Comparative Fake News Detector Using GossipCop and PolitiFact with Traditional and BERT-Based Models

A PROJECT REPORT
SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF THE DEGREE
OF
BACHELOR OF
TECHNOLOGY IN
Electronics and Communication Engineering

SUBMITTED BY:

Jatin (2K21/EC/107)
Kanesh (2K21/EC/110)
Rahul Ranjan (2K21/EC/173)

Under the Supervision of

Prof. Dinesh Kumar

Department of Electronics and Communication Engineering

DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road – 110042
New Delhi

Department of Electronics and Communication Engineering

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bawana Road, Delhi

CANDIDATE'S DECLARATION

We, Jatin (2K21/EC/107), Kanesh(2K21/EC/110) & Rahul Ranjan (2K21/EC/173) students of B. Tech (Electronics and Communication Engineering), hereby declare that the Project Dissertation titled — "Comparative Fake News Detector Using GossipCop and PolitiFact with Traditional and BERT-Based Models" which is submitted by us to the Department of Electronics and Communication Engineering, DTU, Delhi in fulfillment of the requirement for awarding of the Bachelor of Technology degree is not copied from any source without proper citation. This work has not previously formed the basis for the award of any Degree, Diploma, Fellowship or other similar title or recognition.

Place: New Delhi	**Jatin**	**Kanesh**	**Rahul Ranjan**
Date: 27th April,2025	**(2K21/EC/107)**	**(2K21/EC/110)**	**(2K21/EC/173)**

Department of Electronics and Communication Engineering

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bawana Road, Delhi

CERTIFICATE

I hereby certify that the Project title "Comparative Fake News Detector Using GossipCop and PolitiFact with Traditional and BERT-Based Models" which is submitted by Jatin (2K21/EC/107), Kanesh(2K21/EC/110) & Rahul Ranjan(2K21/EC/173) for fulfillment of the requirements for awarding of the degree of Bachelor of Technology (B. Tech) is a record of the project work carried out by the students under my guidance & supervision. To the best of my knowledge, this work has not been submitted in any part or fulfillment for a Degree or Diploma to this University or elsewhere.

Place: New Delhi | Prof. Dinesh Kumar

Date: 27th April,2025 | SUPERVISOR

ABSTRACT

Keywords: Fake News Detection, Machine Learning, Natural Language Processing, BERT, CNN, Hybrid Models

The proliferation of fake news on digital platforms has become a significant challenge, with serious consequences for public opinion, social stability, and democracy. This paper presents a novel approach to detecting fake news using machine learning techniques. We propose a framework that leverages natural language processing (NLP) and deep learning algorithms to classify news articles as either real or fake. Our approach incorporates a combination of feature extraction methods, including lexical, syntactic, and semantic analysis, to capture various characteristics of news content. We train multiple machine learning models, such as logistic regression, support vector machines (SVM), and deep neural networks (DNN), using a labeled dataset of news articles. The models are evaluated using standard metrics like accuracy, precision, recall, and F1-score to assess their performance. Our experimental results show that deep learning models, particularly convolutional neural networks (CNN) and recurrent neural networks (RNN), outperform traditional machine learning models in detecting fake news with high accuracy. The proposed system offers a promising solution to combat the spread of misinformation by automating the detection of fake news and providing a reliable tool for content verification. The framework can be extended and refined for real-time applications, contributing to the development of more trustworthy information ecosystems.

ACKNOWLEDGEMENT

The successful completion of any task is incomplete and meaningless without giving any due credit to the people who made it possible without which the project would not have been successful and would have existed in theory. First and foremost, we are grateful to Prof. O.P. Verma, HOD, Department of Electronics and Communication, Delhi Technological University, and all other faculty members of our department for their constant guidance and support, constant motivation and sincere support and gratitude for this project work. We owe a lot of thanks to our supervisor, Dr. Dinesh Kumar, Professor, Department of ECE, Delhi Technological University for igniting and constantly motivating us and guiding us in the idea of a creatively and amazingly performed Minor Project in undertaking this endeavor and challenge and also for being there whenever we needed his guidance or assistance. We would also like to take this moment to show our thanks and gratitude to one and all, who indirectly or directly have given us their hand in this challenging task. We feel happy and joyful and content in expressing our vote of thanks to all those who have helped us and guided us in presenting this project work for our Minor project. Last, but never least, we thank our well-wishers and parents for always being with us, in every sense and constantly supporting us in every possible sense whenever possible.

Jatin	Kanesh	Rahul Ranjan
(2K21/EC/107)	(2K21/EC/110)	(2K21/EC/173)

CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1: INTRODUCTION

1.1 Overview

In today's digital era, information is easily available and it spreads rapidly, which changes how the people get knowledge about events on a global basis and scenes. But this fast information flow also has generated a worrying trend which is the common spreading of fake news, which is planned to effect public opinion, create unrest, or political gain or money gain, fake news is intentionally false or false information passed off saying honest journalism. With the rise of the online content, social media, and messaging apps, such false information can now spread across millions of users within minutes, so reaching millions before it can be exposed.

Fake news is bad because not just its does not pass as honest news but its tendency to support the individual biases. The customization of the data to preference of the user, algorithms that give content becomes echo chambers where the misinformation succeeds. The scene has been created in way where the people struggle or are not able to identify which news is real or which news is fake.

The social media make it tough in this issue as it is mostly uncontrolled. These sites give people to instantly share the information, but they also open up chances for bots, trolls, or political propagandists to spread misinformation on a big scale. There are some traditional fact-checking methods but they are manual, labour intensive, and not suitable for real-time answers.

Machine learning (ML) has risen as a hopeful solution to this issue. The ML models can recognize the patterns in large number of quantities of text and group material telling the likelihood of news being false. Fake news detection systems tend to bring speed and consistency to tackle the spreading of misinformation by combining various methods like natural language processing (NLP), sentiment analysis, and classification methods.

1.2 Background and Motivation

The motivation to create a fake news detection system came from seeing how serious the effects of misinformation can be in the real world. In recent years, fake news has contributed to political unrest, economic problems, and even outbreaks of violence. False stories can influence elections, stir up fear, and damage reputations—showing just how deeply they can impact democratic systems and society.

The COVID-19 pandemic was a clear example of how harmful fake news can be. Misinformation about vaccines, supposed cures, and the virus itself often spread faster than verified information from scientists. In such critical times, the need for smart tools that can identify or block misleading content becomes very clear.

Social media plays a big role in how easily false information spreads. People often share headlines without checking the source or the full context, especially if it fits their beliefs. This makes it urgent to develop intelligent systems that can help control the spread of misinformation and encourage more responsible sharing online. This natural human tendency, combined with the algorithmic bias of recommendation systems, enables fake news to flourish undetected.

While traditional fact-checking groups are trustworthy, they simply can't keep up with the massive amount of content being posted online every day. That's where machine learning can really make a difference—it allows for real-time analysis, automatic detection, and the possibility to support content moderation systems. By training models on large amounts of data, machine learning can pick up on subtle signs in writing—like tone, unusual patterns, or emotional bias—that humans might miss.

At last, this project is motivated by the idea that technology shouldn't just be used to spread information, but also to protect its truth. Building a system that can catch fake news early wouldn't just help readers and journalists—it would also support a more trustworthy and informed online world.

1.3 Goals and Objectives

The goal of this project is to build a comparative fake news detection system that categorizes news articles as real or fake by checking both the classical machine learning models and transformer-based deep learning models. The system applies natural language processing (NLP) and classification techniques to address the growing problem of misinformation on digital platforms using real-world datasets.

Objectives:

- To compile and evaluate a labeled dataset of authentic and fake news articles for model training and evaluation.
- To preprocess the data, including tokenization, stopword removal, and TF-IDF vectorization, to prepare data for machine learning models.
- To train and execute machine learning models such as Logistic Regression, Naive Bayes, or Decision Trees on the preprocessed data for baseline performance.
- To evaluate model performance using metrics like accuracy, precision, recall, and F1-score for a comprehensive evaluation.
- To identify limitations such as overfitting, misclassification, and data imbalance, and suggest improvements.
- To deploy the most effective model using a Flask API and Chrome Extension for real-time, user-friendly fake news classification.

This project provides a functional prototype for fake news detection and sets the stage for future advancements in AI-driven misinformation management.

CHAPTER 2: LITERATURE REVIEW

2.1 Introduction

A huge and various number of studies on fake news detection underlines the complexity and many aspects of the issue. The usage of digital media for data consumption has led to a considerable increase in the distribution of disinformation, requiring advanced systems for its detection and deletion.

Using of both the common statistical approaches and advanced machine learning tools, researchers have tackled the question of analysing and categorizing false content from many perspectives. The article describes major studies and developments in the area of developing methods for spotting fake news. The change from rule-based systems to more complex hybrid and multimodal models integrating visual and textual data is still ongoing in current research.

Some of case studies show how well these methods tackle practical problems, therefore underlining the need of machine learning in handling the continuously changing strategies used in the generation and spread of fake information.

Significant research has been done using computational techniques to find and eliminate fake knowledge in reaction to the increasing exposure to false news on digital media. Among the most helpful solutions are machine learning-based techniques having different levels of complexity accuracy, complexity, and context sensitivity. Given here are four major research in the field of fake news identification together with their techniques, results, and applications for the present work.

2.2 Case Studies on Fake News Detection

1. "Fake News Detection on Social Media: A Data Mining Perspective" – Shu et al. (2017)

Shu et al. conducted a comprehensive review of fake news detection methods, pointing out the limitations of traditional models when applied to the evolving landscape of social media. They proposed a framework that integrates content-based signals—such as writing style and linguistic features—with contextual indicators, including user engagement patterns, how information spreads through networks, and timing-based behaviours that reveal misinformation dynamics.

Rather than offering a single solution, the study outlines multiple strategies that extract features from both content and social interactions, employing classifiers like SVMs, RNNs, and credibility propagation models. It also explores deeper social factors behind misinformation, such as the role of echo chambers, bot networks, and coordinated manipulation. The work encourages future research that blends psychological insights, advanced algorithms, and scalable real-time systems to counter misinformation effectively.

2. “LIAR: A Benchmark Dataset for Fake News Detection” – William Y. Wang (2017)

Wang introduced the LIAR dataset as a key resource for fine-grained fake news detection. It includes 12,836 short political statements, each fact-checked and labeled into one of six credibility levels, such as “true,” “half-true,” and “pants on fire.” This dataset helps researchers test models based on both language patterns and varying degrees of truth.

The study tested different machine learning and deep learning methods, including Support Vector Machines, Logistic Regression, and LSTM networks. It showed how deep learning models are especially good at recognizing context and patterns in text. However, they also require more data and processing power. Overall, this work provided both a strong dataset and useful guidance on using AI to evaluate political claims.

3. “CSI: A Hybrid Deep Model for Fake News Detection” – Ruchansky et al. (2017)

Ruchansky and his team introduced the CSI models (Capture, Score, and Integrate) to detect fake news using a hybrid strategy. Unlike methods that focus only on content, CSI brings together source credibility, user engagement data (like shares and likes), and textual analysis using RNNs. The model highlights that patterns in user behaviour often reveal misinformation more reliably than content alone.

By examining both the credibility of sources and how false news spreads online, CSI achieved strong performance. Its integration of social context made it one of the more well-rounded frameworks, shaping future research that looks at both user activity and content features.

4. “FakeNewsNet: A Data Repository with News Content, Social Context and Spatiotemporal Information for Studying Fake News on Social Media” – Shu et al. (2020)

In their study, Shu et al. developed FakeNewsnet, a research-focused dataset designed to support deeper understanding of fake news on social media. What sets this dataset apart is its combination of three key elements: the full news article, the social context around how people interact with it (likes, shares, comments), and time-based patterns showing how stories spread. It pulls content from two well-known fact-checking sites PolitiFact and GossipCop providing a reliable mix of both fake and real news. The authors also built example models to show how including social data alongside the article content can improve fake news detection.

Rather than introducing a new model, the paper’s strength lies in the quality and structure of the data it offers, giving researchers a practical tool to explore not just what fake news says, but how it moves through social media—making it a valuable resource in today’s fight against misinformation.

2.3 Comparison with Existing Approaches

1. Compared to FakeNewsNet (Shu et al., 2018):

FakeNewsNet offers a rich dataset with content, social, and spatiotemporal features but lacks a detection model or practical tool.

Our project uses similar multi-dimensional features and compares advanced machine learning models, focusing on a usable system more practical and deployable than FakeNewsNet's static research resource.

2. Compared to LIAR Dataset (Wang, 2017):

LIAR provides short political statements with labels but lacks full-length news, social context, and practical application.

Our project uses full articles and incorporates richer features, going beyond benchmarking to offer a working solution.

3. Compared to CSI (Ruchansky et al., 2017):

CSI uses a hybrid model combining content and user behavior but has a fixed design and no usable tool.

Our project experiments with multiple models, including similar hybrid logic, while also building an interactive system or API, making it more flexible and application-ready.

4. Compared to FakeNewsNet (Shu et al., 2020):

Their work introduced a rich dataset combining news content and social context but relied on conventional models.

Our project builds on this foundation by applying advanced transformers to the same problem, demonstrating how recent innovations in NLP can improve fake news detection through deeper language understanding.

2.4 Contributions and Improvements over Existing Methods

Our project bridges the crucial gap between academic research and real-world application. While existing works focus on datasets, theoretical models, or isolated components, we integrate real, verified data, cutting-edge machine learning techniques, thorough model evaluation, and practical implementation. By developing a functional system or API that can be used beyond research settings, we ensure both technical robustness and usability. This makes our project not only academically valuable but also highly relevant for real-world deployment, public awareness, and combating misinformation at scale. It is a more complete, practical, and impactful contribution compared to the previous works.

Table 2.1 Comparative Study

Paper Title	Methodology	Dataset Used	Key Takeaway	How It Helped Our Project	Why Our Project Improves Upon It
Fake News Detection using TF-IDF & SVM	Used TF-IDF for feature extraction with SVM classifier	LIAR Dataset	Demonstrated that even simple ML models can detect fake news decently	Helped us build strong traditional model baselines	We applied the same traditional models but also compared them with a transformer (BERT) to measure real improvement
Deep Learning Approach to Fake News Detection	Applied LSTM with pre-trained embeddings (like GloVe)	Kaggle Fake News Dataset	Showed deep learning could outperform ML models on complex patterns	Motivated us to explore beyond classical methods	Our BERT-based model understands the meaning of words based on their context, which helps it perform better than older methods like GloVe that treat each word the same no matter where it appears.
Transformer Models for Fake News Detection	Fine-tuned BERT and compared with traditional methods	FakeNewsNet	Proved that transformer models significantly boost accuracy	Inspired us to experiment with BERT as well	We not only fine-tuned BERT but also conducted comparative analysis with multiple classical models on two datasets
Detecting Fake News on GossipCop & PolitiFact	Used BERT on these two real-world datasets	GossipCop & PolitiFact	Established the value of domain-diverse datasets	Led us to select the same datasets for consistency and relevance	We extended their idea by integrating a comparative study with traditional models and analyzing challenges/future scope deeply

CHAPTER 3: METHODOLOGY

3.1 Dataset Description

The project used a combination of the GossipCop and Politifact datasets, obtained via Kaggle. These datasets contain approximately 40,000 labelled news entries categorized as either "real" or "fake." Each record typically includes:

- News Title
- News URL
- Tweet IDs (metadata)
- Label: Fake or Real

The datasets were downloaded in .csv format and loaded into the environment using Python's Pandas library. To ensure class balance, an equal number of real and fake samples were selected during preprocessing.

3.2 Data Preprocessing Techniques

Preprocessing of unstructured text was critical for enhancing model performance. The following preprocessing steps were applied:

- Lowercasing: All text was converted to lowercase to ensure consistency and avoid duplication (e.g., “India” vs. “india”).
- URL and Symbol Removal: URLs and non-alphabetic characters were removed using regular expressions to clean the text.
- Whitespace Normalization: Extra spaces and tabs were reduced to a single space.
- Tokenization: Sentences were broken into single words using NLTK's word_tokenize().
- Stopword Removal: Commonly used but less meaningful words (e.g., "is", "the", "on") were removed using NLTK's built-in stopword list.
- Lemmatization: Words were reduced to their base forms using WordNet Lemmatizer (e.g., “running” → “run”).

```
from nltk.corpus import stopwords

from nltk.stem import WordNetLemmatizer
```

This step produced clean and structured data suitable for feeding into both machine learning and deep learning models.

3.3 Feature Extraction Methods

To enable model training, the textual data was converted into numerical form:

- TF-IDF (Term Frequency-Inverse Document Frequency): Used to extract features for traditional machine learning models. It highlights rare but informative words while down-weighting frequent terms.

from sklearn.feature_extraction.text import TfidfVectorizer

This representation was used to build sparse matrices which were fed into classical models.

For BERT, text inputs were tokenized using Huggingface's BERT Tokenizer, which transforms text into token IDs suitable for transformer-based architectures.

3.4 Model Selection

The system experimented with both classical and deep learning-based models to determine the most effective approach:

- Multinomial Naive Bayes
- Logistic Regression
- Support Vector Machine (SVM)
- Random Forest
- Kneighbors
- Decision Tree

The choice of models was evaluated using the following metrics: Accuracy, F1 score, Precision, Recall.

3.4.1 Transformer-Based Approach (BERT)

- In addition to classical models, we experimented with a transformer-based model using BERT (Bidirectional Encoder Representations from Transformers) to enhance prediction quality.
- Motivation: Classical models were limited in taking contextual relationships in text. BERT, being pre-trained on a massive amount, understands language more deeply.
- Implementation:
  - Used a pre-trained BERT model via Hugging Face's transformers library.
  - Fine-tuned it on our fake news dataset.
  - Preprocessing included tokenization using BertTokenizer and input formatting compatible with transformer models.
- Constraints:
  - We used epoch=2 for fine-tuning due to time and hardware limitations.
  - More powerful models like BERT-v5 or v6 required GPU setups that were not available in our environment.

- Result: BERT improved classification confidence and contextual understanding, especially for unclear headlines. However, due to environment limitations, performance gains were moderate.

3.5 Model Training Procedures

The processed dataset was split into:

- Training Set – 80%
- Testing Set – 20%

```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

Traditional models were trained using this split, with hyperparameters optimized using GridSearchCV where applicable.

For BERT, dynamic padding, attention masks, and batching were used with a training setup involving two epochs and a batch size of 8 using PyTorch. The final model was saved after training for deployment.

3.6 Model Evaluation Metrics

Each model's performance was assessed using:

- Accuracy– Overall correctness of predictions
- Precision - Proportion of predicted fake news that were actually fake
- Recall - Proportion of actual fake news that were correctly identified
- F1 Score - Harmonic mean of precision and recall

These metrics helped evaluate the system’s ability to correctly identify both real and fake news. The BERT-based model outperformed traditional methods across all evaluation criteria, offering improved precision and generalization.

3.7 Output Generation

The system accepts manual input of a news headline or short text and returns a classification result—either “FAKE” or “REAL”—along with a confidence score indicating the prediction certainty.

A Flask-based REST API handles incoming requests and responds with JSON-formatted outputs. A Chrome Extension was also developed, allowing users to highlight any text on a webpage and instantly receive the model’s prediction directly in the browser through alert pop-ups.

CHAPTER 4: IMPLEMENTATION

4.1 Tools and Technologies Used

For the development of the Fake News Detection System, the following tools and technologies were used:

- Programming Language: Python 3.x – Chosen for its simplicity, readability, and extensive machine learning and deep learning libraries.
- IDE/Code Editor: Jupyter Notebook / VS Code – Used to write, debug, and visualize the code interactively.
- Frameworks and Libraries:
    - Scikit-learn – For implementing classical machine learning models and performance evaluation.
    - Pandas – For data reading, cleaning, and manipulation.
    - NumPy – For efficient numerical operations.
    - NLTK (Natural Language Toolkit) – For performing preprocessing tasks like stopword removal and lemmatization.
    - Huggingface Transformers – For implementing and fine-tuning the BERT model.
    - Flask – Used to deploy the trained BERT model as a web API for real-time predictions.
- Dataset Sources:
    - GossipCop and Politifact datasets (collected via Kaggle) – Contain labeled fake and real news samples.
    - Data was downloaded in .csv format for further processing.

4.2 Libraries Used and Their Functionality

1. Pandas
    - Purpose: Handling and manipulating structured data.
    - Usage in Project: Reading .csv files, removing duplicates, merging datasets, and filtering records.
    - Reason: Efficiently processes large structured datasets with labeled fields.
2. NumPy
    - Purpose: Performing numerical and array-based operations.

    - Usage in Project: Used alongside Pandas and Scikit-learn for mathematical calculations.

3. Scikit-learn (sklearn)
    - Purpose: A widely used library for machine learning.
    - Usage in Project:
        - Converting text data into vectors using TF-IDF.
        - Training and testing models such as:
            - Naive Bayes
            - Logistic Regression
            - Support Vector Machine (SVM)
            - Decision Tree
            - Random Forest
            - K-Nearest Neighbors (KNN)
        - Model evaluation using metrics like precision, recall, F1-score, and accuracy.
4. NLTK (Natural Language Toolkit)
    - Purpose: Text preprocessing for NLP tasks.
    - Usage in Project:
        - Lowercasing
        - Tokenization
        - Removing stopwords
        - Lemmatization
    - These steps ensured clean and meaningful textual input for models.
5. Matplotlib / Seaborn
    - Purpose: Visualization of data and results.
    - Usage in Project: Displaying confusion matrices and comparing model performances.
6. Flask
    - Purpose: Backend web development using Python.

- Usage in Project: Hosts the trained BERT model as a REST API. It receives input text via HTTP POST and responds with a prediction.

7. Huggingface Transformers
    - Purpose: Pretrained models and tokenizers for NLP.
    - Usage in Project: Fine-tuned the BERT model for fake news classification tasks. Provided tools to tokenize input, set training arguments, and evaluate results.

4.3 Step-by-Step Implementation Process

1. **Data Collection**
    - Datasets were collected from Kaggle (GossipCop and Politifact).
    - Each dataset contains fields such as news title, label (fake/real), and associated metadata.
2. **Data Cleaning and Preprocessing**
    - Removed null values and duplicate entries.
    - Preprocessing steps included:
        - Lowercasing
        - URL and symbol removal
        - Tokenization
        - Stopword removal
        - Lemmatization
    - These steps ensured uniform and clean data for the models.
3. **Feature Extraction**
    - For traditional models, TF-IDF vectorization was used to convert text into numeric feature matrices.
    - For BERT, Huggingface's tokenizer was used to convert text into token IDs with attention masks.
4. **Model Selection and Training**
    - The following models were used:
        - Logistic Regression
        - Naive Bayes
        - Support Vector Machine (SVM)
        - Decision Tree

        - Random Forest
        - K-Nearest Neighbors
        - BERT (Fine-tuned Transformer-based model)
    - Classical models were trained using TF-IDF features.
    - BERT was fine-tuned using PyTorch with training and evaluation splits.

5. **Model Evaluation**
    - Performance of each model was measured using:
        - Accuracy
            - Overall correctness of predictions
        - Precision
            - Proportion of predicted fake news that were actually fake
        - Recall
            - Proportion of actual fake news that were correctly identified
        - F1 Score
            - Harmonic mean of precision and recall
    - Confusion matrices were plotted to compare results and understand class-wise performance.

6. **Deployment**
    - The fine-tuned BERT model was deployed using a Flask API.
    - A Chrome Extension was created using Manifest V3. It allows users to select text on any webpage and verify if it is real or fake news.
    - The extension sends the selected text to the Flask server and displays the result using an alert popup in the browser.

CHAPTER 5: RESULTS AND DISCUSSION

5.1 Model Performance Evaluation

- Naive Bayes: Accuracy = 83.18%
- Logistic Regression: Accuracy = 84.42%
- Decision Tree: Accuracy = 76.22%
- SVM: Accuracy = 84.05%
- K-Nearest Neighbors: Accuracy = 73.43%
- Random Forest: Accuracy = 82.45%
- BERT (Fine-tuned Transformer Model): Accuracy = 85.31% (Highest accuracy overall)

5.2 Output

```
RAW OUTPUT: {'label': 'LABEL_0', 'score': 0.9456126093864441}
Headline: Schools to remain shut in Indian-administered Kashmir
Prediction: Fake (Confidence: 94.56%)

RAW OUTPUT: {'label': 'LABEL_1', 'score': 0.9642096757888794}
Headline: IPL cricket match stopped midway in Dharamshala
Prediction: Real (Confidence: 96.42%)

RAW OUTPUT: {'label': 'LABEL_1', 'score': 0.9761190414428711}
Headline: 5G towe are responsible for COVID-19 spread.
Prediction: Real (Confidence: 97.61%)

RAW OUTPUT: {'label': 'LABEL_0', 'score': 0.9890580773353577}
Headline: Obama secretly converted to Islam, White House insider claims.
Prediction: Fake (Confidence: 98.91%)

RAW OUTPUT: {'label': 'LABEL_1', 'score': 0.9949017763137817}
Headline: India's Banu Mushtaq makes history with International Booker win
Prediction: Real (Confidence: 99.49%)

RAW OUTPUT: {'label': 'LABEL_1', 'score': 0.7319616079330444}
Headline: Indian YouTuber arrested for allegedly 'spying' for Pakistan
Prediction: Real (Confidence: 73.2%)

RAW OUTPUT: {'label': 'LABEL_1', 'score': 0.9670774936676025}
Headline: WHO declares end to COVID-19 global emergency.
Prediction: Real (Confidence: 96.71%)

RAW OUTPUT: {'label': 'LABEL_1', 'score': 0.9959573149681091}
Headline: Elon Musk's SpaceX completes another successful Starlink mission.
Prediction: Real (Confidence: 99.6%)
```

Fig.5.1 Output of BERT model

```
Logistic Regression
Accuracy: 0.8442
              precision    recall  f1-score   support

           0       0.81      0.43      0.56      1016
           1       0.85      0.97      0.91      3354

    accuracy                           0.84      4370
   macro avg       0.83      0.70      0.73      4370
weighted avg       0.84      0.84      0.83      4370

------------------------------------------------------------
Naive Bayes
Accuracy: 0.8318
              precision    recall  f1-score   support

           0       0.80      0.37      0.51      1016
           1       0.84      0.97      0.90      3354

    accuracy                           0.83      4370
   macro avg       0.82      0.67      0.70      4370
weighted avg       0.83      0.83      0.81      4370

------------------------------------------------------------
Random Forest
Accuracy: 0.8245
              precision    recall  f1-score   support

           0       0.71      0.41      0.52      1016
           1       0.84      0.95      0.89      3354

    accuracy                           0.82      4370
   macro avg       0.78      0.68      0.71      4370
weighted avg       0.81      0.82      0.81      4370

Decision Tree
Accuracy: 0.7622
              precision    recall  f1-score   support

           0       0.49      0.49      0.49      1016
           1       0.85      0.84      0.85      3354

    accuracy                           0.76      4370
   macro avg       0.67      0.67      0.67      4370
weighted avg       0.76      0.76      0.76      4370

------------------------------------------------------------
Support Vector Machine
Accuracy: 0.8405
              precision    recall  f1-score   support

           0       0.80      0.42      0.55      1016
           1       0.85      0.97      0.90      3354

    accuracy                           0.84      4370
   macro avg       0.82      0.69      0.73      4370
weighted avg       0.84      0.84      0.82      4370

------------------------------------------------------------
KNN
Accuracy: 0.7343
              precision    recall  f1-score   support

           0       0.39      0.24      0.30      1016
           1       0.79      0.88      0.84      3354

    accuracy                           0.73      4370
   macro avg       0.59      0.56      0.57      4370
weighted avg       0.70      0.73      0.71      4370

------------------------------------------------------------
```

Fig.5.2 Accuracy of ML models

```
✅ Accuracy Score: 85.31 %

✅ Classification Report:
              precision    recall  f1-score   support

        Fake       0.74      0.60      0.66      1044
        Real       0.88      0.93      0.91      3326

    accuracy                           0.85      4370
   macro avg       0.81      0.77      0.78      4370
weighted avg       0.85      0.85      0.85      4370
```

Fig.5.3 Accuracy of BERT model

Model	PolitiFact				GossipCop			
	Accuracy	Precision	Recall	F1	Accuracy	Precision	Recall	F1
SVM	0.580	0.611	0.717	0.659	0.497	0.511	0.713	0.595
Logistic regression	0.642	0.757	0.543	0.633	0.648	0.675	0.619	0.646
Naive Bayes	0.617	0.674	0.630	0.651	0.624	0.631	0.669	0.649
CNN	0.629	0.807	0.456	0.583	0.723	0.751	0.701	0.725
Social Article Fusion /S	0.654	0.600	0.789	0.681	0.689	0.671	0.738	0.703
Social Article Fusion /A	0.667	0.667	0.579	0.619	0.635	0.589	0.882	0.706
Social Article Fusion	0.691	0.638	0.789	0.706	0.689	0.656	0.792	0.717

Fig.5.4 Accuracy from prior work on the same dataset

5.3 Analysis of Results

- The fine-tuned BERT model achieved the highest accuracy among all models, showcasing its ability to capture deep contextual meaning and subtle linguistic patterns in the text. Its use of bidirectional attention allowed it to better understand sentence structure and nuanced word usage, which is especially important in detecting the misleading or manipulative language in fake news.
- Logistic Regression, Naive Bayes, and SVM also showed strong performance, with over 83% accuracy after preprocessing improvements and balanced training. These models benefited from feature engineering and text vectorization techniques like TF-IDF, which helped highlight the most informative terms.
- The Random Forest model performed better than the Decision Tree and KNN due to its collective nature and resistance to overfitting. Its use of multiple decision trees helped capture different patterns within the data, making it stronger to noise and outliers.
- While BERT required more training time and computational resources, it delivered the most consistent and accurate predictions across both real and fake news classes. This made it particularly valuable in applications where precision and reliability are critical.

- Classical models were faster to train and easier to deploy but had slightly lower precision and generalization compared to the transformer-based approach. However, they still present a practical option for systems with limited computational capacity.

5.4 Limitations

- The BERT-based model, while accurate, is computationally intensive and not ideal for low-resource or real-time applications.
- Some model predictions are overconfident and incorrect, highlighting the need for better interpretability.
- The system focuses only on textual content and does not consider images, videos, or social media context—key elements in real-world misinformation.
- The dataset is limited to English-language news, mostly from specific domains, which may reduce generalizability to other languages or misinformation types.

5.5 Future Scope

- Multimodal Detection: In the future, we aim to analyse not just the text, but also images, videos, and other media linked with the news. This would help the system perform better on platforms like social media, where misinformation often includes visuals.
- Support for Multiple Languages: Adding regional and non-English language support especially in a diverse country like India could greatly expand the system's usability and impact.
- Explanation of the predictions: Making the model's predictions more transparent would help build trust. We would like users specially journalists or fact-checkers to understand why a news piece is flagged.

5.6 Challenges and Error Analysis

During the development process, several technical and implementation-related issues were encountered. These challenges and the corresponding solutions are listed below:

1. Data Imbalance
    - Issue: The dataset initially had a higher number of real news samples compared to fake ones.
    - Effect: This caused models to favour the real class, resulting in poor recall for fake news.

- Fix: Applied train-test splitting and used balanced sampling to equalize class representation.

2. Overfitting

   - Issue: Models trained on TF-IDF features, especially Decision Trees, showed excellent performance on training data but failed to generalize.
   - Symptoms: High training accuracy (~97%) but low test accuracy (~78%).
   - Fix: Reduced the number of TF-IDF features, implemented cross-validation, and applied regularization techniques like tuning the C parameter in Logistic Regression.

3. Preprocessing Errors

   - Issue: Handling of contracted words and special characters caused inconsistent results during model input generation.
   - Fix: Used re.sub() for consistent text cleaning, followed by lemmatization using WordNet for better base word representation.

4. Memory Errors with SVM

   - Issue: Training SVM on the complete TF-IDF matrix for large datasets caused memory overflow errors.
   - Fix: Reduced the TF-IDF feature size using max_features=5000 and adjusted system memory settings to allow more efficient computation.

5. Deep Learning Training Constraints

   - Issue: Initial prediction accuracy from the BERT model was lower due to limited training cycles.
   - Fix: Increased the number of training epochs to 2 for better learning.
   - Additional Challenge: Attempted to use larger BERT variants (v5/v6), but these required a GPU environment that was not available in the current setup.
   - Solution: Continued with bert-base-uncased and optimized training within available CPU resources.

CHAPTER 6: CONCLUSION

The spread of digital media and the ease of sharing information have made fake news one of the most demanding challenges in today's interconnected world. The effects of misinformation are extensive, influencing public opinion, affecting policy decisions, and, at times, provoking social unrest. In this context, the development of an intelligent and automated Fake News Detection System is not only relevant but essential. This project aimed to design such a system by leveraging the power of Natural Language Processing (NLP), traditional Machine Learning (ML) algorithms, and advanced deep learning techniques, including BERT-based transformers.

We followed a comparative modeling strategy, starting with traditional machine learning models like Naive Bayes, Logistic Regression, SVM, Decision Tree, Random Forest, and KNN. These were tested on two well-known benchmark datasets—GossipCop and Politifact. Before training, the data was carefully preprocessed: text was lowercased, punctuation removed, stopwords filtered out, and lemmatization applied. TF-IDF vectorization was then used to turn the cleaned text into structured input features.

We saw notable improvements when BERT (Bidirectional Encoder Representations from Transformers) was introduced. Fine-tuned using Huggingface Transformers and implemented in PyTorch, BERT reached around 85.31% accuracy—outperforming all classical models in metrics like precision, recall, and F1-score. Its ability to understand context made it especially effective at detecting subtle or ambiguous cases of fake news.

In summary, this project presents a solid fake news detection system that blends traditional machine learning with transformer-based models. With real-time predictions and an easy-to-use interface, it offers a practical solution and lays the groundwork for future features like multi-language support, credibility scoring, and social media integration. As misinformation tactics grow more sophisticated, so too must our detection tools and this project marks meaningful progress in that direction.

CHAPTER 7: BIBLIOGRAPHY

[1] Kai Shu, Amy Sliva, Suhang Wang, Jiliang Tang, Huan Liu, “Fake News Detection on Social Media: A Data Mining Perspective,” ACM SIGKDD Explor. Newsl., vol. 19, no. 1, pp. 22–36, 2017.

[2] William Yang Wang, “LIAR: A Benchmark Dataset for Fake News Detection,” in Proc. 55th Annu. Meeting Assoc. Comput. Linguistics (ACL), 2017, pp. 422–426.

[3] Natali Ruchansky, Sungyong Seo, Yan Liu, “CSI: A Hybrid Deep Model for Fake News Detection,” in Proc. 26th ACM Int. Conf. Inf. Knowl. Manag. (CIKM), 2017, pp. 797–806.

[4] Kai Shu, Deepak Mahudeswaran, Suhang Wang, Dongwon Lee, Huan Liu “FakeNewsNet: A data repository with news content, social context and spatiotemporal information for studying fake news on social media,” Big Data, vol. 8, no. 3, pp. 171–188, Sep. 2020, doi: 10.1089/big.2020.0062.

[5] Flask Documentation. [Online]. Available: https://flask.palletsprojects.com/

[6] Scikit-learn Documentation. [Online]. Available: https://scikit-learn.org/stable/

[7] Chrome Developers, “Chrome Extension Developer Guide,” [Online]. Available: https://developer.chrome.com/docs/extensions/

A Z

Prozect 4-3.docx

Delhi Technological University

Document Details

Submission ID

trn:oid:::27535:97171298

Submission Date

May 22, 2025, 2:07 PM GMT+5:30

Download Date

May 22, 2025, 2:12 PM GMT+5:30

File Name

Prozect 4-3.docx

File Size

223.8 KB

17 Pages

4,285 Words

25,970 Characters

12% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Filtered from the Report

- Bibliography
- Quoted Text
- Cited Text
- Small Matches (less than 8 words)

Match Groups

44 Not Cited or Quoted 12%
Matches with neither in-text citation nor quotation marks

0 Missing Quotations 0%
Matches that are still very similar to source material

0 Missing Citation 0%
Matches that have quotation marks, but no in-text citation

0 Cited and Quoted 0%
Matches with in-text citation present, but no quotation marks

Top Sources

7% Internet sources
5% Publications
10% Submitted works (Student Papers)

Integrity Flags

0 Integrity Flags for Review

No suspicious text manipulations found.

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

Match Groups

- **44** Not Cited or Quoted 12%
  Matches with neither in-text citation nor quotation marks
- **0** Missing Quotations 0%
  Matches that are still very similar to source material
- **0** Missing Citation 0%
  Matches that have quotation marks, but no in-text citation
- **0** Cited and Quoted 0%
  Matches with in-text citation present, but no quotation marks

Top Sources

7% Internet sources
5% Publications
10% Submitted works (Student Papers)

Top Sources

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	Submitted works	JIS University on 2025-03-07	1%
2	Internet	download.bibis.ir	<1%
3	Internet	businessdocbox.com	<1%
4	Submitted works	University of Westminster on 2024-09-02	<1%
5	Internet	www2.mdpi.com	<1%
6	Submitted works	Technological Institute of the Philippines on 2025-05-16	<1%
7	Submitted works	University of Hertfordshire on 2025-04-28	<1%
8	Internet	www.coursehero.com	<1%
9	Submitted works	BB9.1 PROD on 2025-04-22	<1%
10	Submitted works	Victoria University on 2023-06-30	<1%

11	Internet	
www.researchgate.net		<1%
12	Internet	
aclanthology.org		<1%
13	Submitted works	
Napier University on 2023-08-16		<1%
14	Internet	
www.jpost.com		<1%
15	Submitted works	
Middlesex University on 2025-04-15		<1%
16	Publication	
Yuelin Li, Zhenjia Fan, Xiaojun Yuan, Xiu Zhang. "Recognizing fake information th...		<1%
17	Internet	
nlistsp.inflibnet.ac.in		<1%
18	Submitted works	
Asia Pacific Instutute of Information Technology on 2024-10-21		<1%
19	Submitted works	
Rochester Institute of Technology on 2016-12-06		<1%
20	Submitted works	
The University of Manchester on 2024-09-16		<1%
21	Submitted works	
University of Bradford on 2025-01-06		<1%
22	Submitted works	
University of Southampton on 2018-09-03		<1%
23	Internet	
acp.copernicus.org		<1%
24	Internet	
ebin.pub		<1%

25 Internet

irjet.net <1%

26 Internet

medium.com <1%

27 Internet

www.mdpi.com <1%

28 Publication

"Advances in Information Retrieval", Springer Science and Business Media LLC, 20... <1%

29 Publication

Natasa Kleanthous, Abir Hussain. "Machine Learning in Farm Animal Behavior usi... <1%

30 Publication

T. Mariprasath, Kumar Reddy Cheepati, Marco Rivera. "Practical Guide to Machin... <1%

31 Submitted works

Tilburg University on 2025-05-16 <1%

32 Submitted works

University of Greenwich on 2020-05-21 <1%

33 Submitted works

University of Houston, Downtown on 2025-04-17 <1%

34 Internet

ktisis.cut.ac.cy <1%

35 Submitted works

Icba on 2025-05-19 <1%

36 Submitted works

uaveiro on 2024-10-13 <1%

37 Internet

www.americaspg.com <1%

38 Internet

www.nature.com <1%