mitigate them. By tackling these issues, future research can enhance the field of false news identification and help to design more accurate and robust methods for combatting disinformation.