# A Hybrid Embedding Model Designed to Better Identify Fake News on Twitter

**Deepthi S[1] , Iliyaz Pasha M [2]**
**[1]M. Tech Student,**
**[2]Assisstant Professor**
**Department of CSE, RLJIT,**
**Doddaballapur**

## Abstract

With the growing use of social media platforms such as Facebook, Instagram, and Twitter for sharing news, misinformation can spread to millions within seconds, causing serious issues like public confusion and biased opinions. The goal of this research is to develop a system to detect bogus news through machine learning and artificial intelligence. By applying binary classification, the system sorts news articles as either real or fake. We use decision tree models in addition to Bayesian classifiers to improve accuracy in identifying misleading content. The project centers on social media networks, particularly Twitter, using a provided dataset to evaluate the approach. Our results indicate that using machine learning effectively helps reduce the proliferation of false information and increases the reliability of what people view online.

**Keywords:** Fake News Detection, Natural language processing, machine learning, and Twitter, Python, Social Media, Misinformation.

## Introduction

In today's social media-driven world, sites such as Facebook and Twitter have become essential sources of news for many people. While these platforms make sharing information quick and simple additionally, they generate chances for fake news to spread rapidly. Fake news refers to intentionally misleading or false information, often created to serve political, financial, or social goals. Because it frequently appears to be authentic news, but it can be challenging totell the difference, leading to confusion, mistrust in the media, and harmful societal effects such as influencing elections or public health decisions. Due to this growing issue, smarter solutions, and machine learning offers one possible approach. By training algorithms to recognize patterns in language, tone, and source credibility, we can teach computers to tell apart real from fake content. In our study, we used a dataset of authentic and fake news articles, cleaned the text, applied TF-IDF for feature extraction, and tested models like Decision Trees and Naive Bayes. Well-performing was the Decision Tree model,though it had some limitations like overfitting and difficulty with unfamiliar data. We also highlighted the importance of high-quality, unbiased, and culturally sensitive datasets, especially since fake news can differ by region or topic. Looking ahead, future systems could be enhanced with more advanced algorithms, real-time tools like browser extensions, and features such as knowledge graphs. However, while technology can assist, ethical considerations, transparency, and user appeal processes are also essential. Most importantly, media literacy— teaching people to question sources and recognize bias—is crucial. Addressing fake news isn't just a technical challenge.

## Related work

Social media platforms like Twitter are increasingly unreliable as news sources due to the rapid spread of fake news. While deep learning models— including powerful transformer-based systems—can detect misinformation with high accuracy, they often function as computers to tell apart real from fake opaque, difficult to see black boxes trust. Using ensemble approaches that blend techniques like BERT, CNN, and Bi-LSTM delivers top-tier accuracy (around **98–99%**) while also allowing for explanations of key text features, making fake news detection both more effective and more transparent.[1]

Social media misinformation is increasingly problematic as many people rely on these platforms for news, yet the powerful transformer-based models we use to detect fake content often act as opaque "black boxes." A

promising solution is a hybrid DistilBERT + BiLSTM model that pairs high accuracy (~98%) with interpretability using LIME, allowing users to see which words influence decisions. This method not only performs better than other popular models like T5 and ALBERT but also builds trust by explaining why a post is flagged as fake.[2]

Fake news can seriously erode public trust, influence elections,harm public health, and even disrupt economies—all thanks to how quickly misleading content can spread across social media. Graph Neural Networks (GNNs) offer a smarter solution by modeling how posts, users, retweets, and comments interact within social networks,learning hiddenpatterns of misinformation. Studies show that ensemble GNNs combining models like GAT, GCN, and BiGCN with text embeddings (e.g., BERT or spaCy) significantly outperform traditional content-based methods—especially when labeled data is limited.[3]

Misinformation spreads rapidly across news websites, undermining public trust, destabilizing democratic processes, and creating confusion. In our real-time fake news detection project, we compared LSTM (~96%), ALBERT (≈100%), FNNet (~93%), and a hybrid CNN+RNN model (~98.8%) tested on standard datasets—the hybrid CNN+RNN stacked the highest overall accuracy after ALBERT, indicating strong real- time performance with both speed and reliability. This highlights how combining spatial and sequential learning architectures can create an effective, trustworthy way to spot false information online.[4]

Diagnosing Autism Spectrum Disorder (ASD) early is crucial, but traditional methods can be subjective and time-consuming. A potential method achieves an astounding 99.3% accuracy in detecting ASD by combining deep learning models applied to clinical data with sentiment analysis of social media posts. In addition to improving diagnosis accuracy, this hybrid architecture provides a quicker and easier way to implement early intervention.[5]

## System Architecture

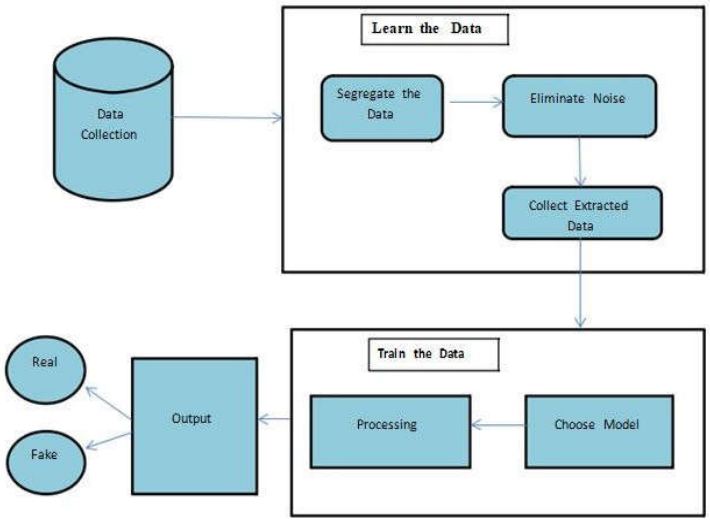



Figure 1: System architecture

Figure 1 describes data collection. Large-scale tweets are harvested using APIs or scraping tools like Twint, based on relevant keywords or hashtags (e.g., “covid19,” “election”) to capture both real and false narratives in the real world. This raw dataset often contains hundreds of thousands of tweets spanning years or live event streams. Next follows data segregation and cleaning, where tweets are scrubbed of noise—URLs, emojis, excessive whitespace, punctuation, non-Latin characters, and stop words—with tools like spaCy, NLTK, TextBlob, and Text2Emotion. Lemmatization standardizes word forms, while sentiment and emotion scoring tag each tweet with polarity and emotional tone to strengthen later feature extraction. Duplicate tweets, retweets, and trivial or very short messages are removed to reduce redundancy and undue influence on models.

Once cleaned, the system moves into noise elimination and feature extraction, by using feature extraction

method constructs a feature space that includes tweet- specific, user metadata, content-based, temporal, stylistic, sentiment, and propagation features: tweet length, punctuation frequency, hashtags, uppercase ratios, presence of media links; user follower/friend counts, account age, verification; network diffusion metrics like retweet/favorite counts and depth; sentiment/emotion scores; readability (e.g., Flesch score); and even text polarity. TF–IDF vectors and word embeddings (Word2Vec, GloVe, BERT) further encode semantic content . For image-based content, some systems may also signal manipulated visuals or deepfakes.The next step is structuring the extracted data into training-ready formats: feature matrices combining numerical, categorical, and embedding inputs, optionallyaugmented by labeled ground truth from crowd-sourced websites like as Amazon Mechanical Turk, verified fact- checker datasets (e.g., PHEME, Liar, FakeNewsNet), or manual review. Class labels (real or fake) are assigned, and the datasets will be divided into test, validation, and training sets.

In model selection, researchers often explore both classical and methods for deep learning. Ensembles of logistics regression, S V M, Random forest, XGBoost, Passive- Aggressive, or SGD provide lightweight, explainable baselines . Hybrid architectures combining CNN or Bi-LSTM layers atop transformer encoders (like BERT or DistilBERT) have also shown strong performance, often achieving >96% F1 scores . Social network embeddings—graph representations of users' follower/friend clusters—have been found to significantly boost accuracy compared to text-only models. During training, feature selection techniques (e.g., variance thresholding, PCA, SMOTE/ADASYN for class imbalance) optimize the input space . Training pipelines may include pretraining transformer components on Twitter text, followed by fine-tuning on labeled news tweets, with cross-validation used to prevent overfitting and tune hyperparameters.

At inference, each new tweet undergoes the same preprocessing and feature encoding, then is classified—often in real time—using the trained model to produce a "real" or "fake" label, sometimes with an associated confidence score. Finally, in the real/fake evaluation stage, model outputs are continuously monitored. Performance indicators including recall and precision, F- score, and inference time are tracked on evolving test sets; error analysis highlights bias or drift. Explainability modules (e.g., highlighting key features like hashtags, sentiment, propagation signatures) help users and moderators understand decisions . Periodic retraining with fresh labeled data ensures the system adapts to changing misinformation tactics.

## Data flow diagram

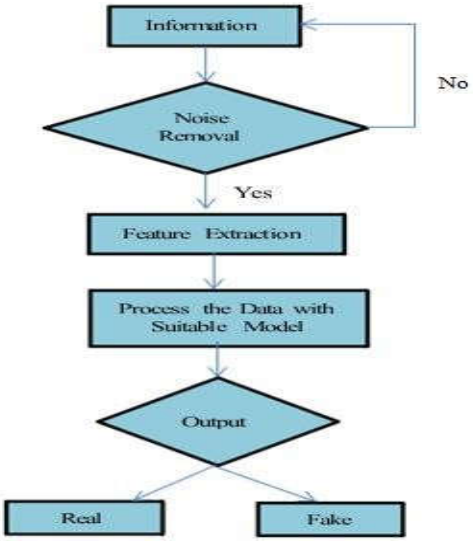



Figure 2: Data flow diagram

Figure 2 starts the process by ingesting tweets—either in real time via the Twitter API or from historical datasets—and filtering out those lacking substantial content. Once meaningful tweets are identified, they

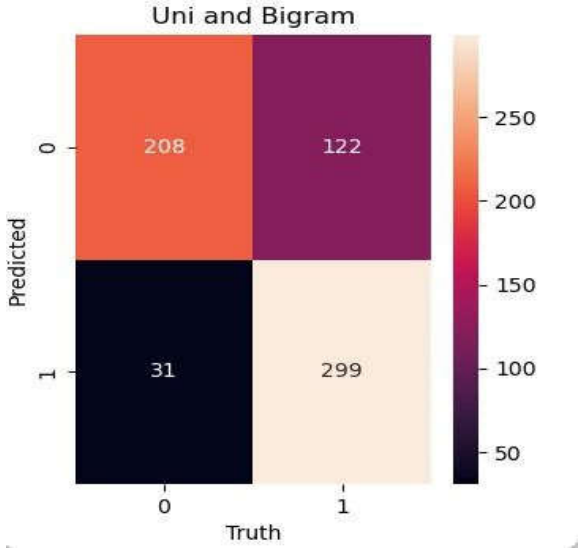
undergo noise removal, where URLs, mentions, hashtags, emojis, punctuation, and non-alphanumeric characters are stripped away. The cleaned text is then tokenized, lowercased, stop-words removed, and optionally lemmatized or stemmed using tools like NLTK or spaCy to normalize the language. Next, each tweet is transformed into a numerical feature vector using techniques like Bag- of- Words or TF–IDF, enabling the model to quantify word frequency and importance across the corpus. A Naïve Bayes classifier that is multinomial is trained on labeled data (real vs. fake), with Laplace smoothing applied to manage unseen terms. This model excels in handling short, high-dimensional text and delivers results quickly. It's evaluated rigorously use of metrics such as F1-score, recall, accuracy, precision and confusion matrices on a held-out test set—studies using similar models report test accuracies around 91– 92%, with F1- scores in the high- 80s to low- 90s range.

During deployment, tweets flow through the same cleaning and vectorization steps, and the trained model produces a real/fake classification (often with posterior probability scores), which can be logged or surfaced via a lightweight API or dashboard. While Naïve Bayes is fast, interpretable, and effective, it does rely on simplification of presuming the independence of features and may misclassify nuanced or sarcastic content. Researchers often enhance performance by incorporating metadata—such as user credibility, tweet popularity, or sentiment—or by ensembling with other models like logistic regression, S V M, or transformer-based classifiers to reduce false positives and boost recall. In production settings, the system is typically wrapped in a REST API using frameworks like Flask or FastAPI and containerized via Docker, with orchestration by tools like Kafka or Airflow. Dashboards built on platforms such as Streamlit or Dash often provide visibility into detected fake-news trends, flagged tweets, and potential model drift over time.

**Algorithm: Fake News Detection System**

1. **Start**
2. **Input Information**
   - Collect data (news articles, tweets, headlines, etc.)
3. **Check for Noise**
   - If the data contains noise (e.g., punctuation, stop words, special characters):
     - Go to Step 4
   - Else:
     - Loop back to Step 2
4. **Noise Removal**
   - Clean the data by removing unwanted elements like:
     - Stop words
     - Punctuation marks
     - HTML tags
     - Special characters
5. **Feature Extraction**
   - Transform text information into numerical form using:
     - TF-IDF (Term Frequency- Inverse Document Frequency)
     - CountVectorizer
     - Word embeddings (optional)
6. **Model Processing**
   - Select and train a machine learning model:
     - Decision Tree / Naive Bayes / SVM / LSTM
   - To categorize the news, enter the features into the model.
7. **Get Output**
   - Considering the trained model's prediction:
     - If classified as **Real**, label the news as Real
     - If classified as **Fake**, label the news as Fake
8. **End**

## Result and Discussion




In fake news detection on Twitter, unigrams (single words like “fake” or “hoax”) and bigrams (two-word combinations like “fake news” or “no evidence”) are essential features. Studies have shown that the combination of unigrams and bigrams enhances model accuracy. For instance, a Decision Tree (C4.5) model achieved 61.13% accuracy using both unigrams and bigrams, compared to 60.93% with unigrams alone and 57.97% with bigrams alone. Applying TF-IDF weighting further improved accuracy, with unigrams reaching 62.96% and bigrams 61.73%. Advanced models like Bi-LSTM showed even greater improvements, with accuracy increasing from 92.70% to 94.07% when combining TF-IDF-weighted unigrams and bigrams. This shows that integrating unigrams, bigrams, and TF-IDF significantly enhances the detection of fake information on Twitter.

## Conclusion

Our fake-news detection system for Twitter is built entirely in Python, leveraging classic machine-learning tools to fight misinformation. We began with a labeled tweet dataset—each marked real or fake—then cleaned it by stripping punctuation, URLs, special characters, stopwords, and normalizing text to lowercase. Next, NLP techniques like tokenization and vectorization (using TF- IDF and CountVectorizer) converted the cleaned tweets into numerical features. Using train/test splits and measures like Accuracy, Precision, Recall, and F1-Score, a number of classifiers are trained, including logistic regression, Naive Bayes, and SVM. The best model delivered strong performance in flagging fake tweets. Beyond coding with Python libraries like pandas, scikit- learn, and NLTK, this project deepened our appreciation of AI’s vital role in responsibly combating social-media misinformation.

## Reference

[1] B. C. Uyanage, G.U. Ganegoda. Fake News Detection on Twitter.2024 9th International Conference on Information Technology Research (ICITR)| DOI: 10.1109/ICITR64794.2024.10857752

[2] kainat Irfan, Muhammad Wasim, Sehrash Safda, Abdur Rehman, Muhammad Usman Ghani.XFND: Explainable Fake News Detection using a Hybrid DistillBERT and BiLSTM.2025 International Conference on Emerging Technologies in Electronics, Computing, and Communication (ICETECC)| DOI: 10.1109/ICETECC65365.2025.11070272

[3] Shivansh Mishra, Naween Kumar, Ojal Agarwal, Saksham Arora, Chesta Mehta.Graph Neural Networks for the Development of Efficient Fake News Detection. 2025 International Conference on Cognitive Computing in Engineering, Communications, Sciences and Biomedical Health Informatics (IC3ECSBHI) | DOI: 10.1109/IC3ECSBHI63591.2025.10990891

[4] K. Praveen Nandan, B. Pakruddin, Syed Afridi , Shailesh K.R , Shubam V. Patil.Real-Time Detection of Fake News Articles Using Deep Learning Techniques.2025 International Conference on Next Generation Communication & Information Processing (INCIP) DOI: 10.1109/INCIP64058.2025.11019208
[5] Leo Prasanth L, Susi E, Selvin Ebenezer S. Hybrid Deep Learning for ASD Prediction: Integrating Sentiment Analysis and Character-Level Embedding from Tweets and Clinical Data. 2025 International Conference on Data Science, Agents & Artificial Intelligence (ICDSAAI)|DOI: 10.1109/ICDSAAI65575.2025.11011691.
[6] Shiyong Xiong , Tingting Huang , Huajian Xie , Jun Shen . Sentence-Guided Comment Tree Fusion for Fake News Detection .2024 4th International Conference on Artificial Intelligence, Robotics, and Communication (ICAIRC)|DOI: 10.1109/ICAIRC64177.2024.10900273
[7] Prakhar Mittal, Jasleen Singh Saini, Ankita Agarwal, Rajeev Kumar Maheshwari, Sandeep Kumar, Arjun Singh .Fake News Detection Using Machine Learning Techniques.2024 4th International Conference on Advancement in Electronics & Communication Engineering (AECE) | DOI: 10.1109/AECE62803.2024.10911448
[8] Madhuri J,Gagan D,Ghana Shyam S,Amarapuram Rohit, Adusumalli Naveen Kumar. Enhanced Fake Review Detection and Sentiment Analysis with Hybrid Deep Learning Models 2024 IEEE International Conference on Blockchain and Distributed Systems Security (ICBDS)| DOI: 10.1109/ICBDS61829.2024.10837222
[9] Prisha Gupta, P. K. Gupta. Performance Analysis of GCN, GNN, and GAT Models with Differentiable Pooling for Detection of Fake News. 2024 3rd Edition of IEEE Delhi Section Flagship Conference (DELCON)|DOI: 10.1109/DELCON64804.2024.10866089
[10] Thanaphan Bhatia, Bundit Manaskasemsak, Arnon Rungsawang. Detecting Fake News Sources on Twitter Using Deep Neural Network 2023 11th International Conference on Information and Education Technology (ICIET)|DOI: 10.1109/ICIET56899.2023.10111446
[11] Matsumoto Hayato, Soh Yoshida, and Mitsuji Muneyasu. Flexible Framework to Provide Explainability for Fake News Detection Methods on Social Media.2022 IEEE 11th Global Conference on Consumer Electronics (GCCE) |DOI: 10.1109/GCCE56475.2022.10014350
[12] Parthiban.G, Dr.M. Germanaus Alex, Dr. S. John Peter. Review of Fake News Detection in Social Media using Machine Learning Techniques. 2022 International Conference on Augmented Intelligence and Sustainable Systems (ICAISS) | DOI: 10.1109/ICAISS55157.2022.10010796