Arpit Prakash Amborkar

Fake News Detection Using NLP and Machine Learning

ADVISOR: Humera Noor Minhas

Summer 2025

A logo for a university

Description automatically generated

Table of Contents

[Abstract 2](#_Toc202305960)

[1. Introduction & Problem Motivation 3](#_Toc202305961)

[2. Problem Statement 4](#_Toc202305962)

[2.1 Scope of Analysis 4](#_Toc202305963)

[3. Background & Literature Review 5](#_Toc202305964)

[4. Data 6](#_Toc202305965)

[4.1 LIAR Dataset 6](#_Toc202305966)

[4.2 FakeNewsNet Dataset 7](#_Toc202305967)

[4.3 ISOT Fake News Dataset 8](#_Toc202305968)

[5. Model and Analysis 9](#_Toc202305969)

[5.1 Data Cleaning & Preprocessing 9](#_Toc202305970)

[5.2 Exploratory Data Analysis 10](#_Toc202305971)

[5.3 Feature Engineering 11](#_Toc202305972)

[5.4 Train & Evaluate Models 12](#_Toc202305973)

[6. Results and Recommendations 14](#_Toc202305974)

[6.1 Classical Models 14](#_Toc202305975)

[6.2 Deep Learning 15](#_Toc202305976)

[6.3 BERT Fine-tuning 17](#_Toc202305977)

[6.4 Model Deployment 18](#_Toc202305978)

[7. Conclusions 20](#_Toc202305979)

[8. Acknowledgements 21](#_Toc202305980)

[9. References 21](#_Toc202305981)

## Abstract

Misinformation is rising day by day in digital media, this raises grave concerns globally. Fake News, defined as the misinformation that replicate the format of mainstream news but lack verifiable evidence that can influence public opinion, incite panic, or sway political outcomes i.e. elections. This project presents a framework to detect fake news using Natural Language Processing (NLP) and Machine Learning. By implementing multiple models such as Logistic Regression, Random Forest, Support Vector Machine (SVM), Naive Bayes, Long Short-Term Memory (LSTM), Convolutional Neural Networks (CNN), and fine-tuned Bidirectional Encoder Representations from Transformers (BERT), we compare classical, deep learning, and transformer-based BERT models.

Used benchmark datasets, perform extensive preprocessing (cleaning, tokenization, stemming/lemmatization, embedding). And analyzed results with accuracy, F1-score, and precision metrics. Highest performing model (SVM) achieved 86.52% accuracy. For deployment, the final model was integrated into a simple Flask Web Application. This app allows users to input news content and receive predictions in real-time. The Application is hosted using render, with the deployment managed via GitHub integration. This ensures scalability, ease of maintenance, and practical usability for end users. The project showcases a complete pipeline from model development to real-world deployment, laying the foundation for future enhancement involving real-time data streams and multimodal fake news detection. Further, the framework can be enhanced for citing the relevant articles along or with fake news detection using Large Language Models (LLMs).

## Introduction & Problem Motivation

Fake News is not a new phenomenon, but with the rise in digitization and social media platforms, its effect and impact on the people have become unprecedented. The year 2020 saw a steep rise of digital presence of people on online platforms. Everyone around the globe were forced to go online due to COVID-19 pandemic. This was the time when the World Economic Forum recognized digital misinformation as one of the top global risks in 2020[1]. The information flows through digital platforms like wildfire, contrary to mainstream news media through print and television. Social media’s viral nature amplifies the spread of misinformation, often before it can be fact-checked. Information Overload helps the misinformation spread quickly [2].

Most of the people stay connected and stay up to date while communicating with each other through, social media, messaging apps, microblogging websites(X(twitter)), and directly speaking to each other. This poses the challenge to stop fake news from spreading. Americans rely heavily on social media as the sources of news. After consuming news over these platforms, they are less likely to check the authenticity of the news and the sources. Ease in usability and sharing further reinforces the mind in sharing the news without scrutiny due to Cognitive biases. It is understanding of new information so that it fits in with what we already know. This tendency is extremely difficult to correct.

This poses the threat to the outcomes of the elections in the electoral democracy. Ranging from political manipulations [4] that has the power to tilt the scale of the world. This also consisting of fake videos or face swapped videos and its news covert by the news outlets are widely being used for malicious intent [5]. This directs steps to be taken to control and detect the spread of misinformation by detecting them early, flagging them, and detecting the authenticity of news being consumed by individual on or without request.

There is need for scalable, real-time fake news detection systems. There are Gaps in existing models in interpretability, generalizability, and real-world deployment. Ranging from web application, browser extension, X(twitter) bots, on device assistance for on screen news.

## **Problem Statement**

In today's digital age, the rapid spread of information across social media and online news platforms has changed how people consume and share content. While this technology offers numerous benefits, it also facilitates the widespread fake news, misinformation or disinformation that is intentionally fabricated to mislead readers for ideological, political, financial, or social gain.

Fake news articles often mimic the structure and language of legitimate journalism, making it challenging for readers, and even manual fact-checkers, to distinguish real from fake news. The speed with which fake news can spread pose significant risks, including:

* Misguided public opinion and behaviour.
* Loss of trust in legitimate media.
* Electoral and democratic event disruption.
* Public health and safety risks (e.g., misinformation during COVID-19 pandemic)

### Scope of Analysis

This project addresses the following core analytical objectives:

1. Binary Classification task

The problem is modeled as a supervised learning task, where the system classifies a given news article as either **Real** or **Fake** based on its content.

1. Dataset Utilization

Datasets containing labeled fake and real news articles were curated from publicly available sources like:

* 1. LIAR dataset
  2. FakeNewsNet corpus
  3. ISOT Dataset

1. Natural Language Processing (NLP) Pipeline

Applied preprocessing steps to convert unstructured text data into a format suitable for machine learning, including:

1. Text normalization
2. Tokenization
3. Lemmatization
4. Feature extraction
5. Model Development and Evaluation

Implemented and compared several classical ML and modern deep learning models, such as:

* 1. Logistic Regression, SVM, Random Forest
  2. LSTM, CNN
  3. Transformer-based model (BERT)

1. Model Deployment

The best-performing model (SVM) was packaged into a Flask web application and deployed via Render, allowing for real-time fake news detection accessible through a simple web interface.

## Background & Literature Review

The detection and mitigation of fake news have attracted significant attention in recent years, especially in the context of social media proliferation. Zhou and Zafarani (2020) provide a comprehensive survey that outlines the foundational theories behind fake news propagation, categorizing detection methods into content-based and social context-based approaches. Their work serves as a conceptual framework for understanding both the technical and psychological dimensions of misinformation.

Building upon this theoretical groundwork, Qian et al. (2021) introduce a novel Hierarchical Multi-modal Contextual Attention Network (HMCAN). This model integrates textual, visual, and contextual data to enhance fake news detection, emphasizing the importance of leveraging multiple modalities to improve accuracy. Their experimental results show superior performance over traditional models, underlining the need for nuanced, layered architectures.

Similarly, Kaliyar et al. (2021) propose FakeBERT, a deep learning model that utilizes BERT for understanding nuanced language patterns in social media posts. The model significantly outperforms classical machine learning methods by capturing deeper semantic meanings. This approach marks a shift toward transformer-based architectures that are now foundational in NLP.

In contrast, Ahmed, Traore, and Saad (2018) focus on opinion spam detection using basic text classification methods. While not leveraging deep learning, their research highlights the value of early detection techniques and shows promising results using traditional NLP pipelines.

Ozbay and Alatas (2020) contribute by testing various supervised machine learning algorithms (e.g., SVM, Decision Trees) to detect fake news. Their findings emphasize the importance of dataset quality and feature engineering, demonstrating that even non-deep learning approaches can yield competitive results with appropriate tuning.

Finally, Hossain et al. (2020) present COVIDLies, a domain-specific application of misinformation detection targeting COVID-19. Their system combines NLP and manual annotation strategies to tackle health-related misinformation. This work is a salient example of how fake news detection frameworks can be tailored for real-time crisis communication.

Together, these studies reflect the evolution of fake news detection from rule-based and classical models to highly contextual, deep learning-based approaches, with growing attention to multimodal and domain-specific data.

## Data

Three widely recognized benchmark datasets were utilized to train, evaluate, and compare the performance of fake news detection models.

### LIAR Dataset

The LIAR dataset, introduced by Wang (2017), is a large-scale benchmark dataset consisting of 12,836 short statements from various public figures, including politicians and media personalities. Each statement is manually labeled across six fine-grained categories: pants-fire, false, barely-true, half-true, mostly-true, and true. Metadata such as speaker affiliation, job title, and context are also included. This dataset emphasizes short-form political claims and supports both binary and multi-class classification tasks.

* 1. Label types: Multi-class (6 categories)
  2. Languages: English
  3. Applications: Fine-grained credibility classification

A screenshot of a computer

Description automatically generated

### 4.2 FakeNewsNet Dataset

FakeNewsNet is a comprehensive dataset built from real-world news content and social media engagement data. It includes news articles, corresponding user reactions on Twitter, and spatiotemporal information. The dataset covers both BuzzFeed and PolitiFact sources and is structured into fake and real news classes.

* 1. Label types: Binary (Fake / Real)
  2. Data modalities: News content, user profiles, replies, shares, timestamps
  3. Applications: Multimodal fake news detection, social context analysis

A screenshot of a computer

Description automatically generated

### 4.3 ISOT Fake News Dataset

The ISOT dataset, developed by the University of Victoria, contains over 44,000 English-language news articles divided into true and fake categories. It draws real news from Reuters and fake news from various unreliable news sources. Unlike LIAR, ISOT articles are significantly longer, allowing models to analyze complex text structures and context.

1. Label types: Binary (Fake / Real)
2. Content length: Full-length articles
3. Applications: Article-level fake news detection, text analysis

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

## 5. Model and Analysis

### 5.1 Data Cleaning & Preprocessing

The combined dataset is then cleaned. Here, the missing values, null values are filled by mean of the data. Duplicate rows are dropped. Tokenization, Removing Stopwords, Lammatization, are applied on the “text” coloumn. And the words text is converted into tokens. All the special characters are removed from the text. At the end of the operations the cleaned dataset is saved.

A screenshot of a computer

Description automatically generated

### 5.2 Exploratory Data Analysis

In Exploratory Data Analysis (EDA) the clean dataset is loaded. Then label distribution is checked using library ‘matplotlib’. It is done to check if the distribution is even.

A graph with a bar and a number of labels

Description automatically generated with medium confidence

After this class imbalance is checked. This is done to make sure the classes are balanced and the trained model will not be biased towards majority of the data.

Here image shows there is a class imbalance.

A screen shot of a computer

Description automatically generated

Since the label distribution is unbalanced. It would be biased against fake news. So, there is need to balance the dataset. We do this by under sampling the majority class i.e. 1(Real) Label. As We don't want to introduce synthetic data for Minority class i.e. o(Fake) Label.

A graph of a bar chart

Description automatically generated with medium confidence

Fig shows the balanced dataset after under sampling the majority class 1(Real).

### 5.3 Feature Engineering

Word cloud diagram for each class, Real or Fake are given in figure a and figure b respectively. And the code is run to get the most common words across the dataset.

A graph of different colored bars

Description automatically generated with medium confidence

Sentiment analysis is performed on the balanced dataset to get sentiment score for each text row. This reveals the emotions involved in the text. The figure shows the class wise sentiment analysis. This shows that fake news contains more sentiments that the real news.

A graph of a bar chart

Description automatically generated with medium confidence

For feature engineering we need TF-IDF Vectorization, Sentiment Score, and Text-Based Features. These combined features will be helpful in training the model.

### 5.4 Train & Evaluate Models

Dataset is devided or split into train and test set using Test-Train-Split scikit.learn library. First the models are build and given training parameters. After this the models are trained and results are saved.

After this the model evaluation is done using accuracy, f1-score and precision.

For Deep Learning models we evaluated the LSTM and CNN models. For these models the data is prepared again as these models doesn’t use TF-IDF Vectorization, so Tensor flow tokenizer is used.

Model architecture for LSTM is given in fig .

A screenshot of a computer

Description automatically generated

Model architecture for CNN is given in figure

A screenshot of a computer

Description automatically generated

Finally with BERT fine-tuning we have to begin data preparation again. It uses different tokenizer known as BertTokenizerFast within the ‘transformers’ library. Here the data is loaded , all the necessary libraries are installed. And this is followed by building model architecture configuring all the paramerters, epoch value and finally training with training data set and then testing with test data set. Then we train the model. It took around 145 mins to perform the training. Total of 1357 iterations, epoch=2.

## 6. Results and Recommendations

### 6.1 Classical Models

The figure show the results of the classical model training of logistic regression, random forest, naïve bayes and svm.

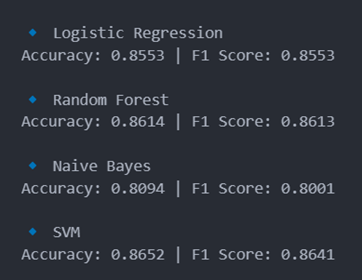
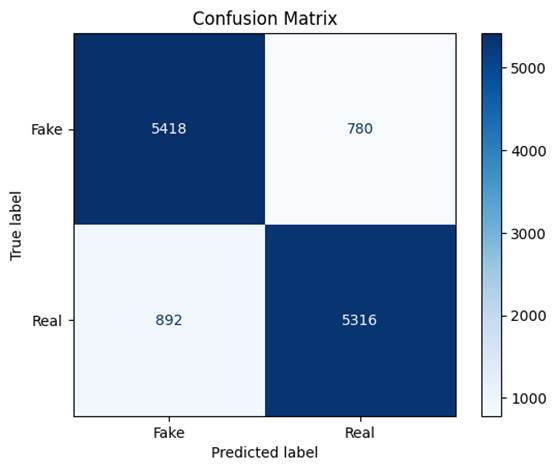
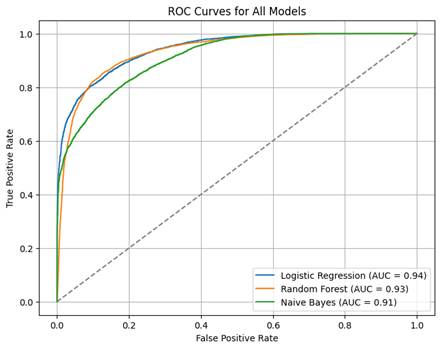


Figure show confusion matrix fro classical models trained .





SVM have the highest Accuracy and F1 score, so among classical model SVM performs the best. So we select SVM with 86.52% accuracy among classical models.

### 6.2 Deep Learning

LSTM results from graph shows its accuracy, model loss, learning rate, and overfitting gap with respect to epoch. The epoch was set to 20 but the model completed training in 11 epoch.

Figure show results of LSTM training.

A graph of different colored lines

Description automatically generated

A screenshot of a computer

Description automatically generated

CNN results from the graph shows its accuracy, model loss, learning rate, and overfitting gap with epoch. The epoch for CNN was set to 25 but it completed training in 11 epoch.

A graph of different types of data

Description automatically generated with medium confidence

Figure shows the CNN training results. This shows among deep learning models CNN performs the best with 84.91 % Final test Accuracy. But SVM out performs CNN. So we select SVM model.

A screenshot of a computer

Description automatically generated

### 6.3 BERT Fine-tuning

Shocking the BERT model performed poorly for the binary classification task. Figure show it managed to get just 76% accuracy. This is the lowest performance among the models in view.

A screenshot of a computer screen

Description automatically generated

For Deployment we choose SVM (with accuracy 86.52%) to be deployed to predict whether the news is Real or Fake.

### 6.4 Model Deployment

SVM is deployed through the use of web application. This web application was made using Flask and it is hosted using render and managed by github repository. So push update and deploy it is very easy with render.

Figure show the deployed and live render server.

A screenshot of a computer

Description automatically generated

Figure show the website page for web application.

A screenshot of a computer

Description automatically generated

Figure show the known Real Unseen News correctly predicts the news as Real News.

A screenshot of a computer

Description automatically generated

Figure show the known Fake Unseen News correctly predicts the news as Fake News.

A screenshot of a computer

Description automatically generated

## 7. Conclusions

In this project, a comprehensive pipeline for fake news detection was developed and tested using a variety of machine learning and deep learning approaches. By applying robust Natural Language Processing (NLP) techniques—such as tokenization, lemmatization, sentiment analysis, and TF-IDF vectorization—text data from multiple benchmark datasets was processed to extract meaningful features. The comparative analysis of models demonstrated that classical algorithms like Support Vector Machine (SVM) still hold competitive advantage over more complex models like BERT, particularly in binary classification tasks. SVM achieved the highest performance with an accuracy of 86.52%, outperforming deep learning models such as CNN and LSTM, as well as transformer-based architectures.

The project's scope extended beyond just model development to include real-world application and deployment. A user-friendly Flask-based web application was created, allowing users to input textual news content and receive real-time predictions on whether the content is real or fake. This deployment highlights the practical relevance and usability of the system, demonstrating that accurate fake news detection tools can be made accessible to the public. Such tools are vital in today’s digital ecosystem where misinformation spreads rapidly and widely, especially on social media platforms.

While the current solution effectively addresses binary classification, there is room for improvement and expansion. Future work could include enhancing model generalizability, integrating real-time social context signals (e.g., user behavior, source credibility), and deploying models on edge devices or browser extensions. The application of more advanced transformer models with domain-specific tuning, or the integration of Large Language Models (LLMs) capable of providing citations or counter-evidence, could further improve detection accuracy and trust. Ultimately, this project lays a strong foundation for scalable, interpretable, and deployable fake news detection systems with real-world impact.

## 8. Acknowledgements

I would like to express my sincere gratitude to all those who supported and contributed to the completion of this report. I am especially thankful to Humera Noor Minhas for their valuable guidance, constructive feedback, and continuous encouragement throughout the process.

## 9. References

1. World Economic Forum. (2020). The global risks report 2020 (15th ed.). <https://www.weforum.org/publications/the-global-risks-report-2020/>
2. Gorvett, Z. (2020, July 20). Information overload helps fake news spread, and social media knows it. Scientific American. <https://www.scientificamerican.com/article/information-overload-helps-fake-news-spread-and-social-media-knows-it/>
3. Shearer, E., & Mitchell, A. (2020, July 30). Americans who mainly get their news on social media are less engaged, less knowledgeable. Pew Research Center. <https://www.pewresearch.org/journalism/2020/07/30/americans-who-mainly-get-their-news-on-social-media-are-less-engaged-less-knowledgeable/>
4. Hern, A. (2023, October 16). The rise of the deepfake and the threat to democracy. The Guardian. <https://www.theguardian.com/technology/2023/oct/16/deepfake-threat-democracy-election-ai>
5. **Wang, C.** (2019, November 1). Deepfakes, revenge porn, and the impact on women. Forbes. https://www.forbes.com/sites/chenxiwang/2019/11/01/deepfakes-revenge-porn-and-the-impact-on-women/
6. Zhou, X., & Zafarani, R. (2020). A survey of fake news: Fundamental theories, detection methods, and opportunities. *ACM Computing Surveys (CSUR)*, *53*(5), 1-40.
7. Qian, S., Wang, J., Hu, J., Fang, Q., & Xu, C. (2021, July). Hierarchical multi-modal contextual attention network for fake news detection. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval* (pp. 153-162).
8. Kaliyar, R. K., Goswami, A., & Narang, P. (2021). FakeBERT: Fake news detection in social media with a BERT-based deep learning approach. *Multimedia tools and applications*, *80*(8), 11765-11788.
9. Ahmed, H., Traore, I., & Saad, S. (2018). Detecting opinion spams and fake news using text classification. *Security and Privacy*, *1*(1), e9.
10. Ozbay, F. A., & Alatas, B. (2020). Fake news detection within online social media using supervised artificial intelligence algorithms. *Physica A: statistical mechanics and its applications*, *540*, 123174.
11. Hossain, T., Iv, R. L. L., Ugarte, A., Matsubara, Y., Young, S., & Singh, S. (2020, December). COVIDLies: Detecting COVID-19 misinformation on social media. In *Proceedings of the 1st Workshop on NLP for COVID-19 (Part 2) at EMNLP 2020*.
12. Wang, W. Y. (2017). LIAR: A benchmark dataset for fake news detection [Data set]. University of California, Santa Barbara. <https://www.cs.ucsb.edu/~william/data/liar_dataset.zip>
13. Shu, K. (2018). KaiDMML/FakeNewsNet: A data repository with news content, social context and dynamic information for studying fake news on social media [Data set]. GitHub. https://github.com/KaiDMML/FakeNewsNet Reuters (true news), various fake news websites (fake news)
14. University of Victoria, Information Security and Object Technology (ISOT) Research Group. (2022, November 27). Fake news detection datasets [Data set]. UVic Online Academic Community. https://onlineacademiccommunity.uvic.ca/isot/2022/11/27/fake-news-detection-datasets/