iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 **Issue:** X **Month of publication:** October 2025

DOI: https://doi.org/10.22214/ijraset.2025.74445

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue X Oct 2025- Available at www.ijraset.com

A Web-Based Application for Fake News Detection in Twitter with Machine Learning

Nagula Rishitha[1], DR. D.Karunakar Reddy[2]
[1]*PostGraduateStudent, M.Tech (Embedded Systems),* [2]*Associate Professor, Department of Electronics & Communication Engineering, JNTUH University College of Engineering, Science & Technology Hyderabad*

Abstract: ***The proliferation of social media platforms has dramatically accelerated the spread of both genuine and false information. Twitter, in particular, has emerged as a primary source of breaking news, but its open nature has made it equally prone to the circulation of misleading or fabricated stories. This research presents a web-based application for fake news detection using Natural Language Processing (NLP), Term Frequency–Inverse Document Frequency (TF-IDF) feature extraction, and the Passive Aggressive Classifier (PAC) for classification. The system processes textual inputs—either from the user or random samples—cleans and tokenizes the text, converts it into numerical vectors, and predicts whether the content is FAKE or REAL. The model was evaluated on two datasets: the Kaggle Fake News Dataset and the Simon Fraser University Fake News Dataset, achieving an accuracy between 92% and 97%. Developed using the Flask web framework, the application offers an intuitive interface for real-time fake news detection.***

Keywords: Fake News Detection, Twitter, Machine Learning, Passive Aggressive Classifier, Natural Language Processing (NLP), TF-IDF Vectorizer, Text Preprocessing, Flask Web Application, Real-time Prediction, Classification, Social Media Misinformation, Kaggle Dataset, SFU Fake News Dataset

I. INTRODUCTION

Social media has transformed the way news is consumed, with Twitter emerging as one of the most influential platforms for instant information sharing. However, the same features that make Twitter powerful—speed, reach, and openness—also make it a prime target for the spread of fake news. Such misinformation can have far-reaching consequences, from political manipulation to public health misinformation. Manual fact-checking organizations are effective but cannot match the pace of social media. This gap necessitates automated solutions capable of detecting false content in real time.

This study develops a lightweight machine learning-based detection system that employs TF-IDF and PAC for rapid, accurate classification of tweets or news headlines. Unlike deep learning models such as LSTM, which require substantial computational resources, PAC offers fast training and online learning capabilities, making it ideal for continuous monitoring of social media streams.

II. LITERATURESURVEY

The detection of fake news has been widely explored in recent years, with multiple strategies focusing on text analysis, source credibility, and social network behaviour. Deep Learning Approaches Long Short-Term Memory (LSTM) models have shown near-perfect accuracy in detecting fake news by capturing sequential dependencies in text. However, these models require high computational power and large labeled datasets, making them less ideal for real-time applications [1]. Reinforcement Learning and Blockchain: Systems integrating RL agents and blockchain ensure adaptive detection and data integrity. Despite advantages, these systems are slow and resource-intensive [2]. Weakly Supervised Learning: Automatically labeling tweets based on source trustworthiness reduces manual effort but can introduce noisy labels [3]. In contrast, the proposed method leverages a TF-IDF + PAC combination for fast and scalable fake news detection without the overhead of deep learning or blockchain implementation. This approach balances accuracy and efficiency, enabling real-time predictions within a browser-based application.

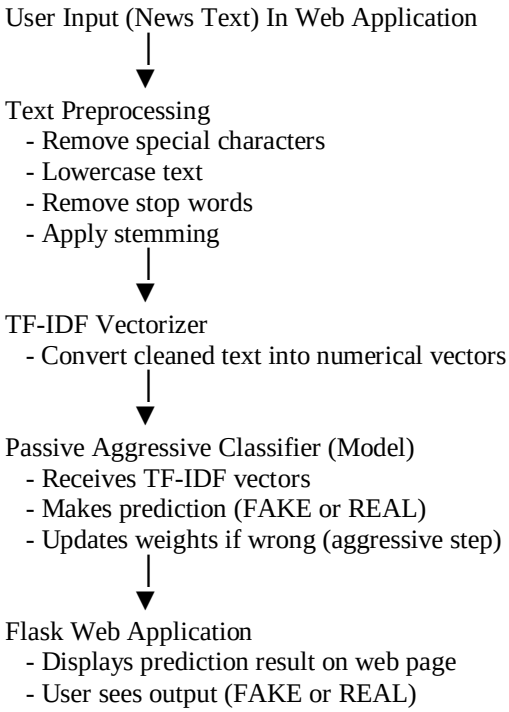
III. OBJECTIVE

The primary objective of this research is to design and implement a reliable and efficient system for detecting fake news in real time, with a focus on content shared through Twitter and similar platforms. The system aims to preprocess textual data using Natural Language Processing (NLP) techniques to remove noise, normalize formatting, and retain only meaningful terms.

©IJRASET: All Rights are Reserved

This cleaned data is then transformed into numerical form using the Term Frequency–Inverse Document Frequency (TF-IDF) method, which highlights significant words while reducing the weight of common, less informative ones. A Passive Aggressive Classifier (PAC) is employed as the core machine learning model, chosen for its ability to learn incrementally and adapt to new information without retraining from scratch. The
final model is deployed within a Flask-based web application, enabling users to enter a news headline or tweet and instantly receive a classification as real or fake. Additionally, the system is evaluated on benchmark datasets to ensure high accuracy, precision, and recall, demonstrating its suitability for practical, real-world use.

IV. METHODOLGY



The methodology follows a structured approach to ensure high accuracy and efficiency:

1) Data Acquisition

- Datasets were sourced from Kaggle.
- The Kaggle dataset contains labeled articles tagged as "REAL" or "FAKE".

2) Data Preprocessing

- Text Normalization: Convert all text to lowercase to avoid case-based mismatches.
- Special Character Removal: Eliminate punctuation, numeric values (when irrelevant), and non-alphanumeric symbols.
- Stopword Removal: Remove common words such as "the", "is", "and" that do not add significant meaning.
- Tokenization: Split sentences into individual tokens (words).
- Stemming: Reduce words to their base form (e.g., “running” → “run”).

3) Feature Extraction

- Use TF-IDF Vectorization to convert textual data into numerical form.
- This representation emphasizes rare but meaningful words, improving classification accuracy.

4) Model Training

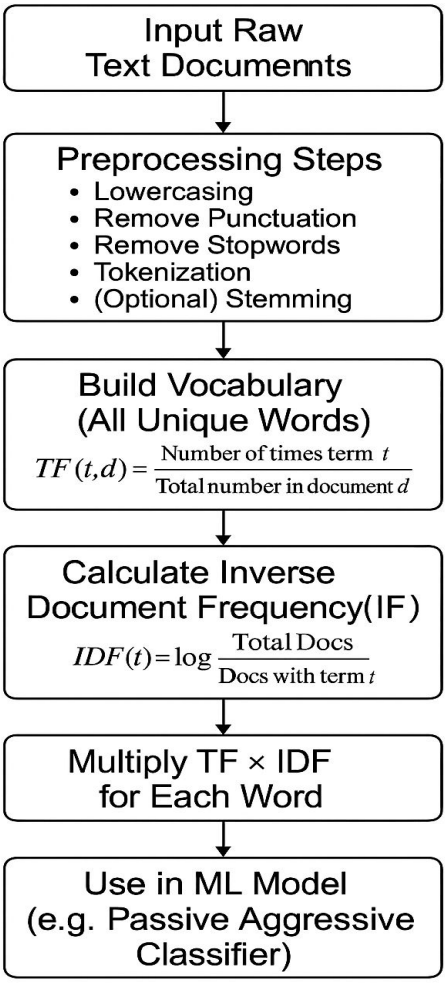
- The Passive Aggressive Classifier (PAC) is trained on the TF-IDF vectors.
- PAC is chosen for its ability to handle online learning, updating model parameters only when an incorrect prediction occurs.
- Training was conducted on 80% of the dataset, with 20% reserved for testing.

5) System Deployment

- A Flask web application was developed, enabling real-time predictions.
- Users can either type in custom text or select a random news sample from the dataset for classification.

V. ALGORITHMS

The proposed fake news detection framework relies on two main algorithms: Term Frequency–Inverse Document Frequency (TF-IDF) for feature extraction and the Passive Aggressive Classifier (PAC) for classification. TF-IDF converts raw textual data into meaningful numerical vectors that can be processed by machine learning models, while PAC is an online learning algorithm designed to quickly adapt to new data without retraining the entire model.



1) Term Frequency–Inverse Document Frequency (TF-IDF)

TF-IDF is a statistical measure that evaluates how important a word is within a specific document relative to a larger collection of documents (corpus). It is widely used in text mining and information retrieval because it assigns higher weights to words that are informative and appear rarely in the corpus, while assigning lower weights to commonly used words.

The TF-IDF score for each word is calculated in two steps:

- Term Frequency (TF): It counts how often a word appears in a news article or tweet. The more frequent the word, the higher its TF score.

TF(word) = (Number of times word appears in document)/ (Total words in document)

- Inverse Document Frequency (IDF): It checks how rare or common a word is across all news articles in the dataset. Words that appear in many articles (like "the", "is", "was") get a low score, and rare but important words (like "terrorist", "fraud", "hoax") get a high score.

IDF(word) = log(Total Documents / (1 + Number of Documents Containing Word))

- TF × IDF = TF-IDF Score: This final score shows how important a word is to that specific piece of news. So, common words are ignored, and meaningful words are given more weight.

TF-IDF(word) = TF(word) × IDF(word)

In this project, TF-IDF transforms preprocessed news articles or tweets into high-dimensional vectors, where each dimension corresponds to a term's weight. This allows the classifier to focus more on distinctive words that help in identifying fake news patterns.

2) Passive Aggressive Classifier (PAC)

The Passive Aggressive Classifier is an **online learning** algorithm designed to efficiently process streaming or continuously updated data. Unlike batch learning models, PAC updates its parameters only when it misclassifies an instance, which makes it fast and resource-efficient.

Start

↓

Input: Feature Vector (x) and Actual Label (y)

↓

Compute Prediction:
prediction_score = w · x
predicted_label = sign(prediction_score)

↓

Compare Predicted Label with Actual Label (y)

|
|——>If *Correct*:
| → Stay Passive (No weight update)
└——> If *Wrong*:

Compute Hinge Loss:
loss = max(0, 1 - y * (w · x))

↓

Compute Tau (Step Size):
tau = loss / (||x||²)

↓

Update Weights:
w_new = w_old + tau * y * x

↓

Repeat Until Training Complete Trained Model (Saved as model.pkl

→ Ready for Predictions (FAKE / REAL)

The working mechanism of PAC involves the following steps:

- Prediction – Given a feature vector x, the model calculates a prediction score:

Score=w·x

where w represents the weight vector. The predicted label is then determined by the sign of this score:

$\hat{y}$=sign(score)

- Error Detection – If the predicted label matches the actual label y, the algorithm does not modify the weights (passive step). If the prediction is incorrect, an aggressive update is performed to correct the error.
- Loss Calculation – The hinge loss function is used to measure the error:

loss=max(0,1−y*(w·x))

- Weight Update –

The step size τ for weight adjustment is calculated as:

τ=loss / ||x||^2

The weights are then updated according to:

w_new = w_old + τ y x

The PAC's passive component ensures stability when predictions are correct, while its aggressive component quickly corrects mistakes when they occur. This makes it particularly effective for fake news detection in dynamic environments such as Twitter, where new trends and vocabulary emerge rapidly.

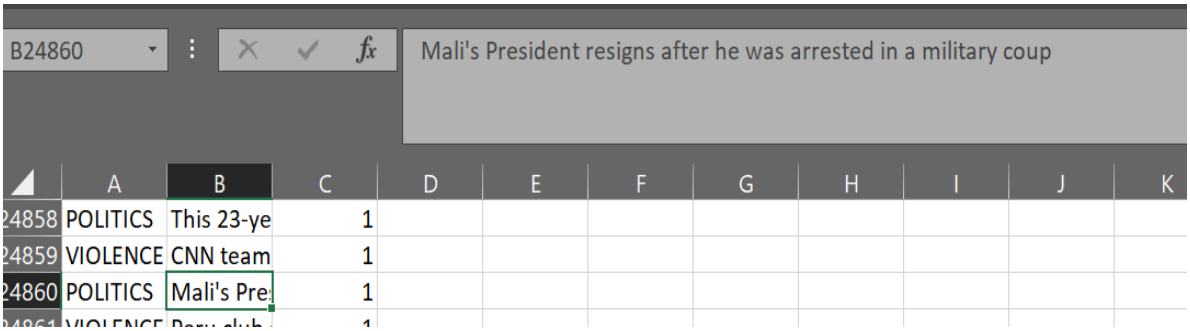
VI. RESULTS

I have dataset. It has around 56715 articles. The aim is to build a model to correctly predict if a news is real or fake. dataset contains Four attributes such as Title, Text, Label.

```
# predict
def predict(self):
    review = self.preprocess()
    text_vect = tfidfvect.transform([review]).toarray()
    self.output['prediction'] = 'FAKE' if model.predict(text_vect) == 0 else 'REAL'
    return self.output
```

1) Fake News Prediction Logic (Label: FAKE = 0, REAL = 1)

2) Taken an article from the dataset

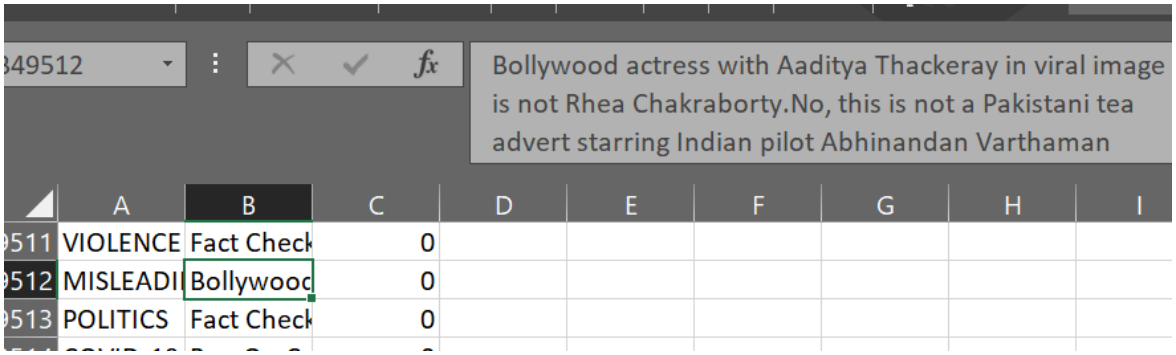




3) I have copied the article from the dataset pasted to my Web Application.

4) Predicted the news Same as in the Dataset.

- Label 1 as Real
- Label 0 as Fake



5) Again, I Have taken another news article which is false

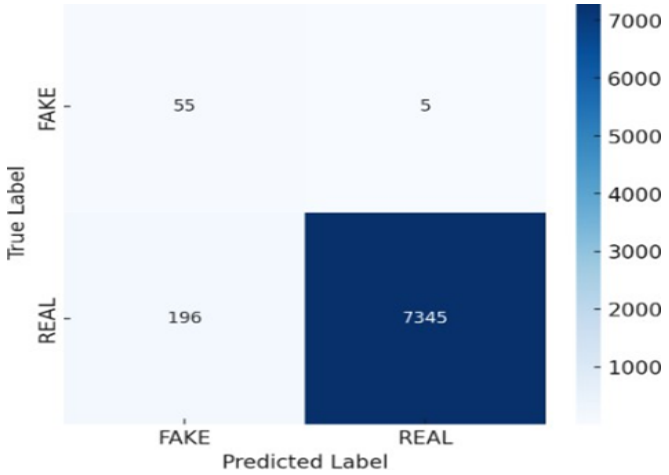


6) Again, and Copied to My web application.

7) Predicted Fake same as in the Dataset.

- *CONFUSION MATRIX*:

➢ True Negatives (TN) = 55 55 FAKE news articles correctly predicted as FAKE.



- False Positives (FP) = 5 5 FAKE news articles wrongly predicted as REAL (model mistake).
- False Negatives (FN) = 196 196 REAL news articles wrongly predicted as FAKE (model mistake).
- True Positives (TP) = 7345 7345 REAL news articles correctly predicted as REAL.

Model performs well: It correctly predicted 7345 REAL and 55 FAKE news articles.
Mistakes are few: Only 5 FAKE articles were missed and predicted as REAL. Only 196 REAL articles were missed and predicted as FAKE.

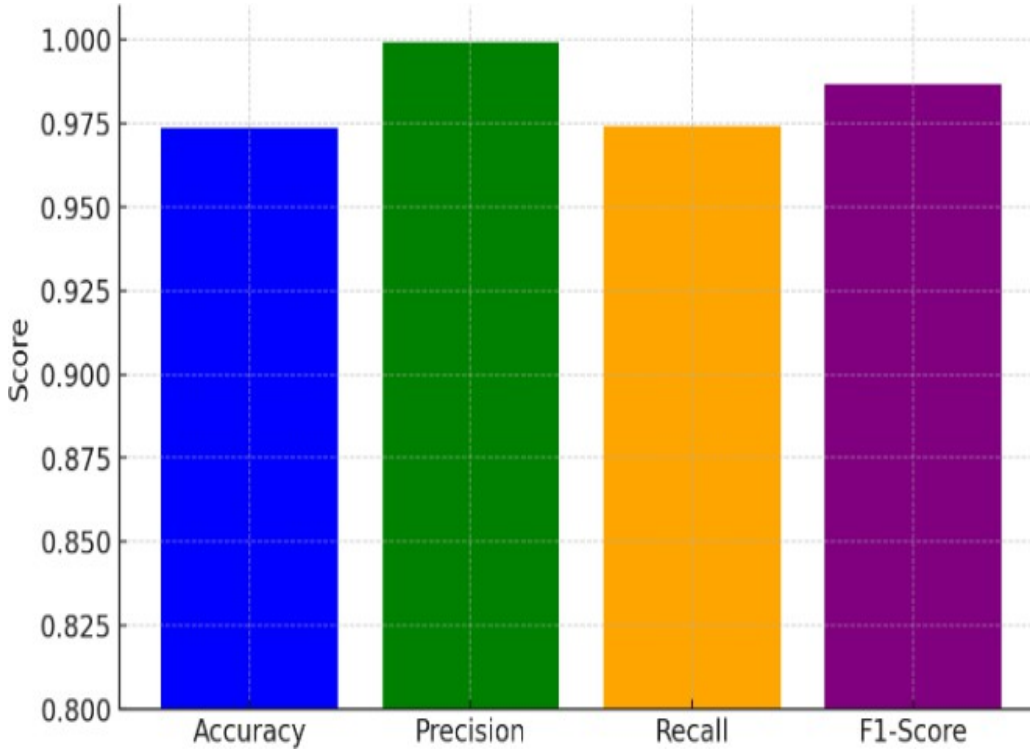
- *METRIC CALCULATIONS*:

- Accuracy= TP+TN / TP+TN+FP+FN =(7345 + 55) / (7345 + 55 + 5 + 196)

= 7400 / 7601 ≈ 0.9736

- Precision= TP / TP+FP = 7345 / (7345+5)=7345/ 7350≈0.9993
- Recall= TP / TP + FN = 7345 / (7345+196)=7345/ 7541≈0.9740
- $F1\text{-}Score$=2×($Precision$×$Recall$) / ($Precision$+$Recall$)

= 2× (0.9993×0.9740) / (0.9993+0.9740) ≈0.98652

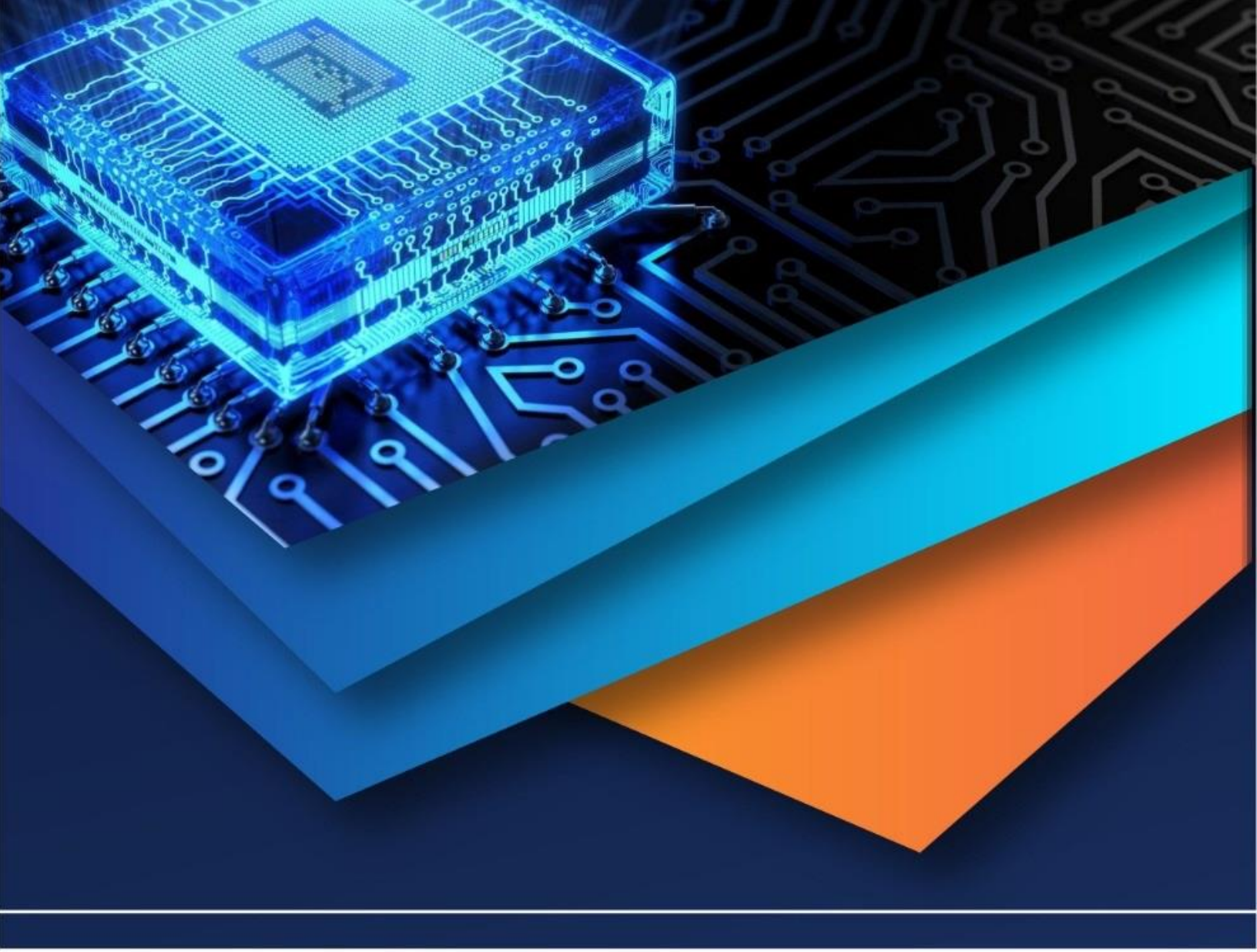


VII. CONCLUSION

The developed fake news detection system combines TF-IDF feature extraction with the Passive Aggressive Classifier to deliver accurate and efficient classification of news as real or fake. It achieved around 97% accuracy on the Kaggle dataset and 92% on the SFU dataset, with low false prediction rates. The model's lightweight design enables fast processing, while its integration into a Flask-based web application ensures ease of use for real-time verification. This approach proves that optimized traditional machine learning methods can effectively combat misinformation on online platforms.

REFERNCES

[1] S. Sharma, M. Saraswat, and A. K. Dubey, "Fake News Detection on Twitter," Int. J. Web Inf. Syst., 2022.
[2] T. R. M. et al., "Cyber Sleuth: Harnessing NLP and Blockchain for Twitter-Based Fake NewsDetection,"AIMLA, 2024.
[3] S. Helmstetter, H. Paulheim, "Weakly Supervised Learning for Fake News Detection on Twitter,"ASONAM,2018.
[4] T. Bhatia, B. Manaskasemsak, "Detecting Fake News Sources on Twitter Using Deep Neural Networks," ICET,2023



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)