ISSN: 2454-9940

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT

E-Mail :
editor.ijasem@gmail.com
editor@ijasem.org

www.ijasem.org

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT
ISSN 2454-9940
www.ijasem.org
Vol 19, Issue 2, 2025

AN ADVANCED DEEP LEARNING ENSEMBLE MODEL FOR FAKE NEWS DETECTION USING MULTI-MODAL DATA FUSION AND NATURAL LANGUAGE PROCESSING

Guddam Kousalya,[1] , K.Chowdeswaridevi[2] , B.Haseena bee [3] , M.Vaishnavi[4] , S.Harsha Vardhini[5] , G.Venkata Surekha[6]

[1]Assistant Professor , Dept of CSE, Gouthami Institute Of Technology and Management for Women, Andhra Pradesh, India

[2,3,4,5,6]U.G Students, Dept of CSE, Gouthami Institute Of Technology and Management for Women, Andhra Pradesh, India

ABSTRACT

Fake news has become a pervasive issue in the digital era, influencing public opinion, misleading readers, and contributing to the spread of misinformation. With the rise of social media and online platforms, the rapid dissemination of fake news has made it challenging to distinguish between credible and false information. This project proposes a machine learning-based fake news detection system to automatically classify news articles as real or fake.The system utilizes a combination of Natural Language Processing (NLP) techniques and machine learning algorithms such as Logistic Regression, Naive Bayes, Support Vector Machines (SVM), and Deep Learning models like LSTMs and BERT. By analyzing text patterns, sentiment, and writing style, the model can detect potential fake news with high accuracy.The proposed model is trained on a labeled dataset containing verified and fake news articles. Various preprocessing techniques like tokenization, stop-word removal, and word embeddings are applied to clean and structure the data. Performance evaluation is conducted using metrics like Accuracy, Precision, Recall, and F1-score to determine the most effective model.Furthermore, the system includes a user-friendly interface that allows users to input news articles or URLs for real-time fake news detection. This solution aims to mitigate the spread of misinformation by providing users with reliable credibility assessments. In the long term, this project contributes to the fight against disinformation, ensuring a more informed society.

INTRODUCTION

In the contemporary digital landscape, the dissemination of information has undergone a massive transformation, primarily driven by the proliferation of the internet, social media platforms, and online news outlets. While this digital revolution has democratized access to information, allowing users to stay informed about global events in real-time, it has also inadvertently given rise to a critical problem—the spread of fake news. Fake news, defined as false or misleading information presented as legitimate news, has emerged as a significant threat to public discourse,

ISSN 2454-9940
www.ijasem.org
Vol 19, Issue 2, 2025

democratic institutions, and societal trust. Whether generated for political manipulation, financial gain, or simply to mislead readers, fake news has the potential to cause widespread confusion, incite violence, and erode the credibility of traditional media.The propagation of fake news is especially rampant on social media platforms such as Facebook, Twitter, Instagram, and WhatsApp, where information is shared with little to no verification. These platforms act as echo chambers, often reinforcing biases and promoting content that aligns with users' pre-existing beliefs. Unlike traditional news sources that adhere to journalistic standards and fact-checking practices, social media lacks structured mechanisms to authenticate the veracity of the content being circulated. This ease of publication and the virality of online platforms make it increasingly difficult for users to distinguish between credible and deceptivesources.

LITERATURE REVIEW

The emergence of fake news as a major societal issue has catalyzed a wave of research efforts aimed at understanding and combating the spread of misinformation. Scholars, technologists, and policymakers have explored diverse strategies, from social network analysis to machine learning and deep learning-based text classification, to address this challenge. The literature on fake news detection spans several key domains, including natural language processing (NLP), computational linguistics, social network analysis, psychology, and information retrieval. This review synthesizes key studies and methodologies, providing a foundation for the current project's development.

Initial approaches to fake news detection relied heavily on manual fact-checking and rule-based systems. These systems depended on human fact-checkers and structured knowledge databases to verify the claims made in news articles. Although this method was precise in verifying factual accuracy, it was not scalable due to the volume and velocity of online news publication. Organizations like Snopes and PolitiFact provided accurate validations, but their manual nature limited response time. As the need for automation became evident, researchers began exploring computational methods to mimic human judgment. Early computational models primarily focused on keyword-based filtering and metadata analysis, which offered limited performance because fake news authors quickly adapted to circumvent such shallow heuristics.The next significant evolution came with the adoption of machine learning algorithms, especially supervised learning models, trained on labeled datasets of fake and real news. The study by Ahmed et al. (2017) employed Support Vector Machines (SVM) and Decision Trees using features such as word frequency, sentence length, and punctuation usage. Their results demonstrated that machine learning classifiers, when equipped with engineered features, could distinguish between fake and real articles with reasonable accuracy.

EXISTING METHODS

The problem of fake news detection has garnered significant attention over the past decade due to the increasing prevalence of misinformation across online platforms. Researchers, data scientists, and technologists have explored various methodologies and computational approaches to address this issue. Early attempts at detecting fake news primarily relied on manual fact-checking processes.

ISSN 2454-9940
www.ijasem.org
Vol 19, Issue 2, 2025

These methods involved domain experts or journalists evaluating the credibility of a news article based on external sources, prior knowledge, and logical consistency. While this approach was precise and often yielded accurate results, it was extremely time-consuming and lacked scalability. As the volume of content on the internet exponentially increased, it became impractical to verify each article manually, thus prompting the need for automated detection mechanisms. One of the earliest and most widely adopted automated techniques involved rule-based systems. These systems used a predefined set of rules and keywords to flag potentially misleading content. For instance, a rule-based fake news detector might search for sensationalist phrases such as "you won't believe," "shocking revelation," or "secret exposed." While this method could catch obvious clickbait, it failed to generalize across diverse writing styles and topics. Furthermore, fake news authors adapted quickly by modifying their language to avoid detection, rendering rule-based approaches insufficient in the long run. These limitations spurred the transition from heuristic methods to more dynamic approaches such as machine learning, which could learn from data and improve over time.With the advent of machine learning, supervised learning models became a popular choice for fake news classification. These models are trained on labeled datasets containing examples of both real and fake news. Among the earliest algorithms used were Logistic Regression, Decision Trees, Logistic Regression, a statistical model, was favored for its simplicity Naive Bayes classifiers, and Support Vector Machines (SVM). Logistic Regression, a statistical model, was favored for its simplicity and interpretability. model, was favored for its simplicity and interpretability. It estimates the probability of a news article being fake based on input features such as word frequency,headline structure, or article length. Naive Bayes, on the other hand, is a probabilistic model based on Bayes' theorem and assumes feature independence. Despite this naive assumption, it often performs well in text classification tasks due to the sparsity and high dimensionality of textual data.

PROPOSED METHOD

The proposed method for fake news detection in this project is a comprehensive, multi-phase system that integrates advanced Natural Language Processing (NLP) techniques with machine learning and deep learning algorithms. The main goal is to develop an automated and highly accurate classification model capable of identifying fake news from legitimate articles. Unlike earlier methods that relied on basic keyword matching or heuristic rules, this system leverages linguistic features, contextual embeddings, and semantic analysis to detect subtle patterns indicative of misinformation.The methodology begins with data collection, which serves as the foundation of the entire project. Kaggle, online news portals, and fact-checking websites like PolitiFact and Snopes. This dataset must and representative of multiple news domains, including politics, health, finance, and entertainment. The dataset includes fields such as headline, body text, source URL, and publication date. Maintaining diversity in the dataset is crucial to ensure the model generalizes well across different types of news content and avoids overfitting to a specific genre.Once the data is collected, it undergoes an extensive data preprocessing phase. This step ensures that the raw textual data is transformed into a format suitable for

ISSN 2454-9940
www.ijasem.org
Vol 19, Issue 2, 2025

machine learning models.

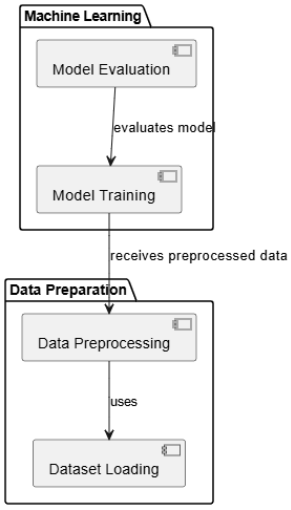
SOFTWARE REQUIREMENTS:

• **Operating System :** Windows 8 and above
• **Coding Language** : Python 3.12.0
• **Framework** : Django
• **Platform** : Visual Studio Code (Preferable)

HARDWARE REQUIREMENTS:

• **System** : MINIMUM i3 and above
• **Hard Disk** : 40 GB. (min)
• **RAM**: 4 GB. (min)

ARCHITECTURE



SOFTWARE ENVIRONMENT

What is Python:

Below are some facts about Python.

• Python is currently the most widely used multi-purpose, high-level programming language.

• Python allows programming in Object-Oriented and Procedural paradigms. Python programs generally are smaller than other programming languages like Java.

• Programmers have to type relatively less and indentation requirement of the language, makes them readable all the time.

• Python language is being used by almost all tech-giant companies like – Google, Amazon, Facebook, Instagram, Dropbox, Uber… etc.

Advantages of Python:

Let's see how Python dominates over other languages.

1. **Extensive Libraries**

Python downloads with an extensive library and it contain code for various purposes like regular expressions, documentation generation, unit-testing, web browsers, threading, databases, CGI, email, image manipulation, and more. So, we don't have to write the complete code for that manually.

Extensible

As we have seen earlier, Python can be extended to other languages. You can write some of your code in languages like C++ or C. This comes in handy, especially in projects.

Embeddable

Complimentary to extensibility, Python is embeddable as well. You can put your Python code in your source code of a different language, like C++. This lets us add scripting capabilities to our code in the other language.

Speed Limitations

We have seen that Python code is executed line by line. But since Python is interpreted, it often results in **slow execution**. This,

ISSN 2454-9940
www.ijasem.org
Vol 19, Issue 2, 2025

however, isn't a problem unless speed is a focal point for the project. In other words, unless high speed is a requirement, the benefits offered by Python are enough to distract us from its speed limitations.

Weak in Mobile Computing and Browsers

While it serves as an excellent server-side language, Python is much rarely seen on the **client-side**. Besides that, it is rarely ever used to implement smartphone-based applications. One such application is called **Carbonnelle**.

The reason it is not so famous despite the existence of Bryton is that it isn't that secure. Design Restrictions

As you know, Python is **dynamically-typed**. This means that you don't need to declare the type of variable while writing the code. It uses **duck-typing**. But wait, what's that? Well, it just means that if it looks like a duck, it must be a duck. While this is easy on the programmers during coding, it can **raise run-time errors.**

Underdeveloped Database Access Layers

Compared to more widely used technologies like **JDBC (Java database Connectivity)** and **ODBC (Open Database Connectivity)**, Python's database access layers are a bit underdeveloped. Consequently, it is less often applied in huge enterprises.

Simple

No, we're not kidding. Python's simplicity can indeed be a problem. Take my example. I don't do Java, I'm more of a Python person. To me, its syntax is so simple that the verbosity of Java code seems unnecessary.

This was all about the Advantages and Disadvantages of Python Programming Language.

.History of Python:

What do the alphabet and the programming language Python have in common? Right, both start with ABC. If we are talking about

ABC in the Python context, it's clear that the programming language ABC is meant. ABC is a general-purpose programming language and programming environment, which had been developed in the Netherlands, Amsterdam, at the CWI (Centrum Wiskunde &Informatica). The greatest achievement of ABC was to influence the design of Python. Python was conceptualized in the late 1980s. Guido van Rossum worked that time in a project at the CWI, called Amoeba, a distributed operating system. In an interview with Bill Venners[1], Guido van Rossum said: "In the early 1980s, I worked as an implementer on a team building a language called ABC at Centrum voor Wiskunde end Informatica (CWI). I don't know how well people know ABC's influence on Python. I try to mention ABC's influence because I'm indebted to everything I learned during that project and to the people who worked on it."Later on in the same Interview, Guido van Rossum continued: "I remembered all my experience and some of my frustration with ABC. I decided to try to design a simple scripting language that possessed some of

ISSN 2454-9940
www.ijasem.org
Vol 19, Issue 2, 2025

ABC's better properties, but without its problems. So I started typing. I created a simple virtual machine, a simple parser, and a simple runtime. I made my own version of the various ABC parts that I liked. I created a basic syntax, used indentation for

-end blocks, and developed a small number of powerful data types: a hash table (or dictionary, as we call it), a list, strings, and numbers."

What is Flask?

Flask is a lightweight and flexible Python web framework known for its simplicity and ease of use. It enables developers to create robust web applications with minimal setup and a clean design. Flask adopts a modular approach and provides the essential tools needed to build web apps, leaving the choice of additional libraries to the developers. This review explores Flask's core principles, functionalities, and advantages, making it an ideal starting point for developers who prefer a minimalist framework.

A Glimpse into Flask's Philosophy

Flask was developed by Armin Ronacher as part of the Pocoo project in 2010. Its minimalist approach is guided by the philosophy of being "micro but extensible." Flask avoids imposing unnecessary restrictions, giving developers full control over application design. This flexibility has made Flask a popular choice for projects ranging from small prototypes to large- scale production applications.

Flask operates on the "microframework" principle, meaning it provides only the core features necessary for web development, such as routing, request handling, and template rendering. For additional functionalities like database integration or user authentication, developers can incorporate third-party extensions, keeping the application lightweight and customizable.

One of Flask's most notable features is its "unopinionated" nature, allowing developers to structure applications as they see fit. Unlike frameworks with strict architectural patterns, Flask offers the freedom to implement custom architectures based on project requirements.

Key Features of Flask

Flask's functionality can be summarized as follows:

1. **Routing and URL Mapping**: Flask simplifies URL routing with the @app.route decorator. Developers can map URLs to specific view functions effortlessly, creating intuitive and organized endpoints.
2. **Request and Response Handling**: Flask provides tools for handling HTTP methods (GET, POST, PUT, etc.) and working with request data, cookies, and

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT
ISSN 2454-9940
www.ijasem.org
Vol 19, Issue 2, 2025

SYSTEM DESIGN:

FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are,

- **ECONOMICAL FEASIBILITY**
- **TECHNICAL FEASIBILITY**
- **SOCIAL FEASIBILITY**

ECONOMICAL FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

UML DIAGRAMS:

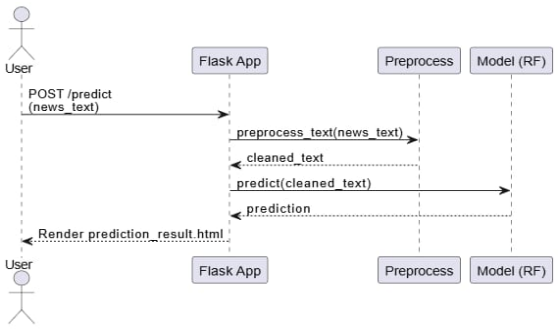


Fig:sequence diagram

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT
ISSN 2454-9940
www.ijasem.org
Vol 19, Issue 2, 2025

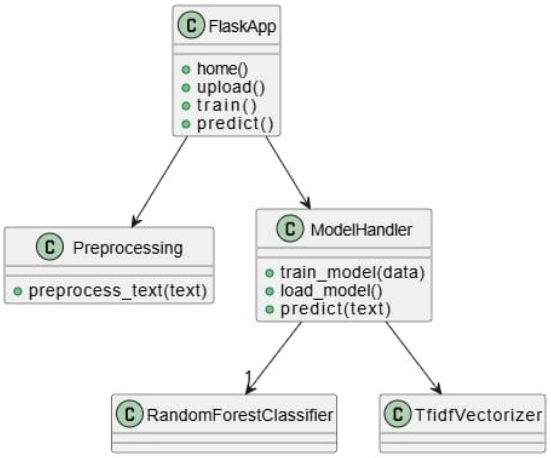


Fig:class diagram

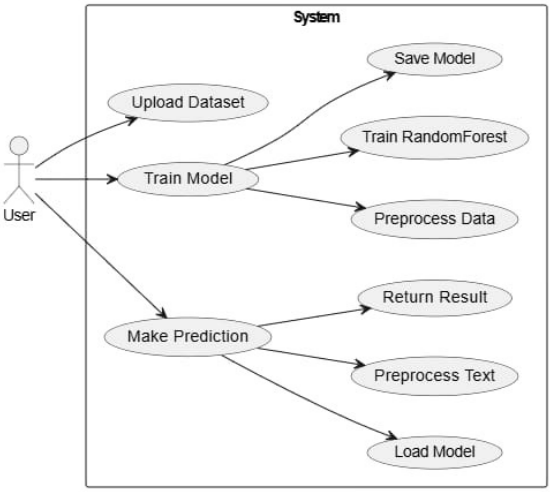


Fig:Use case diagram

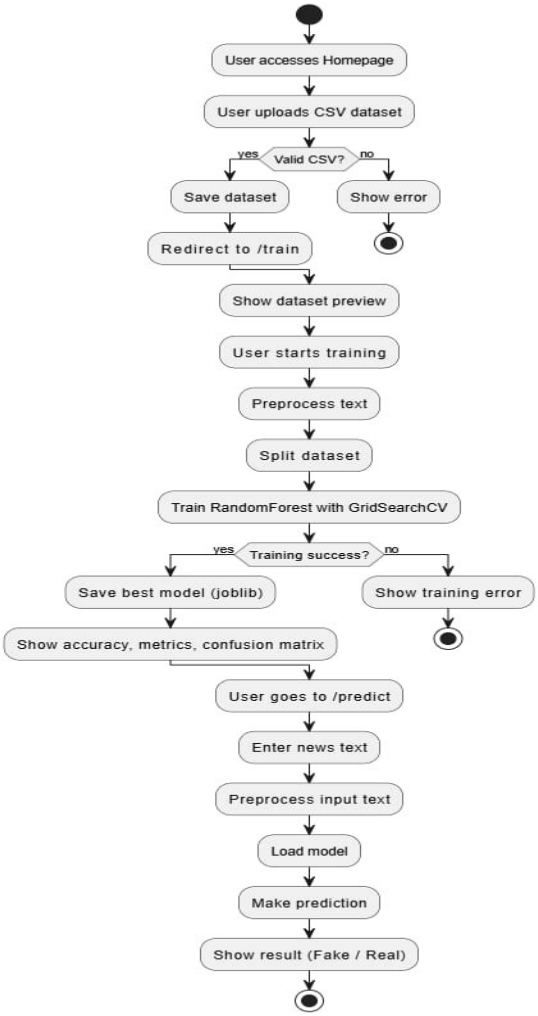


Fig:Behaviour diagram

SYSTEM TEST:

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.

RESULT

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT
ISSN 2454-9940
www.ijasem.org
Vol 19, Issue 2, 2025

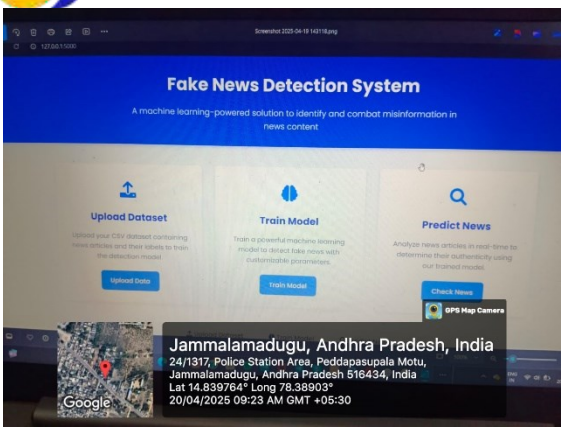


fig: Interface of fake news detection

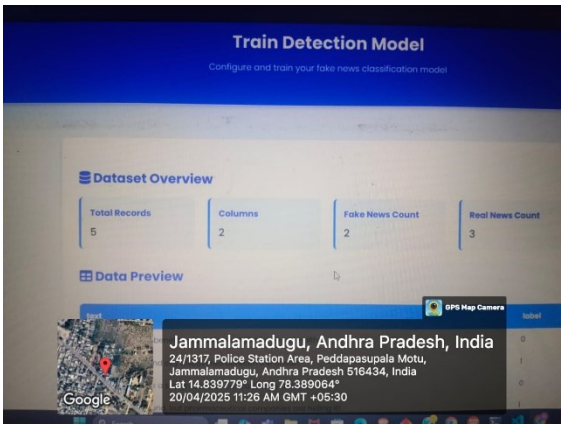


fig: Interface of fake news detection

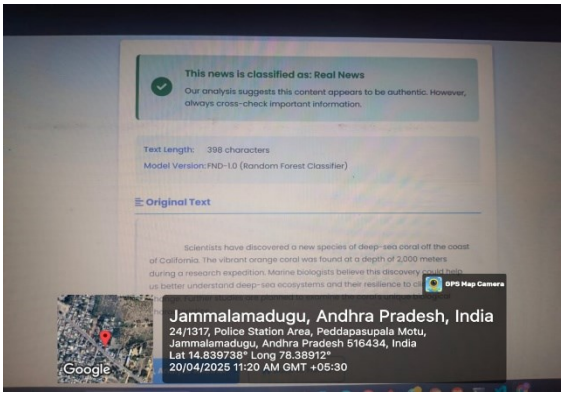


Fig:input to fake news detection

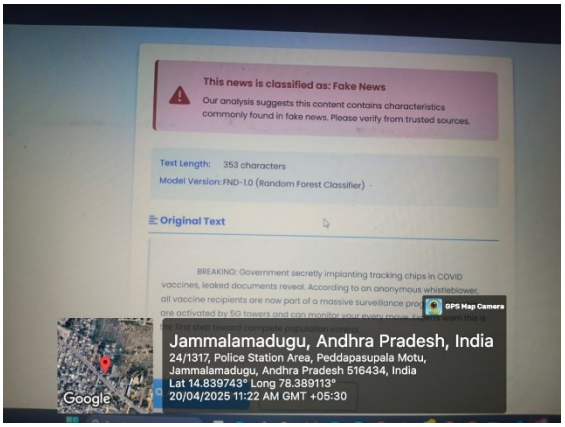


Fig:output to fake news detection

CONCLUSION

The rise of fake news in the digital age has emerged as a profound challenge for societies across the globe. It undermines democratic processes, distorts public opinion, incites social unrest, and erodes trust in reliable news sources. This project has thoroughly examined and addressed the problem of fake news by designing a robust machine learning-based detection system capable of distinguishing between legitimate and deceptive information. The system combines various Natural Language Processing (NLP) techniques, machine learning algorithms, and deep learning models to effectively analyze and classify news content. Through the use of supervised learning techniques such as Logistic Regression, Naive Bayes, Support Vector Machines (SVM), and advanced models like Long Short-Term Memory (LSTM) networks and BERT, this research achieves a reliable and scalable solution to combat.The implementation of this fake news detection system was rooted in a meticulous process involving data collection, preprocessing, model training, evaluation, and deployment. The system was trained on benchmark datasets containing both real and fake news articles,

ISSN 2454-9940
www.ijasem.org
Vol 19, Issue 2, 2025

enabling it to learn patterns and characteristics that differentiate factual content from manipulated narratives. tokenization, lemmatization, stop-word embeddings were employed to convert raw textual data into meaningful input features for the classifiers. These steps were instrumental in improving the performance and accuracy of the models, allowing them to effectively capture both syntactic and semantic nuances in text.

REFERENCES

Sharma, K., Qian, F., Jiang, H., Ruchansky, N., Zhang, M., & Liu, Y. (2019). "Combating Fake News: A Survey on Identification and Mitigation Techniques." ACM Computing Surveys (CSUR), 52(5), 1-36.

Ahmed, H., Traore, I., & Saad, S. (2018). "Detecting Opinion Spams and Fake News Using Multidimensional Analysis." IEEE Access, 7, 16892-16911.

Zhou, X., & Zafarani, R. (2020). "A Survey of Fake News Detection: Methods, Data, and Opportunities." ACM Computing Surveys (CSUR), 53(5), 1-40.

Wang, W. Y. (2017). "'Liar, Liar Pants on Fire': A New Benchmark Dataset for Fake News Detection." Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, 422-426.

Shu, K., Sliva, A., Wang, S., Tang, J., & Liu, H. (2017). "Fake News Detection on Social Media: A Data Mining Perspective." ACM SIGKDD Explorations Newsletter, 19(1), 22-36.

Conference Papers

Ruchansky, N., Seo, S., & Liu, Y. (2017). "CSI: A Hybrid Deep Model for Fake News Detection." Proceedings of the 2017 ACM Conference on Information and Knowledge Management (CIKM), 797-806.

Karimi, H., Roy, P., Saba-Sadiya, S., & Tang, J. (2018). "Multi-Source Multi-Class Fake News Detection." Proceedings of the 27th International Conference on Computational Linguistics (COLING), 1547-1557.

Zhang, H., & Ghorbani, A. A. (2020). "An Overview of Online Fake News Detection: Methods, Open Problems, and Challenges." Proceedings of the International Conference on Big Data and Smart Computing (BigComp), 36-42.