Fake News Detection Using Machine Learning

Project report submitted in partial fulfilment of the requirements for the degree of

Bachelor of Technology

in

Computer Science and Engineering

by

Kavish Jain (23UCS614)
Raghav Malani (23UCC590)

Under the Guidance of

Dr. Nirmal Sivaraman

The LNM Institute of
Information Technology

Department of Computer and Communications Engineering
The LNM Institute of Information Technology, Jaipur

September 2025

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to Dr. Nirmal Sivaraman. His continuous guidance and encouragement were a lifeline throughout this project. More than just helpful, his thoughtful feedback has been fundamental to my learning and growth.

My thanks also go to my college and professors. They have fostered an incredible environment that allowed me to explore and develop this work truly.

And to my family and friends: thank you from the bottom of my heart. Your patience, your motivation, and your unwavering belief in me made all the difference during the challenging times.

This project is built on the support I received from everyone, and I am sincerely thankful to all who contributed.

ABSTRACT

Fake news has emerged as a critical challenge in today's digital ecosystem, where information spreads rapidly across online news platforms and social media. The absence of reliable verification mechanisms often exposes users to misleading or fabricated content. This project presents a machine learning–based system designed to classify textual content as real or fake, using both long-form news articles and short social media posts. The objective is to evaluate how model performance varies across domains and how feature engineering influences classification accuracy.

The system first replicates the methodology described in a published research paper on fake news detection using classical machine learning ensemble models. By training on a news article dataset with TF–IDF and baseline linguistic features, the system establishes strong benchmark performance. In the second phase, the same models and feature set are applied to a manually collected dataset of one-line social media posts, which introduces challenges such as limited context, informal writing, and noisier language.

To address these domain-specific issues, the project incorporates an enhanced set of linguistic, stylistic, and sentiment-based features inspired by another research paper focused on fake news detection on social media. These additional features—covering lexical diversity, punctuation behavior, word-length patterns, character-level statistics, and a sentiment polarity score for each post—significantly improve the system's ability to separate real and fake content in short posts. Ultimately, this work demonstrates how machine learning techniques, combined with thoughtful feature engineering, can adapt to diverse text formats and improve automated fake news detection across different online platforms.

Contents

1 Introduction

1.1 Background

The rapid growth of digital information channels has led to an increased dependency on online news and social media platforms for real-time updates. However, the ease of publishing without verification has resulted in a surge of misleading and fabricated content—commonly known as fake news. Unlike traditional news articles, which are relatively structured and lengthy, social media posts are short, informal, and often lack context. This makes fake news detection a challenging problem across different text domains. The research paper used in this project highlights how machine learning models, especially ensemble methods, can classify fake news effectively using textual features. Building on this, our work extends fake news detection to cross-domain data, including long articles and short one-line social media posts.

1.2 Problem Definition

Manual detection of fake news is neither scalable nor reliable due to the large volume of content generated every minute. In addition, fake news on social media spreads rapidly through shares, reposts, and viral trends. This project aims to develop and analyze automated machine learning models that classify text as real or fake across two different content types. Specifically, the problem is defined as follows:

- Implementing the machine learning models and feature extraction approach described in the research paper on a structured news article dataset.
- Applying the same models and original feature set to a manually collected dataset of one-line social media posts, to test cross-domain generalization.

1.3 Need for Automated Fake News Detection

Automating the detection of fake news provides several advantages:

- Reduces human verification effort, which is impractical at large scales.
- Handles high-velocity data streams, such as social media posts.
- Identifies misleading content early, preventing rapid viral spread.
- Improves reliability by combining machine learning models with relevant linguistic features.

The research clearly shows that effective fake news detection requires not only strong models but also contextually appropriate features, especially for short, noisy text.

1.4 Scope of the Project

This project involves three major stages, aligned with the research papers and the implemented Python scripts:

- **Stage 1:** Implementing the methodology from the research paper using machine learning ensemble models on the structured news article dataset.
- **Stage 2:** Applying the same feature extraction and models to the manually collected short social media posts, to evaluate performance differences due to domain shift.
- **Stage 3:** Enhancing the feature set using linguistic, stylistic, and sentiment-based features proposed in the second research paper, and implementing them to improve detection accuracy on social media posts.
- **Final Comparison:** Analyzing and comparing results across all three stages to understand how different features and data characteristics influence fake news classification quality.

2 Literature Review

2.1 Research Paper 1: Fake News Detection Using ML Ensemble Models

The first research paper focuses on detecting fake news using classical machine learning algorithms combined with TF–IDF and basic linguistic features. The paper evaluates multiple models such as Logistic Regression, SVM, Random Forest, Naïve Bayes, and several ensemble methods. Among these, ensemble models such as Bagging, AdaBoost, Gradient Boosting, and Voting Classifier demonstrated higher accuracy due to their ability to combine decisions from multiple weak learners. The methodology includes preprocessing, TF–IDF vectorization, feature extraction, and training multiple classifiers to identify the best-performing model for long-form news articles.

2.2 Research Paper 2: Fake News Detection on Social Media

The second research paper studies fake news detection specifically on short social media posts, which differ significantly from full-length news articles. Since social media posts are brief, informal, and often lacking in context, the paper proposes enhanced feature engineering techniques. These features include punctuation-based metrics, lexical diversity, uppercase ratio, average word length, digit ratio, and character-level statistics. In addition, our implementation extends this idea by incorporating a sentiment polarity score for each post, motivated by work that links emotional tone to deceptive or sensational content on social media.

2.3 Comparison of Approaches

The first research paper deals with structured, long-form news articles, where TF–IDF combined with classical machine learning models performs effectively. In contrast, the second research paper targets short-form social media text, which requires additional fine-grained linguistic and stylistic features to compensate for limited context. Thus, while both papers use traditional machine learning approaches, their feature engineering strategies differ significantly due to differences in text length, structure, and writing style.

2.4 Key Observations

- Ensemble machine learning models generally outperform single classifiers on well-structured article datasets.
- Social media posts require more detailed linguistic and stylistic features due to their informal and context-limited nature.
- TF–IDF alone is insufficient for very short text samples, making additional feature engineering, including sentiment information, essential.

3 Dataset Description

3.1 Research Paper Dataset (News Articles - ISOT)

The research paper dataset contains two labeled categories:

- Real news
- Fake news

These articles are long, structured, and well-written, making them suitable for TF–IDF and classical machine learning models.

3.2 Manually Collected Dataset

This dataset consists of one-line social media posts collected manually. The posts are:

- One-liner statements
- Informal in writing style
- Noisy and less structured

3.3 Data Labels and Preprocessing

Basic preprocessing steps were applied to both datasets:

- Normalization of text and label columns
- Removal of duplicates and empty entries
- Minimal cleaning to preserve natural writing patterns

3.4 Challenges in Social Media Data

Short posts introduce several difficulties:

- Lack of context reduces model understanding
- Slang, abbreviations and emojis affect tokenization
- TF–IDF vectors become sparse due to low word count

4 Methodology

4.1 Stage 1: Implementing Research Paper Model

4.1.1 Dataset Preparation

The research paper provides two datasets: `True.csv` and `Fake.csv`. These were merged into a single labelled dataset, unnecessary symbols were removed, and text fields were normalized.

4.1.2 Feature Extraction

The original feature extraction approach from the research paper was implemented:

- TF–IDF vectorization (unigrams and bigrams)
- Basic linguistic features:
  - token count
  - average word length
  - punctuation ratio

4.1.3 Models Used

The following classical machine learning models were trained:

- Logistic Regression
- Linear SVM
- Random Forest
- AdaBoost, Gradient Boosting
- Voting Classifier (ensemble)

4.1.4 Training Procedure

An 80:20 train–test split was used. Models were evaluated using accuracy, F1-score, and confusion matrix to identify the best-performing classifier for long news articles.

4.2 Stage 2: Same Model on Social Media Posts

4.2.1 Using `all_posts.csv`

This dataset contains manually collected one-line social media style posts labelled as real or fake.

4.2.2 Preprocessing

Only light preprocessing was applied because the posts are short:

- lowercase conversion
- removal of empty or duplicate entries

4.2.3 Model Reuse

The same TF–IDF settings and linguistic features from Stage 1 were reused to test cross-domain performance on short posts.

4.2.4 Training Procedure

Models were trained using 5-fold cross-validation to ensure stable results, as the dataset is smaller and more variable compared to the news dataset.

4.3 Stage 3: Improved Features

4.3.1 Motivation

Stage 2 revealed that features designed for long articles do not generalize well to short one-line posts, leading to lower accuracy and sparse TF–IDF vectors.

4.3.2 Enhanced Feature Set

Based on the second research paper, additional linguistic, stylistic, and sentiment-based features were added:

- lexical diversity (type–token ratio)
- uppercase character ratio
- digit ratio
- punctuation count and ratio
- average word length
- sentiment polarity score for each post (range -1 to $+1$)

4.3.3 Model Training

The same set of classifiers from Stage 1 were trained again, but now using the combined TF–IDF and enhanced feature set to better capture writing patterns and emotional tone in short posts.

4.3.4 Feature Impact Analysis

The newly engineered features, including sentiment polarity, significantly improved model performance on social media posts, demonstrating the importance of domain-specific feature design when working with short, informal text.

5 System Architecture

5.1 Architecture Diagram

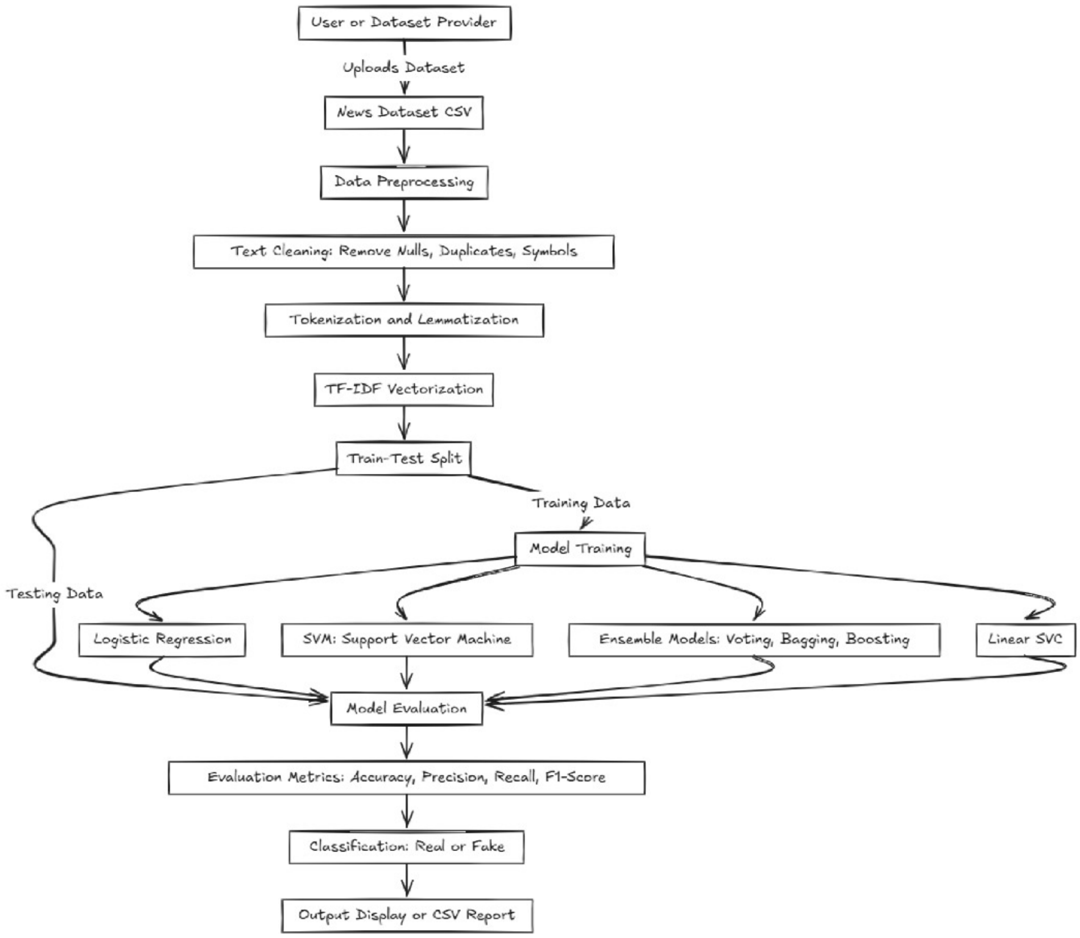

Figure 1: Overall System Architecture for Fake News Detection

5.2 Feature Engineering Pipeline

1. Tokenization
2. TF–IDF vectorization
3. Linguistic and stylistic feature extraction
4. Sentiment polarity computation (Stage 3)
5. Feature concatenation

5.3 Stage 1 Feature Extraction (Research Paper 1 – News Articles)

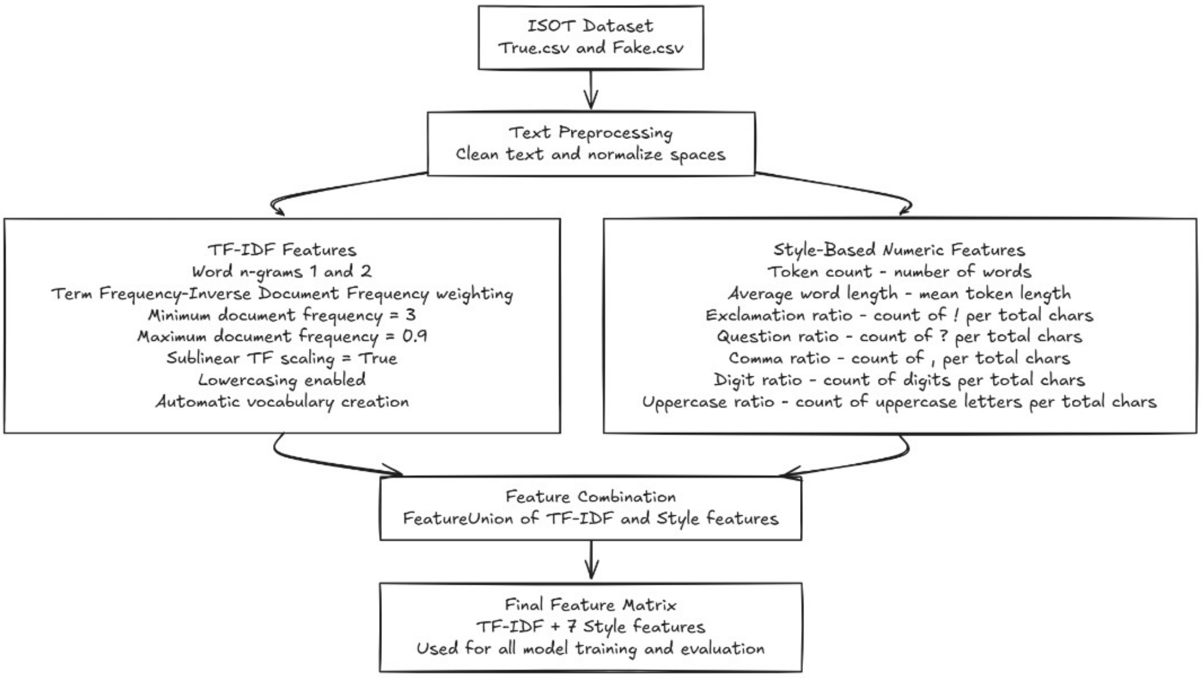

Figure 2: Feature extraction pipeline used in Stage 1 based on Research Paper 1, incorporating TF–IDF and basic linguistic features for news articles.

5.4 Stage 3 Feature Extraction (Research Paper 2 – Social Media Posts)

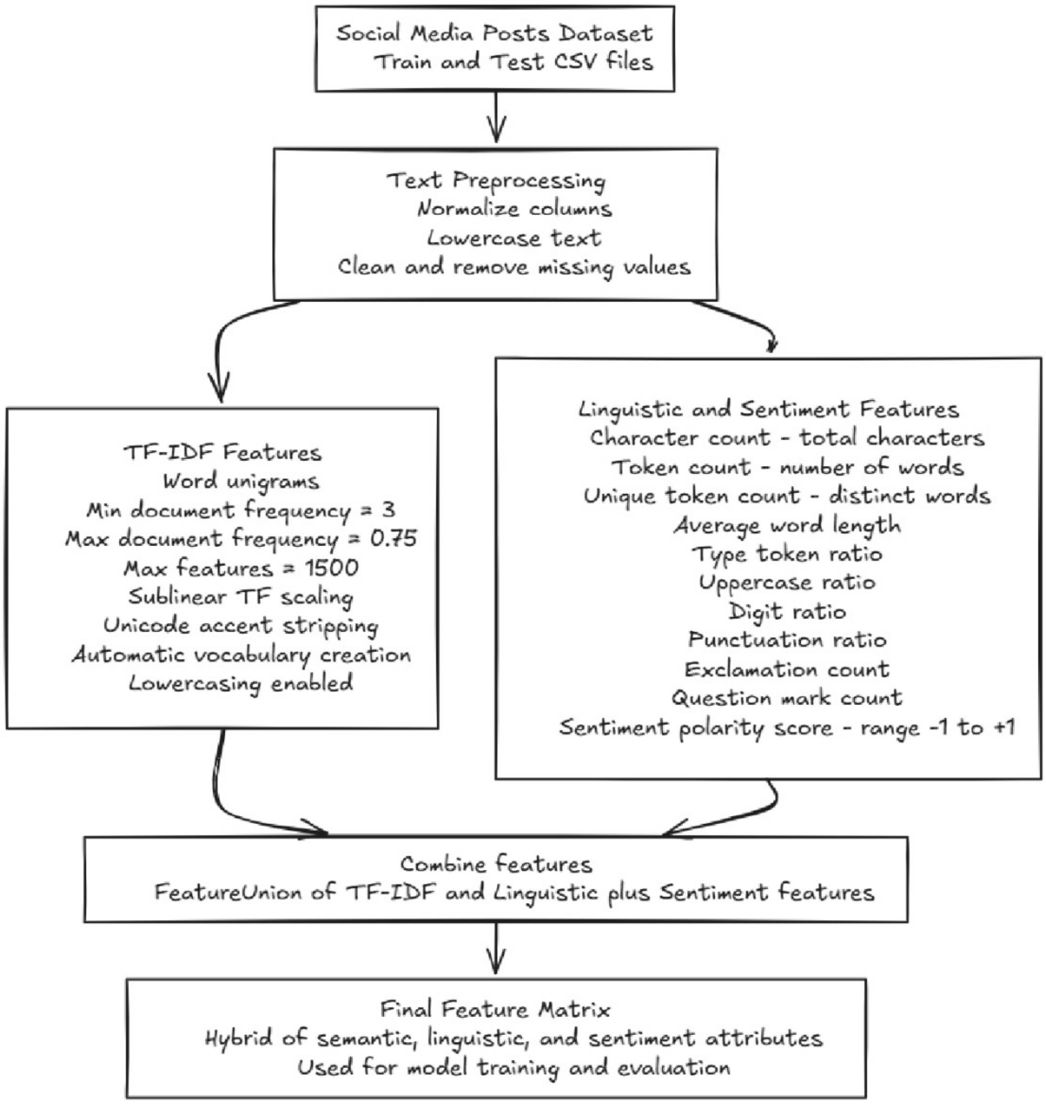

Figure 3: Enhanced feature extraction pipeline used in Stage 3, combining TF–IDF with linguistic, stylistic, and sentiment polarity features for social media posts.

5.5 Model Pipeline

1. Feature input
2. Classifier training
3. Output prediction

6 Implementation Details

6.1 Tools and Technologies

The project was implemented using Python and standard machine learning libraries. All data preprocessing, feature extraction, sentiment analysis, and model training were carried out using:

- Python 3.x
- Scikit-learn for ML models and evaluation
- Pandas and NumPy for data handling and numerical processing
- TextBlob for computing sentiment polarity scores of posts
- Matplotlib for plotting accuracy comparisons

These tools provide a stable and reproducible workflow for text classification tasks.

6.2 Libraries Used

The following Python modules were essential for building and evaluating the models:

- `sklearn.feature_extraction.text` (TF–IDF vectorizer)
- `sklearn.ensemble` (Random Forest, AdaBoost, Gradient Boosting, Voting Classifier)
- `sklearn.svm` (Linear SVM classifier)
- `sklearn.model_selection` (train–test split, cross-validation)
- `sklearn.metrics` (accuracy, F1-score, precision, recall)
- `textblob` (sentiment polarity calculation)
- `matplotlib.pyplot` (visualization of model accuracy)

6.3 Script Details

6.3.1 main.py

Implements the research paper's methodology using the news article dataset. It performs TF–IDF extraction, computes basic linguistic features, trains classical ML models, and evaluates their performance.

6.3.2 mainpostfinal.py

Handles the manually collected social media posts using the same models and feature set from Stage 1. It evaluates how the article-trained feature design performs on short informal text.

6.3.3 newfeaturesposts.py

Implements enhanced linguistic, stylistic, and sentiment-based features inspired by the second research paper. Using TextBlob, a sentiment polarity score is added to each post. The models are retrained with these new features to improve fake news detection accuracy on short social media posts.

7 Evaluation Metrics

7.1 Accuracy

Accuracy was used to measure the overall correctness of each model. It represents the proportion of total predictions that were correctly classified as real or fake.

7.2 Precision, Recall, and F1-Score

Since real and fake news may not be evenly distributed, precision, recall, and F1-score were used to provide a more balanced evaluation:

- **Precision** measures how many predicted fake news samples were actually fake.
- **Recall** measures how many fake news samples the model successfully detected.
- **F1-Score** is the harmonic mean of precision and recall, giving a single balanced metric.

These metrics help evaluate the model's performance beyond simple accuracy, especially in cases of uneven class distribution.

8 Results and Analysis

8.1 Stage 1 Results

The models were trained on the news article dataset using the feature extraction approach described in the research paper. The table below summarizes the performance of all classifiers.

Model	Precision (0/1)	Recall (0/1)	F1-Score (0/1)	Accuracy
Logistic Regression	0.9929 / 0.9877	0.9887 / 0.9923	0.9908 / 0.9900	0.9904
Linear SVM	0.9994 / 0.7307	0.6640 / 0.9995	0.7979 / 0.8442	0.8241
Random Forest	0.9930 / 0.9921	0.9928 / 0.9923	0.9929 / 0.9922	0.9925
AdaBoost	0.9972 / 0.9933	0.9938 / 0.9970	0.9955 / 0.9951	0.9953
Bagging (Decision Tree)	0.9981 / 0.9995	0.9996 / 0.9979	0.9988 / 0.9987	0.9988
Gradient Boosting	0.9979 / 0.9967	0.9970 / 0.9977	0.9974 / 0.9972	0.9973
Voting Classifier	0.9907 / 0.9937	0.9943 / 0.9897	0.9925 / 0.9917	0.9921

Table 1: Performance metrics for Stage 1 models trained on the news article dataset.

8.2 Stage 2 Results

The following results were obtained by applying the same models and features from Stage 1 to the manually collected social media posts. The models appear in the same order as Stage 1 for consistency.

Model	Precision (0/1)	Recall (0/1)	F1-Score (0/1)	Accuracy
Logistic Regression	0.68 / 0.68	0.68 / 0.68	0.68 / 0.68	0.6786
Linear SVM	0.84 / 0.84	0.84 / 0.84	0.84 / 0.84	0.8393
Random Forest	0.77 / 0.76	0.77 / 0.77	0.77 / 0.76	0.7679
AdaBoost	0.76 / 0.74	0.75 / 0.75	0.75 / 0.75	0.7500
Bagging (Decision Tree)	0.75 / 0.75	0.75 / 0.75	0.75 / 0.75	0.7500
Gradient Boosting	0.77 / 0.76	0.77 / 0.77	0.77 / 0.77	0.7679
Voting Classifier	0.75 / 0.75	0.75 / 0.75	0.75 / 0.75	0.7500

Table 2: Performance metrics for Stage 2 models trained on one-line social media posts, ordered to match Stage 1.

8.3 Stage 3 Results

The following results were obtained by training all models using the enhanced feature set inspired by the second research paper. The table below follows the same model ordering as Stage 1 and Stage 2 for consistency.

Model	Precision (0/1)	Recall (0/1)	F1-Score (0/1)	Accuracy
Logistic Regression	0.8780 / 0.9524	0.9558 / 0.8696	0.9153 / 0.9091	0.9123
Linear SVM	0.8527 / 0.9697	0.9735 / 0.8348	0.9091 / 0.8972	0.9035
Random Forest	0.8560 / 0.9417	0.9469 / 0.8435	0.8992 / 0.8899	0.8947
AdaBoost	0.8168 / 0.9381	0.9469 / 0.7913	0.8770 / 0.8585	0.8684
Bagging (Decision Tree)	0.7985 / 0.9362	0.9469 / 0.7652	0.8664 / 0.8421	0.8553
Gradient Boosting	0.8120 / 0.9474	0.9558 / 0.7826	0.8780 / 0.8571	0.8684

Table 3: Performance metrics for Stage 3 models trained on social media posts using enhanced linguistic features.

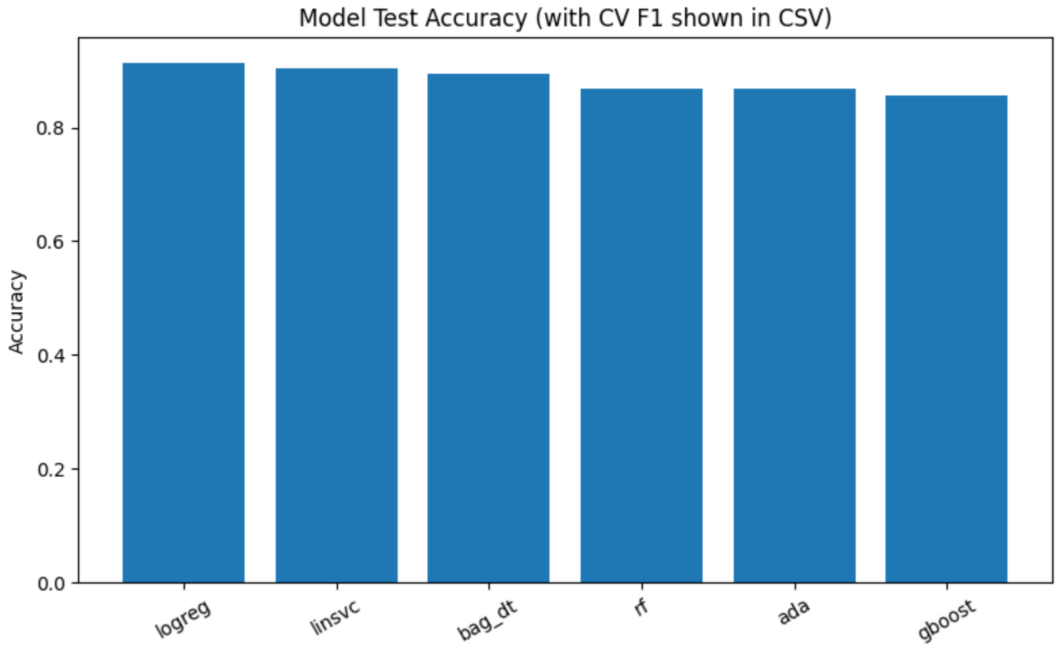

Figure 4: Model Test Accuracy for Stage 3 Using Enhanced Feature Set

8.4 Comparative Study: Stage 2 vs Stage 3

This section compares the performance of all models on Stage 2 (original features on social media posts) and Stage 3 (enhanced linguistic features). The comparison includes accuracy, precision, recall, and F1-score for both classes.

Model	Stage 2 Accuracy	Stage 3 Accuracy
Logistic Regression	0.6786	0.9123
Linear SVM	0.8393	0.9035
Random Forest	0.7679	0.8947
AdaBoost	0.7500	0.8684
Bagging (Decision Tree)	0.7500	0.8553
Gradient Boosting	0.7679	0.8684

Table 4: Accuracy comparison between Stage 2 and Stage 3.

Model	Stage 2 Precision (0/1)	Stage 3 Precision (0/1)
Logistic Regression	0.68 / 0.68	0.8780 / 0.9524
Linear SVM	0.84 / 0.84	0.8527 / 0.9697
Random Forest	0.77 / 0.76	0.8560 / 0.9417
AdaBoost	0.76 / 0.74	0.8168 / 0.9381
Bagging (Decision Tree)	0.75 / 0.75	0.7985 / 0.9362
Gradient Boosting	0.77 / 0.76	0.8120 / 0.9474

Table 5: Precision comparison between Stage 2 and Stage 3.

Model	Stage 2 Recall (0/1)	Stage 3 Recall (0/1)
Logistic Regression	0.68 / 0.68	0.9558 / 0.8696
Linear SVM	0.84 / 0.84	0.9735 / 0.8348
Random Forest	0.77 / 0.77	0.9469 / 0.8435
AdaBoost	0.75 / 0.75	0.9469 / 0.7913
Bagging (Decision Tree)	0.75 / 0.75	0.9469 / 0.7652
Gradient Boosting	0.77 / 0.77	0.9558 / 0.7826

Table 6: Recall comparison between Stage 2 and Stage 3.

Model	Stage 2 F1-score (0/1)	Stage 3 F1-score (0/1)
Logistic Regression	0.68 / 0.68	0.9153 / 0.9091
Linear SVM	0.84 / 0.84	0.9091 / 0.8972
Random Forest	0.77 / 0.76	0.8992 / 0.8899
AdaBoost	0.75 / 0.75	0.8770 / 0.8585
Bagging (Decision Tree)	0.75 / 0.75	0.8664 / 0.8421
Gradient Boosting	0.77 / 0.77	0.8780 / 0.8571

Table 7: F1-score comparison between Stage 2 and Stage 3.

8.5 Insights

The comparison between Stage 2 and Stage 3 clearly shows that feature engineering has a significant impact on fake news detection for short social media posts. While the original TF–IDF-based features performed adequately on longer news articles, they failed to generalize well to one-line posts due to their limited context and sparse representations.

The enhanced linguistic and stylistic features introduced in Stage 3 provided additional discriminative information such as lexical diversity, punctuation patterns, digit usage, and uppercase proportions. These features capture writing behaviour more effectively, allowing the models to distinguish between real and fake posts even when textual content is minimal.

Models such as Logistic Regression, Linear SVM, and Random Forest benefited the most from the improved feature set, showing substantial gains in accuracy, precision, recall, and F1-score. Overall, the results highlight that domain-specific feature engineering is essential for achieving robust performance on short and noisy social media text.

9 Discussion

9.1 Performance on News Articles

Models demonstrated strong performance on the news article dataset because the text is long, structured, and context-rich. TF–IDF features capture meaningful word patterns effectively in such settings, allowing classical machine learning models to separate real and fake articles with high accuracy. Ensemble models in particular benefited from the availability of consistent linguistic cues.

9.2 Performance on Social Media Posts

Performance dropped considerably in Stage 2 due to the extremely short and informal nature of one-line social media posts. These posts provide limited context, causing TF–IDF vectors to become sparse and less informative. As a result, the models relied heavily on subtle stylistic cues rather than content. Performance improved in Stage 3 only after incorporating enhanced linguistic, stylistic, and sentiment-based features, which captured writing behaviour and emotional tone more effectively and reduced the impact of missing context.

9.3 Limitations

- The manually collected social media dataset is relatively small, limiting the generalizability of results.
- The enhanced features in Stage 3 are still hand-crafted and may not generalize to different writing styles or platforms.

10 Conclusion

This project successfully reproduced the methodology of a published research paper on fake news detection and extended it across multiple text domains. The models achieved strong performance on the news article dataset due to the availability of structured, context-rich text. However, applying the same models and TF–IDF features to short social media posts in Stage 2 resulted in a notable drop in performance, underscoring the difficulty of cross-domain generalization.

By incorporating enhanced linguistic, stylistic, and sentiment features inspired by the second research paper, Stage 3 achieved substantial improvements on the social media dataset. These additional features enabled the models to capture writing behaviour, stylistic patterns, and emotional polarity that are crucial for detecting fake news in short and noisy text.

Overall, the results demonstrate that domain-specific feature engineering plays a central role in improving fake news detection performance across diverse text formats. The project highlights the importance of adapting features—not only models—when transferring fake news detection systems from formal news articles to informal social media content.

11 References

1. H. Ahmed, I. Traore, and S. Saad, "Detecting Fake News Using Machine Learning and Ensemble Methods," *Proceedings of the 2017 International Conference on Computer Systems and Applications*, IEEE, 2017.

2. S. Kaur and S. Sharma, "Fake News Detection on Social Media Using Linguistic and Stylistic Features," *International Journal of Engineering and Advanced Technology*, vol. 8, no. 5, 2019.

3. `scikit-learn`: Machine Learning in Python. Documentation available at: https://scikit-learn.org/

4. Python Data Analysis Libraries: `NumPy`, `Pandas`, `Matplotlib`. Documentation available at: https://numpy.org/, https://pandas.pydata.org/

5. TextBlob: Simplified Text Processing for Python. Documentation available at: https://textblob.readthedocs.io/

6. Manually collected social media fake news dataset (`all_posts.csv`), created as part of this project.

7. Instagram news source used for collecting short social media posts: https://www.instagram.com/thetatvaindia/

8. Instagram satire/fake content channel used for pattern observation: https://www.instagram.com/fakenewsnetwork/channel/

9. BBC News – real news source referenced for dataset understanding: https://www.bbc.com/news

10. Reuters – trusted global news source referenced during dataset study: https://www.reuters.com/

11. Hindustan Times – Indian news source used for dataset comparison: https://www.hindustantimes.com/

12. ISOT Fake News Dataset (used in Research Paper 1): https://www.kaggle.com/datasets/csmalarkodi/isot-fake-news-dataset

13. Satire News Source (used while understanding fake news textual patterns): https://theonion.com/

14. Satire News Source for linguistic comparison: https://www.thedailymash.co.uk/