Fake/ Real News Headlines Identification Using Real-time API Fetching System

Mrs. M. Pandi Deepa[1], S. Kumanan[2],
[1]*Assistant Professor, Department of Computer Application, SRM Valliammai Engineering college, Anna University, Chennai, Tamil Nadu, India*
[2] *PG Student, Department of Computer Application, SRM Valliammai Engineering college, Anna University, Chennai, Tamil Nadu, India*

Abstract- In the current quick-paced digital landscape, the wide dissemination of fake news and disinformation has grown to be an important challenge. Social media sites and online news websites allow users to post whatever they want to, making it more challenging to differentiate between factual and fabricated information. Fake news can have harsh effects, determining public opinion, elections, medical choices, as well as societal stability. Hence, creating a system to identify automatically fake news is crucial in order to enhance credible information and minimize misinformation. The system is trained on a labeled dataset with both real and fake news items. It applies TF-IDF vectorization to extract significant features from the text and a Naive Bayes classifier for prediction. After the model has been trained, it will be able to scan a user-input headline and provide a judgment of whether or not it is likely real or fake. The solution aids in automating verification and provides instant, reliable outcomes. In order to enhance the system's reliability, the application is coupled with the News API that retrieves associated news articles in real-time according to the input headline of the user. This integration of machine learning with real-time data brings a solid layer of authenticity. The application is developed with Flask for web backend and HTML and CSS for frontend. Machine learning parts are developed utilizing Python libraries such as scikit-learn and pandas. The end result is a simple web interface where users can just enter the news headline and get an instant verification of its authenticity. This system can prove to be an effective weapon in the hands of journalists, students, researchers, and the common public to battle the increasing menace of false news.

Keywords: Fake News Detection, Machine Learning, Natural Language Processing (NLP), Real-time News Validation, News API Integration, Headline Classification, Misinformation Detection.

1. INTRODUCTION

In today's world, where digital media is the primary source of information for billions of people across the world, the credibility of the received news has emerged as a top priority. With the increasing popularity of social media platforms, news websites, and instant messaging apps for daily news, it has become scarily easy for fake information, or "fake news," to travel fast across the globe. Such a trend is threatening the foundations of democracy, public trust, national security, and social harmony. Fake news can influence the opinion of people, impact the outcome of elections, cause violence, and lead to mass confusion or panic. Hence, the need for smart systems that can verify the credibility of news content in real-time is imperative.

Conventional fact-checking is dependent on human effort and manual verification, which is not only time-consuming but also impractical due to the sheer amount of news produced every minute. Automated fake news verification systems offer a scalable and efficient way to manually fight misinformation. Most current systems, however, are pre-trained models and historical data, which might not be able to process current and changing news content. This is where real-time verification systems step in.

This project proposes a Fake/Real News Headlines Identification System that verifies news headlines based on real-time API fetch and intelligent comparison techniques. Unlike static models, this system periodically calls reliable news APIs such as NewsAPI, ContextualWeb, and GNews, fetching the latest news articles and corresponding headlines for a given user-input. It then moves on to contextual

analysis, keyword identification, and similarity matching to determine whether the news is supported by original sources. The system functions on the assumption that real news will typically be reported by multiple credible sources and will share comparable context, title, and synopsis. If the headline submitted is highly similar to information retrieved from credible news APIs, it gets marked as "Real". Yet, if there is no match or considerable discrepancy or if facts contradict verified sources, then the headline is marked as "Fake" or "Unverified".

Using real-time news APIs makes the system event-sensitive and dynamic, reducing the dependency on previous datasets. The project can also be extended to include machine learning models, such as logistic regression or transformer models (e.g., BERT), to enhance the accuracy of detection and facilitate continuous learning.Effectively, this system is a computer fact-checking assistant for the average user, journalists, and organizations that need rapid, real-time fact verification of news headlines. It closes the gap between static fake news filters and dynamic real-world news events, presenting a proactive solution to the fight against digital disinformation.

2. LITERATURE REVIEW

The propagation of fake news on the internet has led to extensive research on the application of automated methods for verifying the authenticity of news and avoiding misinformation. Researchers and developers have explored a range of methods including machine learning, deep learning, natural language processing (NLP), crowd-sourced verification, and rule-based systems. Each of these methods has its pros and cons, particularly when applied to real-time detection of fake news.

Traditional research has predominantly applied the standard supervised machine learning models of Naïve Bayes, Support Vector Machines (SVM), Decision Trees, and Logistic Regression. The models revolved around the exploration of linguistic features, writing styles, and metadata in classifying authentic vs. forged or satirical texts. Rubin et al. (2016), for instance, focused on employing similar approaches in detecting satire and forged texts. However, the primary drawback of the earlier models is that they employ labeled predefined training data and hence become very poor performers on varied or never-before-seen news articles.

With advancements in deep learning and NLP, more recent methods have emerged that offer more contextual sensitivity. Recurrent Neural Network (RNN) and transformer-based models such as BERT and RoBERTa have significantly enhanced the accuracy of text classification by capturing semantic relationships and linguistic subtleties. Vaswani et al. (2017) introduced the transformer model, which opened the doors to attention-based NLP. Then, Kaliyar et al. (2021) demonstrated the effectiveness of BERT in detecting fake news with a better understanding of context. Although strong, these models have a tendency to utilize vast computational resources and still depend on large-scale, labeled data. Researchers have also relied heavily on publicly available datasets like the LIAR dataset, FakeNewsNet, and ISOT for training and testing fake news detection models. While these datasets are valuable, their static nature and limited update frequency reduce their relevance for identifying fake news in real-time scenarios.

Very little has been accomplished in the domain of real-time verification through the use of dynamic data sources such as news APIs. The majority of the systems continue to run on historical data rather than querying reliable sources in real time. Nonetheless, research, by Shu et al. (2020), suggested the use of information sources in real time toward the detection of misinformation. API-based systems on platforms such as NewsAPI.org or GDELT have the potential toward providing timely content but will lack significant semantic comparison capacity and contextual reasoning.

Since source-based and content-based systems are not efficient, hybrid systems came into existence. They incorporate semantic analysis of text with external credibility evaluation. The proposed system by Popat et al. (2018) is a case in point, which utilizes external evidence and knowledge graphs to evaluate claims. Although they are advantageous, they are difficult to design and require well-prepared data sources that can be flexible and scalable. In brief, despite remarkable advances in machine learning and NLP-based fake news verification, the majority of current systems

remain without real-time, user-initiated verification. It is this that serves to underscore the necessity for dynamic systems with real-time API download and smart matching processes—above and beyond static dataset limitations and improving the applicability and reliability of fake news verification in today's rapidly changing digital landscape.

3. SYSTEM ANALYSIS

System analysis entails the analysis of the present situation of news verification technologies and determining improvements that can be incorporated in a new system. This section presents the current systems, their weaknesses, and how the proposed system improves upon the limitations.

3.1 .SYSTEM STUDY

System study covers an in-depth study and knowledge of the proposed solution, including how it operates, engages with the users and other external entities, and how it addresses the problems raised by the existing systems. It serves as the foundation for designing and implementing the system successfully.

3.2 EXISTING SYSTEM

The idea of identifying fake news through the current systems is heavily reliant on manually fact-checking websites, pre-trained machine learning models, and platform-specific filtering like that which is used by social media companies. While such methods provide some level of validation, they also come with tremendous flaws. Manually fact-checking is tiresome and is not scalable to the volume of content generated each day. Pre-trained models rely on outdated datasets and may not be capable of identifying newly emerging false headlines. Social media filters are confined to their individual sites and are not public domain.

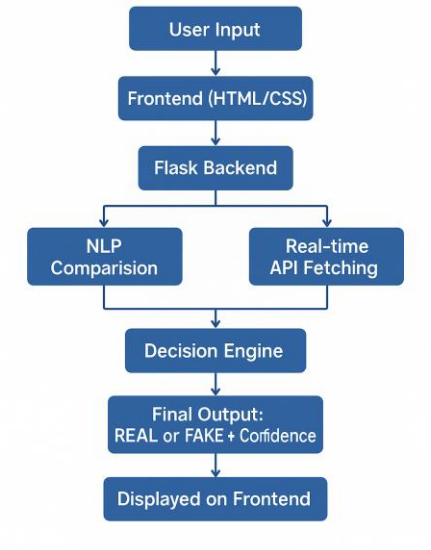
3.3 PROPOSED SYSTEM

The system sent here offers a hybrid approach with real-time data pulling from credible news APIs intertwined with intelligent comparison techniques and optional machine learning. The question for the user is what headline they want to look for, and the system verifies it by querying sources such as NewsAPI or GNews to receive similar recent news articles.

Input is matched with pulled results either via keyword match or natural language processing. In the case where it is on, it will further study the semantic format of the headline to determine if it categorizes more precisely through machine learning.

The final output clearly states whether the news is true or not, along with supporting information such as matched headlines, sources, and confidence scores. This approach enables fast, precise, and scalable verification of news content in real time.

3.4 SYSTEM DESIGN



4. DATA COLLECTION

4.1 Dataset

We are using a movie review dataset for sentiment analysis for this NLP task. The dataset is normally publicly available from sources such as IMDb or Kaggle, with labeled reviews as actual or forged. There is a review (text data) and its sentiment label per row.

Example Dataset Format:

Review (Text)	Sentiment (Label)
"I loved this movie! It was so amazing!"	real
"The plot was boring, and the acting was poor."	fake

4.1.1 Data Preprocessing

4.1.2Clearing the Dataset (Preprocessing Steps)

Preprocessing involves removing noise from the text data by cleaning it so that it can be prepared for training. The following are typically used as part of the routine steps:

4.1.3 Lowercasing

Convert the entire text to lower case to normalize the input and prevent repetition of words based on differences in cases.

4.1.4 Removal of Special Characters and Numbers

Delete all the numbers, punctuation, and special characters from the text to deal with meaningful words.

4.1.5 Tokenization

Split the text into individual words (tokens) to analyze.

4.1.6Stopword Removal

Remove common words (e.g., "the", "and", "is") that do not have high-value meaning in text analysis.

4.1.7 API Fetching:

Data fetching from an API entails making HTTP requests to the API and getting back the response (typically in JSON format). Here is an example of how data can be fetched using Python with the requests library.

For this case, let's consider a public API such as the NewsAPI, which offers news articles from different sources.

5. MODEL TRAINING

After post-preprocessing, the next step is converting the text data into numerical features to be processed by machine learning algorithms. TF-IDF (Term Frequency-Inverse Document Frequency) will be used to transform the preprocessed text into vectors.

5.1 Model Selection

For the sentiment analysis, we can use a Logistic Regression classifier. It is simple and is appropriate for text classification.

5.2 Model Evaluation

After we have trained the model, we gauge how well the model performs by using standard measures of classification such as accuracy, precision, recall, and F1-score.

Example:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	1
1	1.00	1.00	1.00	1
accuracy			1.00	2
macro avg	1.00	1.00	1.00	2
weighted avg	1.00	1.00	1.00	2

6. Algorithms Implemented

6.1 Machine Learning Model Prediction Probability

The majority of ML classifiers (such as Logistic Regression, Naive Bayes, etc.) give a probability score for the prediction: Example:

P(REAL) = 0.87 → Confidence = 87%

P(FAKE) = 0.13 → Confidence = 13%

This value directly becomes your confidence level for the prediction.

Formula (Logistic Regression Output):

$$\text{confidence} = \frac{1}{1 + e^{-z}}, \quad \text{where } z = w \cdot x + b$$

6.2 Cosine Similarity from NLP Comparison

If you are also applying cosine similarity with dataset or real-time headlines, this similarity score can be added in order to narrow down the confidence:

Example:Cosine similarity = 0.72 (between input and real API news)

ML model predicts REAL with 0.84 probability

$$\text{Average: } \frac{0.72+0.84}{2} = 0.78 \rightarrow 78\% \text{ Confidence}$$

6.3 Final Confidence Calculation (Custom Logic)

If you want to merge both sources (NLP and ML), you can specify a weighted formula:

Example Formula:

$$\text{Final Confidence} = \alpha \cdot \text{ML Probability} + (1 - \alpha) \cdot \text{Cosine Similarity}$$

7. SYSTEM TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

8.IMPLEMENTATION & RESULT

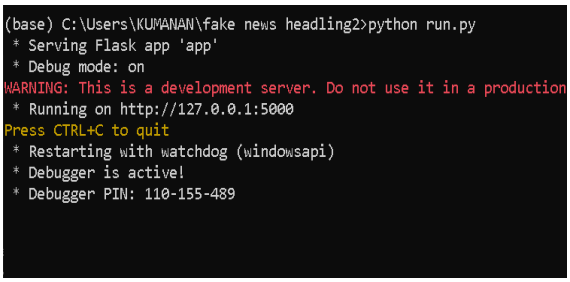


Fig 1.1 Anaconda cmd

8.1 Anaconda cmd

To run the Fake/Real News Detection System, open the Anaconda Prompt and optionally create a virtual environment to manage dependencies. Install all necessary libraries such as Flask, Requests, NLTK, and Scikit-learn. Navigate to your project directory containing the Flask app files and start the server using Python. Once running, access the web application through a browser at http://127.0.0.1:5000, where users can input headlines for analysis and receive a result indicating whether the news is real or fake.

Fig 1.2 web browser(UI)

8.2 User Interface (UI) and User Experience (UX)

The User Interface (UI) and User Experience (UX) of the Fake/Real News Headlines Identification System are built using a combination of HTML, CSS, and Flask templating. These components ensure that the application is both visually appealing and user-friendly.

Fig 1.3 web browser (output/result)

CONCLUSION

The Fake/Real News Headlines Identification System successfully demonstrates that real-time news verification can be achieved by integrating machine learning, natural language processing, and live news APIs into a Flask web application. By the choice of taking input from a user in the form of a headline and comparing with real, recent sources by API fetching, the system efficiently labels the news as REAL or FAKE along with a confidence measure to decide certainty of prediction. The project illustrates an effective way to combat misinformation with the help of pre-trained models and semantic comparison techniques. An accessible interface built using HTML and CSS supports simple and smooth experiences, while data processing and the predict logic is taken care of effectively with the backend. In short, this system is a wonderful resource in the ongoing mission

to provide news authenticity in the digital age and demonstrates the promise of AI-powered solutions towards enabling responsible information consumption.

FUTURE ENHANCEMENTS

Future enhancements to the Fake/Real News Headlines Identification System are incorporating multilingual capabilities, using more advanced ML models such as BERT for higher accuracy, and enabling user feedback for ongoing learning. The system can further be augmented using database storage, a mobile application variant, visual analytics dashboards, and detection of fake news sources. Such enhancements would enhance accuracy, useability, and flexibility and make the tool more potent in the war on misinformation.

REFERENCE

[1] Parikh, S. B., & Atrey, P. K. (2018, April). Media-Rich Fake News Detection: A Survey. In 2018 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR) (pp. 436-441). IEEE.

[2] Conroy, N. J., Rubin, V. L., & Chen, Y. (2015, November). Automatic deception detection: Methods for finding fake news. In Proceedings of the 78th ASIS&T Annual Meeting: Information Science with Impact: Research in and for the Community (p. 82). American Society for Information Science.

[3] Helmstetter, S., & Paulheim, H. (2018, August). Weakly supervised learning for fake news detection on Twitter. In 2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM) (pp. 274-277). IEEE.

[4] Stahl, K. (2018). Fake News Detection in Social Media.

[5] Della Vedova, M. L., Tacchini, E., Moret, S., Ballarin, G., DiPierro, M., & de Alfaro, L. (2018, May). Automatic Online Fake News Detection Combining Content and Social Signals. In 2018 22nd Conference of Open Innovations Association (FRUCT) (pp. 272-279). IEEE.

[6] Tacchini, E., Ballarin, G., Della Vedova, M. L., Moret, S., & de Alfaro, L. (2017). Some like it hoax: Automated fake news detection in social networks. arXiv preprint arXiv:1704.07506.

[7] Shao, C., Ciampaglia, G. L., Varol, O., Flammini, A., & Menczer, F. (2017). The spread of fake news by social bots. arXiv preprint arXiv:1707.07592, 96-104.

[8] Chen, Y., Conroy, N. J., & Rubin, V. L. (2015, November). Misleading online content: Recognizing clickbait as false news. In Proceedings of the 2015 ACM on Workshop on Multimodal Deception Detection (pp. 15-19). ACM.

[9] Najafabadi, M. M., Villanustre, F., Khoshgoftaar, T. M., Seliya, N., Wald, R., & Muharemagic, E. (2015). Deep learning applications and challenges in big data analytics. Journal of Big Data, 2(1), 1.

[10] Haiden, L., & Althuis, J. (2018). The Definitional Challenges of Fake News.