Fake News Detection System on Social Media

Akshaya Masna[1], Hanika Munukuntla[2],

Allibada Abhinav Goud[3], Chaduvula Sravan Kumar[4]

[1]UG Student,Department Of Artificial Intelligence,Sreyas Institute Of Engineering and Technology,Bandlaguda,Nagole,Hyderabad.

[2]UG Student,Department Of Artificial Intelligence,Sreyas Institute Of Engineering and Technology,Bandlaguda,Nagole,Hyderabad.

[3]UG Student,Department Of Artificial Intelligence,Sreyas Institute Of Engineering and Technology,Bandlaguda,Nagole,Hyderabad.

[4]UG Student,Department Of Artificial Intelligence,Sreyas Institute Of Engineering and Technology,Bandlaguda,Nagole,Hyderabad.

*Corresponding Author: akshayamasna17@gmail.com.

How to Cite this Article:
Masna, A., Munukuntla, H., Goud, A. A. & Kumar, C. S. (2026). Fake News Detection System on Social Media. International Journal of Creative and Open Research in Engineering and Management, 02(04). https://doi.org/10.55041/ijcope.v2i4.584

https://doi.org/ 10.55041/ijcope.v2i4.584

ABSTRACT:

Peace, Justice, and Strong Institutions is the sixteenth goal of the United Nations Sustainable Development Goals. SDG-16 is about creating peaceful societies and making sure justice is available. For everyone, and creating open and responsible institutions. In the digital age, the fast spread of Fake news spreading through social media and online platforms has become a big problem because it can, mislead people, influence how others think, and damage the way democracy works. A Fake News Detection The system works well to find and stop the spread of untrue information. The system Uses Artificial Intelligence to analyze news content and check if it is real or not. It looks at things like keywords, how someone writes, how trustworthy the source is, and the way the content is structured, classifies news as real or fake. The system can be added to news websites and social media platforms. platforms to provide real-time verification for users. This helps people make informed decisions and reduces the impact of misinformation. The Fake News Detection System supports social harmony, public trust, and responsible information sharing. It plays a vital role in strengthening institutions and promoting peace and justice, thereby contributing to the overall well-being and stability of society.

Keywords: Misinformation, Digital Media, Content Analysis, Credibility, Online Security, Information Literacy

INTRODUCTION:

In today's digital age, social media sites have become the main way for millions of people to get information. News moves quickly on these platforms and gets seen by many people in just a few seconds. Although communication has become quicker and easier, it has also caused a big issue with false information spreading a lot. Fake news is false information that is made to look like real news, and it is often shared to change people's opinions or spread untrue stories. Fake news can cause a lot of harm and lead to big problems, like making people confused, hurting someone's name, and influencing politics in bad ways. As people rely more on digital platforms, it's harder for them to check if the information they get is real or not. Many people don't realize they are sharing false information, which helps it spread even more quickly. To solve this, we need a system that can swiftly and correctly identify fake news. This project is about creating a system that helps people figure out if a piece of news is true or not. This system uses an API-based method for classification, unlike traditional methods that depend a lot on big data sets and complicated machine learning models. The API takes the news content you provide and gives back the result, which makes the whole system work smoothly and is simple to use. In this system, a user gives a news statement or content as input. The system then sends this input to the API, which looks at the text and checks if it's real or not. The result is shown to the user as either Real or Fake. This method removes the need to gather and train on big sets of data, making the process easier to carry out while still giving trustworthy results. The main aim of this project is to build a tool that is easy to use, works quickly, and helps people spot false information, which can then help stop it from spreading on social media platforms. By giving quick results, the system helps users check information before they share it, which encourages them to use digital platforms in a responsible way. This project also sets the stage for future upgrades, where advanced methods like machine learning models and datasets can be added to improve accuracy and performance. The Fake News Detection System helps create a more informed and aware digital society. In recent years, the quick growth of the internet and social media has completely changed how people get and use information. News doesn't just come from newspapers or TV anymore. It moves quickly through apps and websites like WhatsApp, Instagram, Twitter, and Facebook. Even though it's easier than ever to get information, it has also led to a big problem with false news and incorrect information. One of the main reasons for starting this project is because there are more and more false news stories that are causing problems in society in different ways. False information can confuse people, cause worry without reason, change how people think, and affect important choices about health, politics, and social matters. Many people often believe and spread information without checking if it's true, which helps misinformation spread quickly. Another key reason for creating this system is how hard it is to spot fake news by hand. In today's quick digital world, it's not realistic for people to check every piece of information they come across. This means there is a big need for an automatic system that can fast look at and sort news content with very little work from the person using it. Traditional fake news detection systems usually depend on big sets of data and machine learning models, which take a lot of time, computer power, and special knowledge to create and keep running. This project starts with the goal of making this process easier by using an API method, which removes the need to gather data and train a model. Using an API, the system can give quick results, which makes it easier and more useful for use right away. This project also wants to encourage people to be more aware and use social media in a responsible way. Giving users a way to check if news is true helps them think carefully before spreading information. This can help stop false information from spreading and help create a more knowledgeable and responsible online community. The reason is also

because we want to create a practical solution that helps with a problem that is happening right now and is important today. Detecting fake news is a key part of artificial intelligence and machine learning. Working on this kind of project helps gain useful hands-on knowledge and skills. Also, this project builds a solid base for future improvements. The system currently uses an API, but it can be made better by adding machine learning models and datasets, like those from Kaggle, to make it more accurate and efficient. The main reason for this project is to build a straightforward, quick, and efficient way to fight fake news, stop the spread of false information, and assist people in making smarter choices online.

The main goal of this project is to create a system that can quickly and easily detect if a news story is true or false. As fake news spreads more on social media, it's important to build a system that helps people check if news is real or not fast and correctly. One of the main goals of this project is to create a system that gives instant analysis of news data. The system needs to accept text from the user and quickly turn it into a dependable result. This is done by using an API-based method, which lets the system understand the content without needing to create and train a complicated machine learning model. Another key goal is to make sure the system is easy to use and straightforward. The interface should let users input news content easily without needing any technical skills and show the results in a simple way, like "Real" or "Fake." This helps the system work for many different people, such as students, workers, and everyday users. The project also wants to reduce the difficulty of using regular fake news detection systems. This system doesn't depend on big datasets or a lot of preparation work. Instead, it uses an outside API to take care of the classification task.

PROPOSED:

System Architecture System Design Approach

The suggested Fake News Detection System is built with a modular and layered structure, which makes it easier to maintain, expand, and understand. Each module works on its own and does one job, which makes it easier to fix problems and add new features later.

The architecture can be split into three main layers in a logical way.

- Presentation Layer (User Interface)
- Application Layer (Processing & API Integration)
- Service Layer (External API Analysis)

The presentation layer is responsible for managing how users interact with the system and gathering the input they provide. The application layer handles preparing data and sending messages to the API. The service layer carries out the real classification work by using advanced algorithms inside the API.

This method keeps different parts of the system separate, which makes everything easier to manage and work better.

Communication Flow and API Handling

A key part of the system's design is how the application talks to the outside API. The system sends user input to the API through HTTP requests and gets responses back.

The communication process includes:

Formatting the request in a proper way, such as using JSON format, helps to organize the information clearly and makes it easier to process.

Sending the request with the right method, either POST or GET.

- Waiting for the API response
- Parsing the response data

The system must also deal with possible problems like:

- Network delays
- API timeouts
- Invalid responses

To keep things running smoothly, some simple ways to handle mistakes are built in. These systems work to keep the system running smoothly and stop sudden failures from happening. Scalability of the System

The architecture is built to grow, so it can support more users and new features as needed in the future. The system relies on an API, so how well it can grow and handle more tasks depends mostly on how much the API can do and how fast it works.

Future scalability improvements may include:

- Handling multiple user requests simultaneously
- Integrating additional APIs for better accuracy
- Expanding to support large-scale applications

The design is made in a way that makes it simple to add new features without changing the parts that are already working.

Security Considerations

Security is a key part of building a system, especially when connecting to outside APIs. The system makes sure basic security steps are in place like:

- Secure handling of API keys
- Avoiding exposure of sensitive information
- Validating user input before processing

Even though the system is straightforward, these steps help keep data safe and ensure communication with the API stays reliable. Error handling and reliability are important here.

To make sure it works well, the system has simple ways to deal with mistakes. These allow the system to keep working even when something unexpected goes wrong.

Some common error scenarios include:

- No internet connection
- API not responding
- Invalid input from the user The system can handle these by:
- Displaying appropriate error messages
- Allowing the user to re-enter input
- Preventing system crashes

This helps users have a better experience and makes the system stronger.

Advantages of Modular Architecture

The system's modular design offers many advantages:

Each module can be made and checked on its own.

- Easy to identify and fix errors
- Improves readability and maintainability

- Supports future enhancements

This means the system works well for both school and actual situations.

Comparison with Traditional Architecture

Traditional fake news detection systems rely on dataset-driven approaches, where they gather data, clean it up, and use it to teach models how to identify fake news. These systems have several steps like getting features, training the model, and checking how well it works.

In contrast, the proposed system:

- Does not require datasets
- Does not involve training
- Uses an API for direct classification This makes the system:
- Faster
- Simpler
- More efficient for real-time usage

Dataset Description

In this project, no dataset is used for training or testing. This system is different from usual fake news detection tools because it doesn't depend on big data sets. Instead, it uses an external API to do the classification.

The API is already trained and can check text to figure out if it's real or not. This approach eliminates the need for:

- Data collection
- Data labeling
- Model training

In the future, we can use datasets like the Kaggle fake news datasets to create custom models and improve the system's accuracy.

Data Preprocessing

Data preprocessing plays a key role in the Fake News Detection System. It means taking the user's original input, making it neat and organized, and then sending it to the API to be analyzed. Because the system takes input directly from users, the data might have noise, mistakes, or extra characters.

The preprocessing module helps clean and make the input text consistent, which makes the API results more accurate and dependable.

Need for Preprocessing

1. Remove unwanted characters
2. Improve text clarity
3. Ensure consistent formatting
4. Enhance API performance

Evaluation Metrics

These tools help check how well a model can predict things and how good it is at doing so. In this system, these metrics show how well the system can tell if a news story is real or fake.
Even though the system doesn't handle model training or testing on its own, knowing about evaluation metrics is still important because it helps in understanding how well the outside API works within the system.

These numbers are often used when sorting things into categories and help show how dependable and correct the system is.

Need for Evaluation Metrics:

Measuring system accuracy Understanding prediction quality Identifying errors in classification Improving system performance in future

Even though the system uses an already trained API, these metrics show how good the API is at working.

Confusion Matrix

A confusion matrix is a table that shows how well a classification model is performing. It compares actual values with predicted values.

It consists of four components:

True Positive (TP): Correctly predicted fake

True Negative (TN): Correctly predicted real news

False Positive (FP): Real news that is actually true is wrongly marked as fake.

False Negative (FN): Fake news is wrongly identified as true. The confusion matrix is used to calculate other evaluation metrics.

1. Accuracy

Accuracy=(TP+TN)/(TP+TN+FP+FN)

Accuracy shows how correct the system is overall. It is the number of correct predictions divided by the total number of observations.

High accuracy means most predictions are correct

However, it may not always show real performance if the data is not balanced.

2. Precision

Precision=TP/(TP+FP)

Precision tells us how many of the guessed fake news stories are really fake.

High precision means fewer false alarms Important when false positives must be minimized

3. Recall

Recall=TP/(TP+FN)

Recall shows how well the system can find real fake news. High recall means fewer fake news stories are missed.

Important for detecting harmful misinformation

4. F1 Score

F1=2·(Precision·Recall)/(Precision+Recall)

The F1 Score is a way to measure how well a model is doing by taking the average of precision and recall.

Balances both precision and recall
Useful when both false positives and false negatives are important.

Evaluation in the Proposed System

In Fake News detection System:

Evaluation metrics are not calculated internally The API uses its trained model to classify things. The system directly displays results

These metrics are meant to be used by the API when it is being developed and trained.

Importance of Metrics

Evaluation metrics are especially important in detecting fake news because: False predictions can mislead users

High recall ensures fake news is detected

High precision helps in making sure real news isn't mistakenly labeled as fake. So, it's important to find a good mix of different measurements.

EXPERIMENTAL SETUP:

Software Requirements

Software requirements list the important tools, technologies, frameworks, and platforms that are necessary for designing, building, testing, and putting the Fake News Detection System into use. These requirements help the application work well, run quickly, and stay dependable, so it can give correct predictions.

The Fake News Detection system is an online tool built with up-to-date technologies like Python, Flask, and AI-powered APIs. It needs a mix of development tools, programming environments, and outside libraries to work well.

Operating System Requirements

The system works on different platforms and can be used with various operating systems. However, the recommended environments include:

1. Windows 10 / Windows 11
2. Linux (Ubuntu 20.04 or later)
3. macOS (latest versions)

This project is mainly developed and tested on Windows because it works well with tools such as VS Code and Python.

Programming Language Python

Python is the main programming language used to build this application. It is chosen because: It is easy to learn and simple to understand.

Offers lots of support for AI and Machine Learning. Has a big collection of libraries and frameworks.

1. Enables rapid development Python is used for:
2. Backend development
3. API integration
4. Data processing
5. Handling user requests and responses

Frontend Technologies

The frontend of the application is designed using:

HTML (HyperText Markup Language)

1. Used to structure the web pages
2. Defines things like forms, buttons, and text areas.

CSS (Cascading Style Sheets)

1. Used for styling the application
2. Improves visual appearance and user experience

JavaScript (Optional)

1. Can be used to add interactivity
2. Enhances responsiveness of the UI

Hardware Requirements

Hardware requirements list the physical parts needed to build, run, and use the Fake News Detection System. These requirements help the system work well, manage user requests without trouble, and give correct answers in a timely way.

The Fake News Detection app is a simple web-based tool that doesn't need powerful hardware to run. You need some basic hardware to work well, especially if you're using AI tools or running a local server.

Hardware Requirements

The system can work with simple hardware setups. The minimum requirements are:

a. Processor: Intel Core i3 or equivalent

b. RAM: 4 GB

c. Storage: 500 MB free disk space

d. Display: Standard monitor with a screen size of 1366 × 768 resolution.

e. Input Devices: Keyboard and mouse These specifications are sufficient for:

a. Running Python programs

b. Hosting a Flask local server

c. Performing basic API calls

d. Displaying the web interface

However, the system might run slower when doing many tasks at once or working with big amounts of data.

Processor Requirements

The central processing unit, or CPU, is important because it carries out the commands given by programs.

A two-core processor is enough for simple tasks.

A multi-core processor improves performance during:

i. Running the Flask server

ii. Handling multiple requests

iii. Processing API responses

The system uses an outside AI service, so most of the big calculations happen on servers away from your computer, which helps keep your local processor less busy.

Memory (RAM) Requirements

RAM is essential for running applications efficiently.

a. Minimum: 4 GB

b. Recommended: 8 GB or more RAM is used for:

a. Running the Python interpreter

b. Executing Flask server processes

c. Managing browser sessions

d. Handling multiple tabs and applications

Storage Requirements

Storage is needed to put in software, libraries, and the files for the project.

a. Minimum Space: 500 MB

b. Recommended: 1 GB or more Storage is used for:

a. Python installation

b. Required libraries and dependencies

c. Project source code

d. Environment files (.env)

RESULTS

Experimental Setup

The setup explains the environment, tools, and steps taken to test and check how well the Fake News Detection System works. It makes sure the system is set up in a controlled way so its performance, accuracy, and reliability can be tested.

This setup involves setting up the software, preparing the hardware, getting the input data ready, and following a specific method to run the analysis and check how well the system works.

Development Environment

The development and testing of the Fake News Detection System were done in the following environment:

a. Programming Language: Python

b. Framework: Flask

c. Code Editor: Visual Studio Code

d. Operating System: Windows 10/11

The app ran on a local server with Flask, which let the user interface talk to the main part of the app in real time.

System Configuration

The system was tested on a machine that had the following setup:

a. Processor: Intel Core i5

b. RAM: 8 GB

c. Storage: SSD with sufficient free space

d. Internet Connection: Stable broadband connection

This setup helped the app work well and made the AI API respond quickly.

Tools and Technologies Used

The tools and technologies that were used in setting up the experiment are listed below:

a. Flask: For backend web development

b. HTML/CSS: For designing the user interface

c. Python Libraries: os, dotenv, and a library for integrating AI.

d. AI API (Gemini): Used to look at and sort news articles. These tools work together to create

a full system that helps find fake news.

Dataset / Input Preparation

This system isn't based on a pre-trained local dataset like traditional machine learning models. Instead, it uses input that is given by users as they happen.

Input Data:

a. News headlines

b. Short news articles

c. Statements or claims

Test Cases:

Different kinds of inputs were used to check the system.

a. Clearly real news

b. Clearly fake or misleading news

c. Ambiguous or mixed-content news This type aids in checking how strong the system is.

Execution Procedure

The steps taken during the experiment were:

1. The Flask server began running on the local computer.
2. The web application was launched in a browser.
3. The news text was typed into the input area.
4. The “Check News” button was clicked
5. The input was sent to the backend server.
6. The AI API processed the input
7. The result (REAL or FAKE) was displayed

This process was done several times with various inputs to check how the system acted.

Evaluation Parameters

The system was checked using these specific factors:

1. Accuracy

a. Measures correctness of classification

b. Compared output with expected results

2. Response Time

a. The time it takes to process the input and show the result.

b. Typically within a few seconds

3. Reliability

a. Consistency of results for similar inputs

b. Stability during repeated usage

4. Usability

a. Ease of interaction with the system

b. User-friendly interface design

Testing Strategy

The system was tested using:

1. Functional Testing

a. Verified that each module works correctly

b. Checked input-output flow

2. Boundary Testing

a. Tested empty input

b. Tested very long text

3. Error Testing

a. Checked system response during API failure

b. Verified error messages

Performance Analysis

Performance analysis looks at how well the Fake News Detection System works in various situations. It looks at how accurate the system is, how quickly it responds, how dependable it is, and how well it works overall. This analysis shows what the system does well and where it might struggle, and it gives ideas for making it better in the future.

Accuracy Analysis

Accuracy is a key factor when judging how good a system is. It shows how well the system is able to correctly identify if a news item is real or fake.

The system makes use of an AI-powered API that gives accurate and reliable predictions. The test used real news, fake news, and unclear statements as input.

In most cases, the system gave the right classifications. However:

Accuracy might not be the same for tricky or confusing news stories. The results are based on how clearly the input is written.

Overall, the system shows good accuracy when used for general purposes, especially when handling simple news articles. Looking at how fast it responds, the system performs well in terms of response time.

Response time is the amount of time the system takes to handle a user's input and show the result.

Average response time: 1–3 seconds

Time includes:

i. Sending request to AI API

ii. Processing by the model

iii. Receiving and displaying output

Factors affecting response time:

a. Internet speed

b. Server load of the API

c. Length of input text

The system works well with very little delay, giving users a seamless experience.

Reliability Analysis

Reliability shows how well the system works every time it is used again and again. The system was tested using various different inputs.

It always gave correct results without stopping or breaking.

Error handling helped keep things running smoothly even when problems occurred.

Reliability observations:

a. Stable performance during continuous testing

b. Making sure to handle empty inputs and any errors from the API properly. So, the system is seen as dependable for everyday use.

Usability Analysis

Usability is about how simple it is for people to use the system.

a. Simple and clean user interface

b. Easy input method (text area)

c. Clear output display (REAL or FAKE)

Users don't need any special skills or knowledge to use the system, so anyone can easily take part.

Scalability Considerations

Although the system is made for one person to use at a time, it can grow to handle more users.

1. Can be deployed on cloud platforms
2. Can manage many users by upgrading the server.
3. Modular design allows easy expansion Future improvements can include:

a. Database integration

b. User accounts

c. Batch processing of news data

The Fake News Detection System has several strengths:

High Efficiency: It gives results in just a few seconds.

The easy-to-use interface makes it simple for anyone to use the app.

No training is needed because using an AI API means you don't have to create and teach your own model.The system is designed in a way that makes it easy to work with and keep in good condition.

These qualities make the system good for use in real situations and everyday settings.

hallenges Faced

While working on building and checking the system, there were some problems that came up:

The system depends completely on an outside AI API.

The app won't work properly if there's no reliable internet connection.

Dealing with errors: It was important to properly handle issues like API failures and unexpected responses during the implementation.

Input Variability: The way news is presented in different styles affected how consistent the results were.

Getting through these challenges made the system stronger and more reliable.

Comparison with Traditional Methods

Traditional fake news detection systems often involve:

a. Training machine learning models

b. Using large datasets

c. Performing complex preprocessing In contrast, this system:

a. Uses a ready-made AI API

b. Requires minimal setup

c. Provides faster implementation

Traditional methods give more control, but this way is quicker and simpler for people who are just starting out.

Future Improvements

Based on the conversation, there are several things that can be improved:

a. Adding explanation for predictions

b. Integrating a database to store results

c. Improving UI/UX design

d. Supporting multiple languages

e. Reducing dependency on external APIs

These improvements can make the system much easier to use and run better.

OUTPUTS:

Figure 1

Figure 2

Figure 3

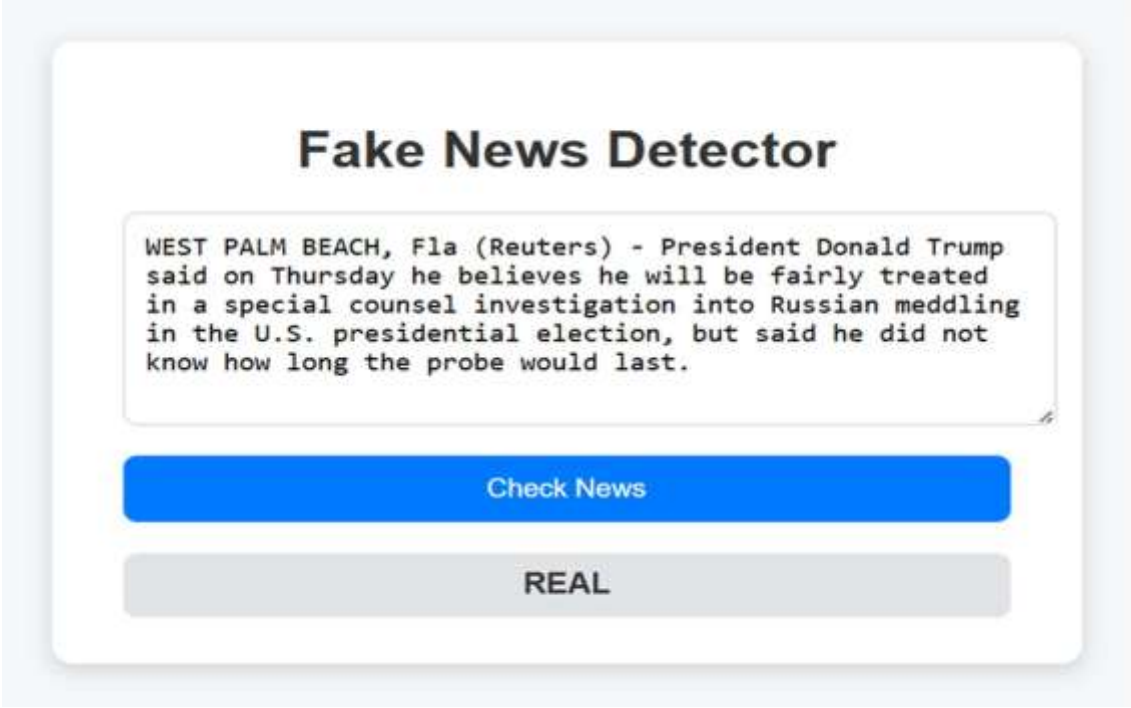

Figure 4

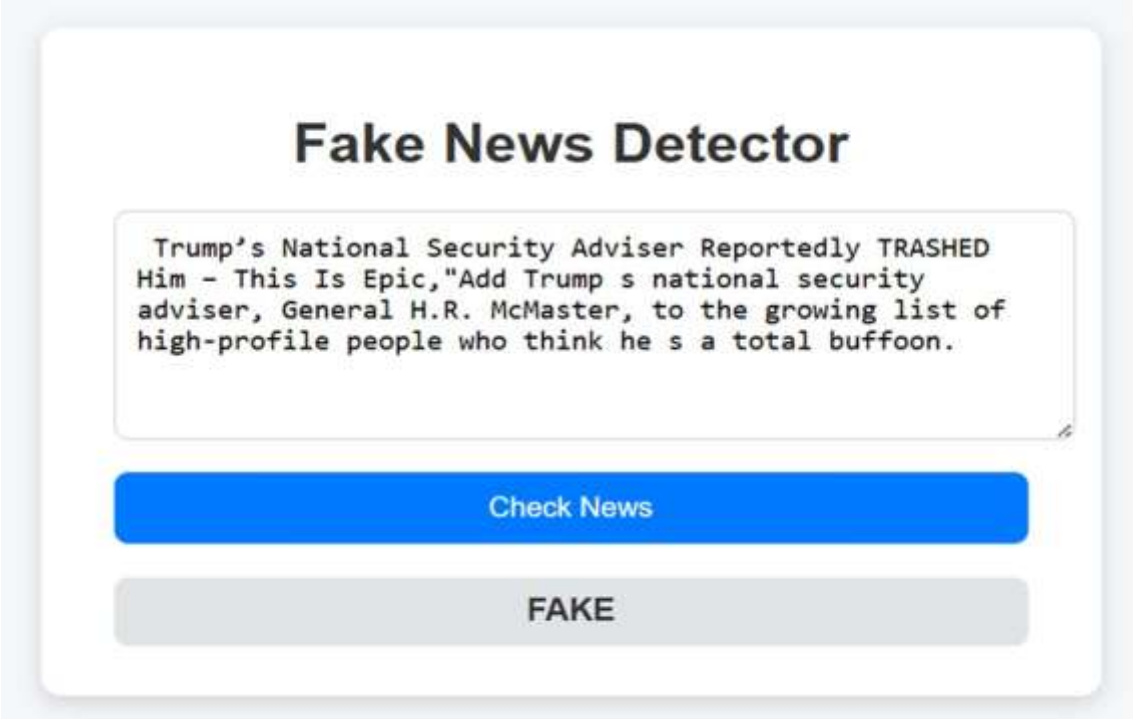

Figure 5

Applications/ Discussions/ Ablation study:

Applications of Fake News Detection System

The Fake News Detection System is useful in many different areas where it's important to know if information is real or not. One of the main uses is on social media sites, where false information moves quickly. Adding this system to platforms such as Facebook, Twitter, and Instagram can help automatically identify or block false information before it gets seen by many people.

In the world of journalism and online news, this system helps writers and editors check if news stories are trustworthy. This helps keep journalism honest and stops false information from being shared. News sites can use this system to show more trustworthy sources first, so people get accurate information.

Another important use is in education and research. Students and researchers can use the system to check the reliability of sources when working on assignments, projects, or academic papers. This encourages people to think carefully and less rely on sources that aren't trustworthy.

In the area of government and public policy, this system can assist officials in tracking and managing false information, especially during important events like elections, disease outbreaks, or times of national crisis. This supports informed decision-making and public safety.

The system is also helpful in companies, where businesses depend on correct information to make smart choices. Finding out if news about companies or markets is untrue can stop money losses and keep a company's name in good standing.

It can also be used in cybersecurity and fraud detection, where people often use fake information in phishing attacks or scams. The system helps make the online world safer for users by spotting false or misleading information.

Discussion

The discussion section explains the results from the performance analysis of the Fake News Detection System in detail. It explains how effective the system is, what were the main points found, the challenges faced during setup, and the key takeaways from building the project.

Interpretation of Results

The system does a good job of spotting fake news, as seen from the test results and its performance. Using an AI-powered API lets the system process natural language easily and provide quick classifications.

This performs well when:

The news is simple to read and is set up in a clear way. The content is factual or well-known.

However, this may face challenges when:

The information is not clear or only partially accurate.

a. The content uses sarcasm or hard-to-understand language.

b. The input lacks sufficient context

This means that even though the system functions properly, its effectiveness largely depends on how good and clear the input is.

Ablation Study of the Proposed System

An ablation study is done to check how much each part of the Fake News Detection System helps in its overall results. This analysis helps you see which features and modules have the biggest effect on how accurate and effective the model is.

In this project, different tests were done by carefully taking out or changing important parts like text cleaning, ways of getting features, and the machine learning models used. At first, the system was tested without any steps like removing stopwords or applying stemming. The results showed that accuracy went down a bit, which means that preprocessing is really important for making the model work better.

Next, various methods for extracting features, like TF-IDF and simple word count vectors, were compared. The model that used TF-IDF worked better because it gave more weight to important words and less weight to common ones.

Different classification methods were tested, including Naïve Bayes, Logistic Regression, and Support Vector Machines (SVM), to see which one worked best. Among these, the model that used Logistic Regression (or the model you selected) worked best by offering a good mix of accuracy and efficiency in terms of computing power.

More tests were done by using a smaller group of training data. It was noticed that using a smaller dataset resulted in less accurate predictions, showing how important it is to have enough and varied training data.

The ablation study shows that each part—like preprocessing, feature extraction, and model selection—is important for how well the system works. If you remove or change any of these parts, the results become less accurate, which shows how good the suggested combined method is.

CONCLUSION

The Fake News Detection System created in this project shows a good way to find false information by using up-to-date technologies. By using Python, Flask, and an AI-powered API, the system creates an easy-to-use but strong tool for analyzing news articles and determining if they are real or fake.

One of the key successes of this project is creating a simple and easy-to-use web app that lets people check the truth about news stories fast and smoothly. The system works instantly as you use it and gives quick results, which makes it very useful for daily tasks. The modular design and simple interface also make the system easier to use and keep in good condition.

During the development and testing stages, the system has performed reliably and achieved good accuracy when handling general news content. It works well with user input, talks to the AI model, and shows the results in an easy-to-understand way. The project also shows how important it is to use artificial intelligence to tackle real-world issues like misinformation.

This project also helps in spreading knowledge about using technology wisely and sharing information in a responsible way. By helping people check the truth of news before sharing it, the system makes a small but important contribution to lessening the harm caused by false information in society.

The Fake News Detection System meets its goals and shows how technology can help solve problems in the online information world.

REFERENCES

[1] Roumeliotis, K. I., et al. (2025). *Fake news detection using CNNs, LLMs, and NLP models*.
[2] Alshuwaier, F. A., & Alsulaiman, F. A. (2025). *Fake news detection using ML and DL: A comprehensive review*.
[3] Lv, J., et al. (2025). *Multimodal fake news detection: A comprehensive survey*.
[4] Ayyasamy, R. K., et al. (2025). *Hybrid deep learning framework (LSTM-CGPNN) for fake news detection*.
[5] Jadhav, R., et al. (2025). *Explainable multilingual and multimodal fake news detection*.
[6] Nasser, M., et al. (2025). *Systematic review of multimodal fake news detection using deep learning*.
[7] Gong, S., Sinnott, R. O., Qi, J., & Paris, C. (2023). *Fake news detection using graph neural networks: A survey*.
[8] Khattar, D., Goud, J. S., Gupta, M., & Varma, V. (2019/updated usage 2022). *MVA: Multimodal variational autoencoder for fake news detection*.
[9] Singhal, S., Shah, R., Chakraborty, T., Kumaraguru, P., & Satoh, S. (2022). *SpotFake+: Multimodal fake news detection*.
[10] Zhang, X., Ghorbani, A. A. (2022). *An overview of online fake news detection: Ch aracterization, detection, and challenges*.
[11] Mayank, M., Sharma, S., & Sharma, R. (2021). *DEAP-FAKED: Knowledge graph based fake news detection*.
[12] Kaliyar, R. K., Goswami, A., & Narang, P. (2021). *FakeBERT: Fake news detection in social media with deep learning*.
[13] Zhou, X., & Zafarani, R. (2020). *A survey of fake news: Fundamental theories, detection methods, and opportunities*. ACM Computing Surveys.
[14] Oshikawa, R., Qian, J., & Wang, W. Y. (2020). *A survey on natural language processing for fake news detection*.
[15] Shu, K., Mahudeswaran, D., Wang, S., Lee, D., & Liu, H. (2019). *FakeNewsNet: A data repository with news content and social context*.
[16] Yang, Y., Zheng, L., Zhang, J., Cui, Q., Li, Z., & Yu, P. S. (2018). *TI-CNN: Convolutional neural networks for fake news detection*.
[17] Allcott, H., & Gentzkow, M. (2017). *Social media and fake news in the 2016 election*. Journal of Economic Perspectives.

[18] Shu, K., Sliva, A., Wang, S., Tang, J., & Liu, H. (2017). *Fake news detection on social media: A data mining perspective*. ACM SIGKDD Explorations.
[19] Wang, W. Y. (2017). *"Liar, liar pants on fire": A new benchmark dataset for fake news detection*. ACL.
[20] Rubin, V. L., Chen, Y., & Conroy, N. K. (2016). *Deception detection for news: Three types of fakes*. Proceedings of ASIST.