COMSATS University Islamabad

Abbottabad, Pakistan

AI-Based Fake News Detection App for Social Media Platform

By

Bilal Hamid	**CIIT/SP20-BSE-018/ATD**
Syed Taimur Hassan	**CIIT/SP20-BSE-026/ATD**
Muhammad Abbas Khan	**CIIT/SP20-BSE-031/ATD**

Supervisor

Sir Ahsan Khan

Bachelor of Science in Computer Science (2020-2024)

The candidate confirms that the work submitted is their own and appropriate credit has been given where reference has been made to the work of others.

COMSATS University, Islamabad, Pakistan

AI-Based Fake News Detection App for Social Media Platform

A project presented to

COMSATS Institute of Information Technology, Islamabad

In partial fulfillment

of the requirement for the degree of

Bachelor of Science in Software Engineering (2020-2024)

By

Bilal Hamid	**CIIT/SP20-BSE-018/ATD**
Syed Taimur Hassan	**CIIT/SP20-BSE-026/ATD**
Muhammad Abbas Khan	**CIIT/SP20-BSE-031/ATD**

DECLARATION

We hereby declare that this software, neither whole nor as a part has been copied out from any source. It is further declared that we have developed this software and accompanied report entirely on the basis of our personal efforts. If any part of this project is proved to be copied out from any source or found to be reproduction of some other. We will stand by the consequences. No Portion of the work presented has been submitted of any application for any other degree or qualification of this or any other university or institute of learning.

Syed Taimur Hassan	Muhammad Abbas Khan	Bilal Hamid
_______________	_________________________	________________

CERTIFICATE OF APPROVAL

It is to authenticate that the final year project of BS (SE) "AI-based Fake News Detection System" was proposed by **Muhammad Abbas khan (CIIT/SP20-BSE-031/ATD)**, and **Taimur Hassan (CIIT/SP20-BSE-026/ATD) and Bilal Hamid(CIIT/SP20-BSE-018/ATD)** under the supervision of "**Sir Ahsan Khan**" and that in her opinion; it is completely adequate in extent and virtue forth Bachelor of Science in Software Engineering degree.

Sir Ahsan Khan

Supervisor

External Examiner

Mazhar Ali

Head of Department

(Department of Computer Science)

EXECUTIVE SUMMARY

The motivation for this project stems from the rising prevalence of fake news in today's digital age. The spread of misinformation can have serious social, political, and economic consequences. By providing users with a tool that can help them verify the authenticity of news, we aim to promote informed decision-making and curb the spread of misinformation.

Our project is a comprehensive solution for real-time fake news detection. It operates as a browser extension that users can easily use on platforms such as Facebook and Twitter. By leveraging state-of- the-art Natural Language Processing (NLP) and Machine Learning techniques, it provides users with arobust and reliable tool to distinguish between true and false news.

The functionality of this system revolves around three primary techniques: Sentiment Analysis,Linguistic Analysis, and Knowledge-Based Analysis.

The extracted features from these techniques then feed into a Random Forest model, which makes the prediction about the veracity of the news.

The design of the project is focused on maintaining a high degree of modularity and scalability. We adhere to object-oriented programming principles, creating reusable, independent classes responsible forspecific project functions. This approach allows for easy maintenance and upgrades, and it facilitates theaddition of more features in the future.

ACKNOWLEDGEMENT

All praise is to Almighty Allah who bestowed upon us a minute portion of His boundless knowledge by virtue of which we were able to accomplish this challenging task.

We are greatly indebted to our project supervisor "**Sir Ahsan Khan**". Without his personal supervision, advice and valuable guidance, completion of this project would have been doubtful. We are deeply indebted to him for his encouragement and continual help during this work.

We are also grateful to our parents and family, who have always been a source of inspiration and instilled in us the principles of honesty and hard work.

Syed Taimur Hassan   Muhammad Abbas Khan   Bilal Hamid

______________   ______________________   _______________

ABBREVIATIONS

SQE	Software Quality Engineering
SDLC	Software development Life Cycle
HCI	Human Computer Interaction
UI	User Interface
SPM	Software Project Management
ML	Machine Learning
SRE	Software Requirement Engineering
WT	Web Technologies
SDA	Software Design and Architecture
DFD	Data Flow Diagram

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. Introduction:

The Fake News Detection system represents a pivotal advancement in the ongoing battle against the dissemination of misinformation within the contemporary digital landscape. In an era where the internet is rife with deceptive narratives, this comprehensive solution stands as a formidable defense mechanism. Operating seamlessly as a browser extension, it seamlessly integrates into popular social media platforms such as Facebook and Twitter.

Incorporating cutting-edge technologies like Natural Language Processing (NLP) and Machine Learning, the system stands as a beacon of reliability for users seeking to discern the authenticity of news articles. The triumvirate of techniques employed—Sentiment Analysis, Linguistic Analysis, and Knowledge-Based Analysis—work synergistically to fortify the system's efficacy. Sentiment Analysis delves into the emotional resonance encapsulated within news content, Linguistic Analysis employs sophisticated NLP techniques to dissect textual structures and patterns, while Knowledge-Based Analysis harnesses the prowess of a pre-trained BERT model to achieve a nuanced understanding of semantics.

1.1.Brief

The Fake News Detection system, a ground-breaking addition to the technological landscape, emerges as a response to the escalating menace of false or misleading information, especially on social media platforms. In an environment where the lines between truth and falsehood blur, this innovative system takes centre stage. Designed from the ground up, it stands as a testament to our commitment to fostering a digital space characterized by accuracy and credibility. This proposed system, a paragon of cutting-edge technology, seeks to harness the power of advanced Natural Language Processing (NLP) and Machine Learning (ML) techniques. Its mission is clear: to empower users with an accessible and user-friendly tool capable of unravelling the authenticity of news content in a sea of information overload. As we delve deeper into the intricate web of misinformation, the Fake News Detection system emerges as a beacon of clarity, providing users with the means to navigate through the complexities of the digital information landscape.

1.2. Relevance to Course Modules

Table 1. 1: Relevance to Course Module

Serial	Course Name	Course Code	Course Relevance
1.	Programming Fundamentals	PF	Basic Programming skills
2.	Introduction to Software Engineering	SE	Documentation and architectural designs

3.	Human Computer Interaction	HCI	Designing the application according to design principles
4.	Artificial Intelligence	AI	For performing training and deploying model
5.	Software Testing	ST	For Testing application and creating test case.
6.	Web	WEB	Creating extension and server

1.3.Project Background:

The Fake News Detection system is a browser extension designed to combat the proliferation of misinformation in today's digital landscape, particularly on platforms like Facebook and Twitter. Leveraging cutting-edge Natural Language Processing (NLP) and Machine Learning techniques, the system provides users with a reliable tool to distinguish between true and falsenews.

This project revolves around three core techniques:

Sentiment Analysis: This assesses the emotional tone of news content, using sentiment as a cuefor authenticity. Beyond mere semantic analysis, this technique probes the emotional resonance embedded within news content, offering a nuanced understanding that contributes to the overall determination of authenticity.

Linguistic Analysis: Employing NLP techniques like Part of Speech (POS) tagging and NamedEntity Recognition (NER), the system examines the text's structure, uncovering linguistic patterns common in deceptive news stories. A sophisticated array of NLP techniques, including Part of Speech (POS) tagging and Named Entity Recognition (NER), meticulously scrutinizes the structural nuances of text. This level of linguistic scrutiny unveils patterns inherent in deceptive news stories, enriching the system's discernment capabilities.

Knowledge-Based Analysis: A pre-trained BERT model is used to understand the contextual relationships between words in the text, offering deeper insights into semantic content. Leveraging the prowess of a pre-trained BERT model, the system dives deep into contextual relationships between words. This approach not only enhances semantic comprehension but also adds a layer of sophistication to the discernment process.

These techniques feed into a Random Forest model, which provides the final verdict on news veracity.

The project's design prioritizes modularity and scalability, adhering to object-oriented programming principles. This approach ensures easy maintenance, upgrades, and the potentialfor future feature additions. Ultimately, the Fake News Detection system aims to empower userswith a practical solution for making informed decisions and combating the spread of misinformation in the digital age. Converging seamlessly, these techniques funnel into a robust Random Forest model, the arbiter that issues the final verdict on the veracity of news. The project's design ethos, underpinned by modularity and scalability, adheres to object-oriented programming principles. This meticulous approach ensures not only the ease of system maintenance and upgrades but also opens avenues for future feature enhancements. In essence, the Fake News Detection system aspires to be a dynamic, evolving solution, committed to empowering users in their quest for information accuracy amid the pervasive misinformation in the digital age.

1.4. Literature Review

The literature review highlights the evolution of fake news detection methodologies, emphasizing the transition from manual fact-checking and rule-based systems to the integration of advanced Natural Language Processing (NLP) and Machine Learning (ML) techniques. This shift has significantly enhanced the scalability and accuracy of fake news detection. Feature engineering, acritical aspect of these approaches, has evolved to include linguistic cues, social network analysis, and user behavior patterns, with recent advancements in word embeddings and transformer-basedmodels like BERT improving the precision of feature extraction.

Social network analysis and the study of user behavior patterns have emerged as pivotal dimensions in the ongoing quest for more effective detection systems. Understanding the dynamics of information dissemination within social networks, as well as discerning patterns in user interactions with content, contributes to a holistic approach. Integrating these social dimensions into detection algorithms enhances the contextual understanding required to differentiate between authentic and deceptive information, reflecting the interconnected nature of the digital information ecosystem.

Recent advancements in word embeddings and transformer-based models like BERT have not only revolutionized feature extraction but have also paved the way for more nuanced contextual understanding. The ability to capture intricate relationships between words and their contextual meanings contributes significantly to the accuracy of distinguishing between genuine and fake news.

While machine learning models, ranging from traditional Random Forests to sophisticated deep learning architectures like CNNs and RNNs, showcase their prowess in pattern recognition, the literature recognizes the persistent challenges. Adversarial attacks, where malicious actors deliberately manipulate content to deceive detection models, pose an ongoing threat. The dynamic evolution of disinformation tactics further complicates the landscape, requiring adaptive and resilient detection mechanisms.

Machine learning models, encompassing both traditional methods such as Random Forests and deep learning architectures like Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), have demonstrated their ability to capture intricate patterns in textual data, thereby automating the process of fake news identification. The availability of benchmark datasets like LIAR-PLUS and the Fake News Challenge dataset has facilitated rigorous evaluation and benchmarking of these detection algorithms, fostering healthy competition and innovation in the field.

However, the literature also underscores the enduring challenges in fake news detection, includingvulnerabilities to adversarial attacks, the evolution of disinformation tactics, and the issue of class imbalance in datasets. These challenges necessitate ongoing research efforts, including the exploration of multimodal approaches that combine textual and visual cues, as well as addressing ethical considerations related to content moderation. In summary, while significant progress has been made, the dynamic nature of fake news dissemination calls for continued interdisciplinary collaboration and innovative approaches to effectively combat its spread.

1.5.Methodology and Software Lifecycle for This Project

Data and the methods that operate on it are bundled together into objects. This encapsulation makes it easier to manage complexity, as each object's behaviour is defined by its class. In your project, each class has a specific purpose, like the Data Handler for handling data, Feature Extractor for feature extraction, or Classifier for classification. Each object forms a separate entity whose internal workings are decoupled from other parts of the system. This makes it easier to make changes to one part of the system without affecting others. For instance, if you wanted to change the way you pre-process data, you would only need to modify the Pre-processor class. In the context of this project, we started with the essential features (loading data, pre-processing, features extraction, etc.) and then iteratively improved upon the model by adding advanced features like BERT combining it with a Random Forest model. This iterative approach allows you to evaluate the system at each step, make necessary modifications, and adapt to changes.

The motivation for this project stems from the rising prevalence of fake news in today's digital age. The spread of misinformation can have serious social, political, and economic consequences. By providing users with a tool that can help them verify the authenticity of news, we aim to promote informed decision-making and curb the spread of misinformation.

Our project is a comprehensive solution for real-time fake news detection. It operates as a browser extension that users can easily use on platforms such as Facebook and Twitter. By leveraging state-of-the-art Natural Language Processing (NLP) and Machine Learning techniques, it provides users with a robust and reliable tool to distinguish between true and false news.

The functionality of this system revolves around three primary techniques: Sentiment Analysis, Linguistic Analysis, and Knowledge-Based Analysis.

- Sentiment Analysis: It gauges the emotional resonance of the news content. It's based on the understanding that the sentiment, tone, or emotional charge carried in a text provides valuable clues about its authenticity.
- linguistic Analysis: By employing NLP techniques such as Part of Speech (POS) tagging and Named Entity Recognition (NER), the system carries out a detailed linguistic examination of thetext. It helps in identifying the grammatical components and vital entities in the text, thereby uncovering patterns and structures common in deceptive news stories.
- Knowledge-Based Analysis: The system utilizes a pre-trained BERT (Bidirectional Encoder Representations from Transformers) model. BERT helps in understanding the contextual relationships between words in the text, offering deeper insights into the semantic content of the text.

The extracted features from these techniques then feed into a Random Forest model, which makes the final prediction about the veracity of the news.

The design of the project is focused on maintaining a high degree of modularity and scalability. We adhere to object-oriented programming principles, creating reusable, independent classes responsible for specific project functions. This approach allows for easy maintenance and upgrades, and it facilitates the addition of more features in the future.

The software lifecycle of the Fake News Detection project can be delineated into distinct phases, each contributing to the evolution and refinement of the system.

1.5.1. Inception and Motivation:

The project is conceived in response to the escalating prevalence of fake news in the digital age.

Motivation centres around addressing the potential social, political, and economic ramifications of misinformation.

1.5.2. Design and Architecture:

The design emphasizes high modularity and scalability, aligning with object-oriented programming principles.

Reusable classes, such as Data Handler, Feature Extractor, and Classifier, are crafted to encapsulate specific functionalities.

1.5.3. Iterative Development:

Iterations commence with foundational features, progressively incorporating advanced techniques like BERT for semantic understanding.

The iterative nature allows for continuous evaluation, modification, and enhancement, ensuring the system's adaptability.

1.5.4. Functionality and Techniques:

The Fake News Detection system operates as a browser extension on platforms like Facebook and Twitter.

Advanced Natural Language Processing (NLP) and Machine Learning techniques, including Sentiment Analysis, Linguistic Analysis, and Knowledge-Based Analysis, form the foundation of the system.

1.5.5. Feature Extraction and Classification:

Extracted features from NLP techniques feed into a Random Forest model, culminating in a final prediction regarding the authenticity of news content.

The integration of diverse techniques ensures a comprehensive understanding of news articles.

1.5.6. Maintainability and Future Enhancements:

Object-oriented design principles contribute to the system's maintainability and ease of upgrades.

The architecture allows for the seamless addition of new features, ensuring the system's adaptability to emerging challenges.

In summary, the methodology embraces OOP principles and an iterative development approach, while the software lifecycle encompasses inception, design, iterative development, incorporation of advanced techniques, and a focus on maintainability and future enhancements. This comprehensive approach positions the Fake News Detection system as a dynamic and adaptable solution in the ongoing battle against misinformation.

2. Problem Definition

2.1. Problem Statement

In the era of digital communication and information sharing, the proliferation of fake news hasemerged as a significant societal concern. The problem at hand revolves around the need to develop an effective and reliable Fake News Detection system to combat the widespread dissemination of misinformation on social media platforms and news websites. The rise of deceptive news articles, often designed to manipulate public opinion, jeopardizes the credibility of information sources, and undermines the foundations of informed decision-making. Therefore, the problem definition encompasses the creation of a technological solution capable of discerning authentic news from fake news, thereby safeguarding the integrity of informationin the digital age.

The core problem lies in the urgent need to create a technological solution capable of effectively discerning authentic news from fake news. This imperative task aims to fortify the integrity of information in the digital age, where the consequences of misleading narratives can extend to societal, political, and economic realms. The evolving nature of misinformation tactics requires a dynamic system that can adapt to the ever-changing strategies employed by purveyors of fake news.

In essence, the problem definition encapsulates the development of a Fake News Detection system that serves as a digital sentinel, safeguarding the veracity of information circulating through online channels. This solution becomes paramount in preserving the trustworthiness of news sources, thereby fostering an environment where individuals can make informed decisions based on accurate and reliable information. As the digital age continues to evolve, addressing this challenge is pivotal for maintaining the integrity of public discourse and ensuring a well-informed and resilient society.

2.2. Deliverables and Development Requirements

2.2.1. Deliverables:

- **Mockups:**
  Visual representations of the system's interface, providing stakeholders with a preview of the design and layout.

- **Final UI Design:**
  A polished and user-friendly graphical interface, incorporating feedback from stakeholders and designed to enhance the user experience.

- **Extension:**
  The fully functional browser extension implementing the Fake News Detection system, seamlessly integrating with platforms like Facebook and Twitter.

- **Test Cases and Test Plans:**
  Comprehensive test cases and plans to ensure the system's reliability, including unit tests, integration tests, and end-to-end testing.

- **User and Technical Documents:**
  User documentation providing instructions on how to use the Fake News Detection system. Technical documentation detailing the system architecture, data flow, and functionalities for developers and maintainers.

- **Deployed and Live System:**
  A deployed and live version of the Fake News Detection system accessible to users. This involves hosting the extension and associated services to ensure real-time functionality.

2.2.2. Development Requirements:

- **UI/UX Designer Skills:**
  Proficiency in designing intuitive and aesthetically pleasing user interfaces. Ability to create a seamless user experience, considering user interactions and feedback.

- **Proficiency in ML:**
  Strong understanding and expertise in Machine Learning concepts, particularly related to fake news detection.
  Knowledge of algorithms, models, and techniques for analysing and classifying textual data.

- **Back-End Development:**
  Development skills for the server-side components of the system, including data handling, feature extraction, and model implementation.

- **Front-End Integration:**
  Ability to integrate the developed back-end functionalities into a responsive and interactive front-end interface.
- **TensorFlow:**
  Proficiency in using TensorFlow, a popular open-source machine learning library, for implementing and training machine learning models.
- **QA Testing:**
  Quality Assurance skills to create robust test cases, execute testing procedures, and ensure the system meets specified requirements.
- **Collaboration Skills:**
  Effective communication and collaboration skills to work seamlessly with UI/UX designers, back-end developers, and other team members.

3. Requirement Analysis

3.1. Use Cases Diagram(s)

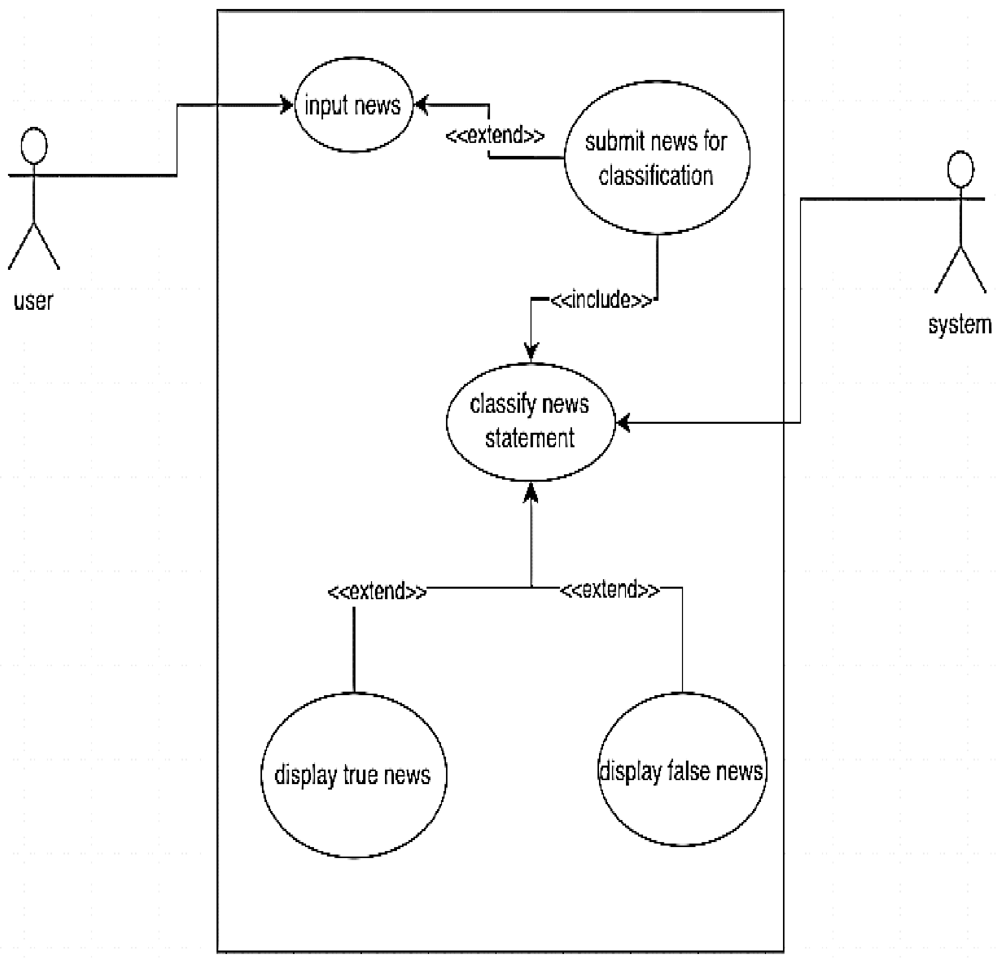

Fig 3. 1: Use Case Diagram

3.2.Detailed Use Cases:

3.2.1. Input News Use case:

Table 3. 1: UC-1 Input News

Use Case ID:	UC-1
Use Case Name:	Input News
Actors:	Actor: User
Description:	The user will copy or type the news in the extension.
Trigger:	The click on the field and input text
Preconditions:	PRE-1. The extension must be installed. PRE-2. User has access to News.
Postconditions:	POST-1. The news is sent to sever
Normal flow	User would copy to type news in the field and press the submit button the news will be sent to server.
Alternative flow of events	AF-1. Extension doesn’t load. 1- Retry or check connection and send again.

3.2.2. Submit news for Classification:

Table 3. 2: UC-2 Submit News for Classification

Use Case ID:	UC-2
Use Case Name:	Submit news for classification
Actors:	User
Description:	Users send the request to the server.
Trigger:	User presses the login button.
Preconditions:	PRE-1. News should be entered into the field. PRE-2. Extension should be loaded
Postconditions:	POST-1. Response comes back form server.
Normal flow	After writing the news in the field user would press the button
Alternative flow of event	AF-1. Field is empty. 1- Write the news and then press enter.

3.2.3. Classified News Statement:

Table 3. 3: UC-3 Classified News Statement

Use Case ID:	UC-3
Use Case Name:	Classified news statement
Actors:	System
Description:	System does the classification.
Trigger:	Upon receiving the news from the user, system does the classification.
Preconditions:	PRE-1. User had sent the news to the server
Postconditions:	POST-1. Classification is done and response sent back.
Normal flow	Pre-processing of the news occur, and features are extracted and entered classifier for classification.
Alternative flow of events	AF-1. Server is down. 1- Restart the server. AF-2. Received no news. 1- Wait till someone sends the news

3.2.4. Display result:

Table 3. 4: UC-4 Display Result

Use Case ID:	UC-4
Use Case Name:	Display result
Actors:	System
Description:	Result is shown into extension whether the given news is fake or true.
Trigger:	System is done with Classification and response is sent to extension.
Preconditions:	PRE-1. System has received the request form the user. PRE-2. The system is up and running
Postconditions:	POST-1 User is displayed the prediction.
Normal flow	After the system is done with classification the response is sent to extension
Alternative flow of events	AF-1. If the prediction was true 1- Legitimate news with probability will be shown. AF-2. If the prediction was false 1- False news with probability will be shown

3.3.Functional Requirements

FR-1. The system shall allow users to input a news statement in a text input field.
FR-2. The system shall validate that the news statement is not empty.
FR-3. The system shall provide a 'Submit' button for users to submit their newsstatement for classification.
FR-4. The system shall display an error message when the submission fails.
FR-5. Upon successful submission, the system shall classify the submitted news statement as true or false.
FR-6. The system shall use trained ML models for the classification of the news statements.
FR-7. The system shall handle any errors during the classification process and notifythe user.
FR-8. The system shall display the result of the classification to the user.
FR-9. The system shall show the confidence level of the classification result.
FR-10. The system shall provide an option to classify another news statement after the results are displayed.

3.4.Non-Functional Requirements

3.4.1. Usability

USE-1: The system shall provide a user-friendly interface, allowing users to easily input a newsstatement for classification.

USE-2: The system shall provide a clear indication of the classification result, along with aconfidence score for better understanding.

USE-3: The system shall guide users in case of any errors or issues, offering solutions tocommon problems, such as failed requests or invalid inputs.

3.4.2. Performance

PER-1: 95% of news classification requests should be processed and results displayed within 10 seconds from the time the user submits the request.

PER-2: The system should be capable of handling at least 1000 concurrent requests withoutdegradation in performance or response time.

3.4.3. Reliability

REL-1: The system should be available 90% of the time, with accuracy of classification above80%.

REL-2: The system must be able to recover from failures quickly and easily.

4. Design and Architecture

4.1. Class diagram:

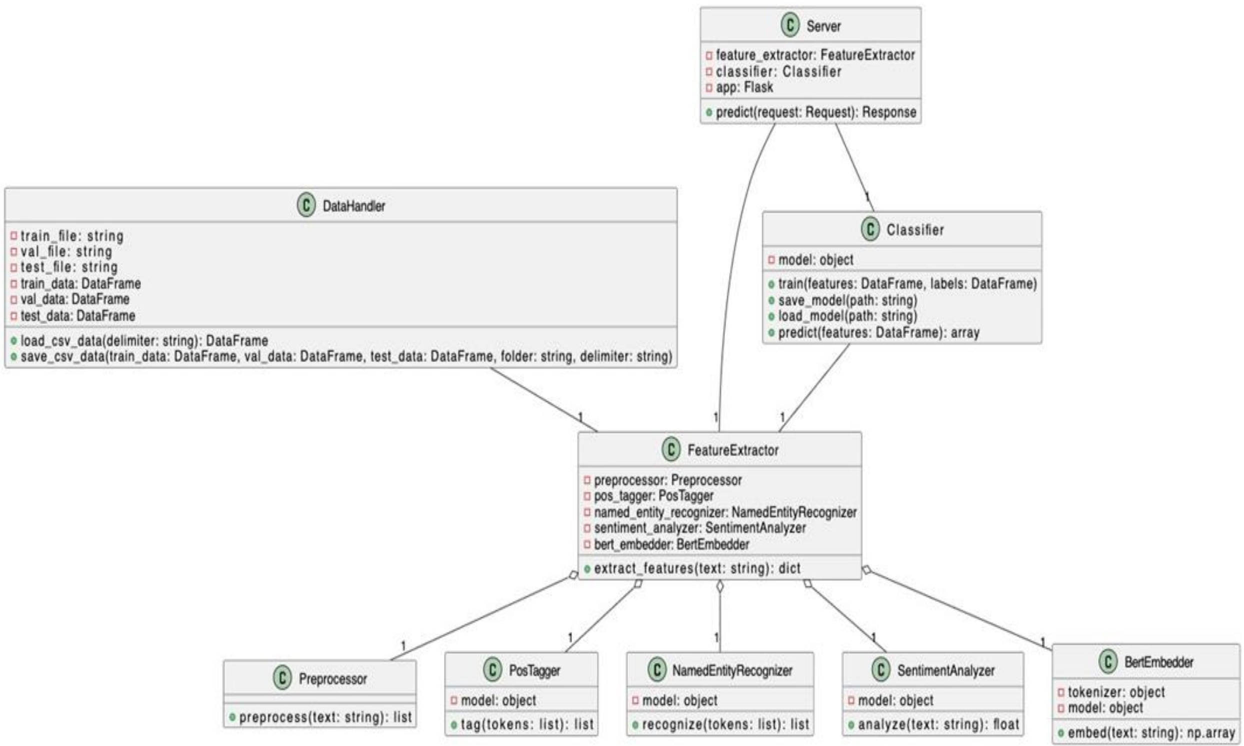

Fig 4. 1: Class Diagram

4.2. Data Representation

4.2.1. Sequence Diagram

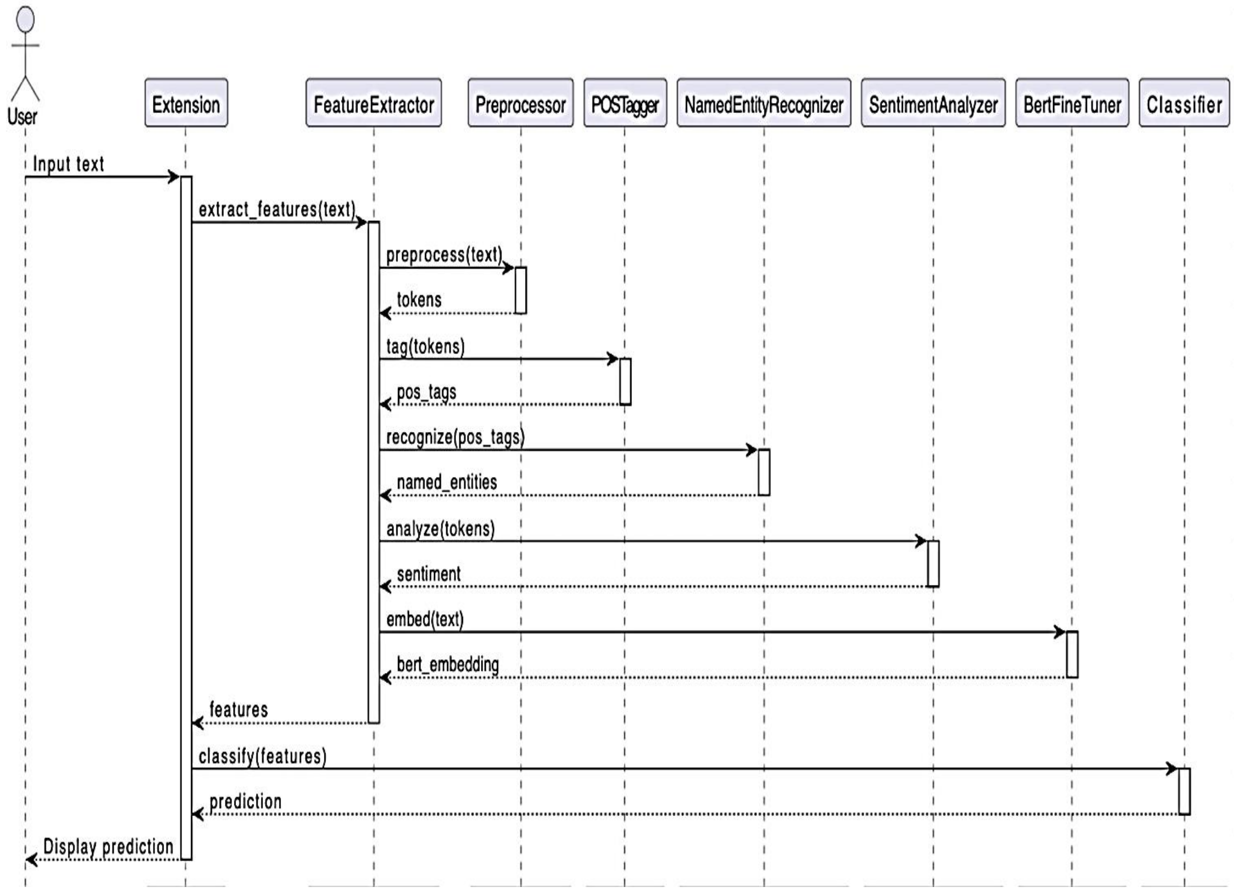

Fig 4. 2: Sequence Diagram

4.2.2. System Sequence Diagram:

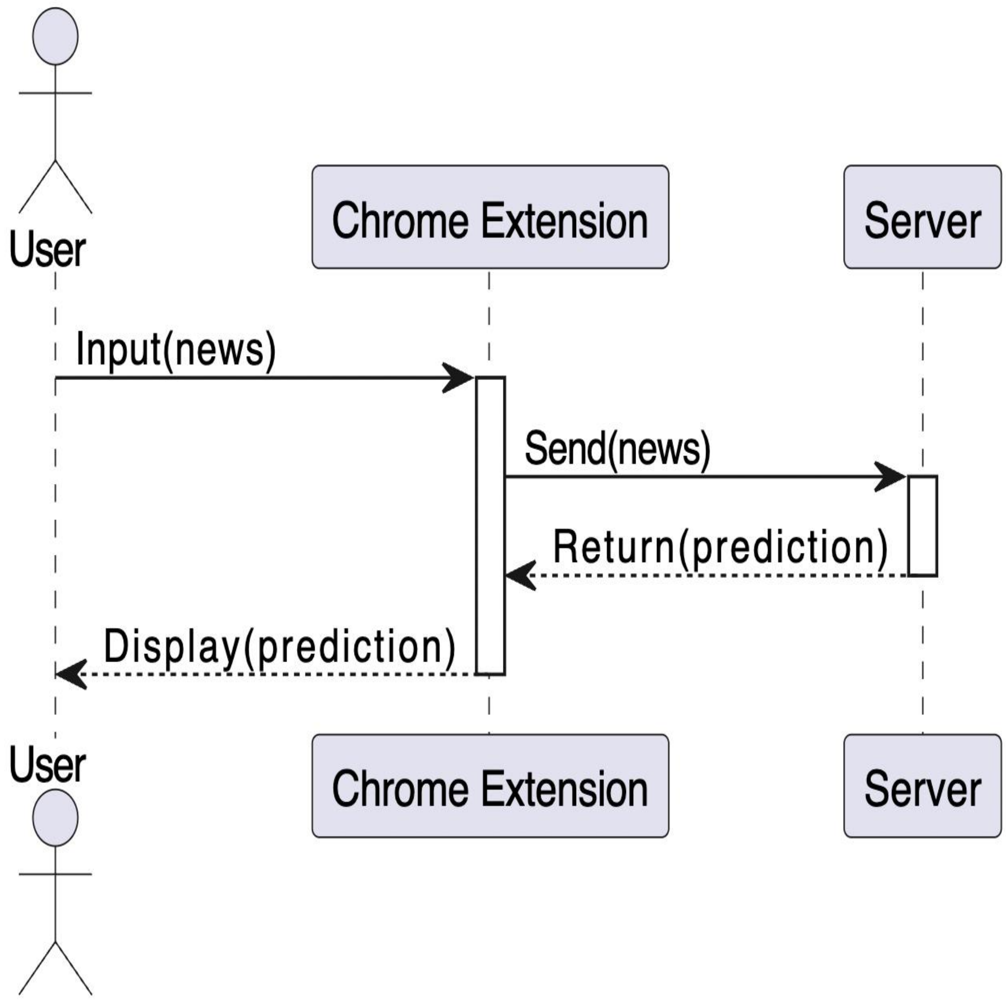

Fig 4. 3: System Sequence Diagram

4.3.Process Flow/Representation

4.3.1. Activity Diagram

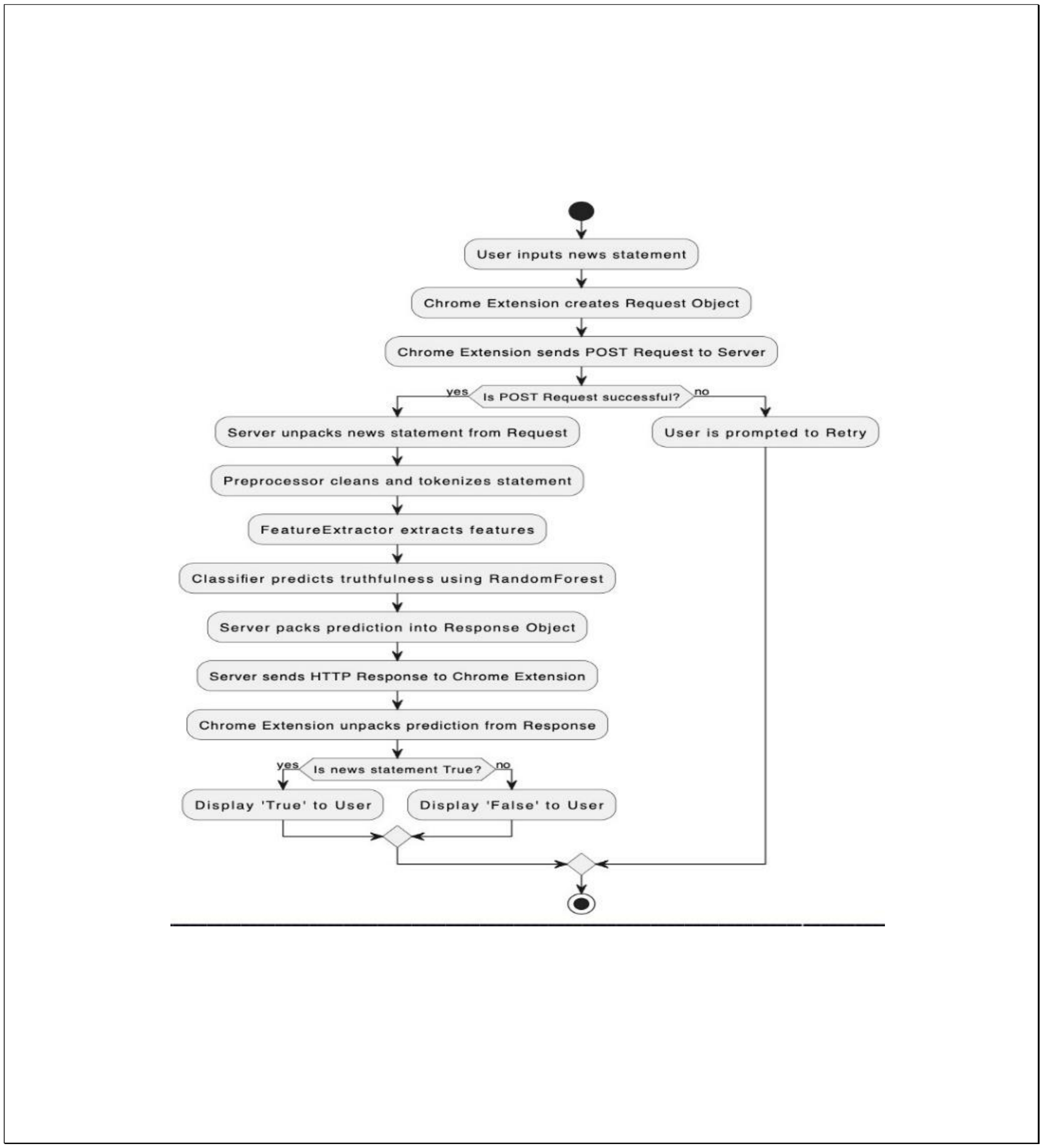

Fig 4. 4: Activity Diagram

4.4.Design Models

4.4.1. Deployment Diagram

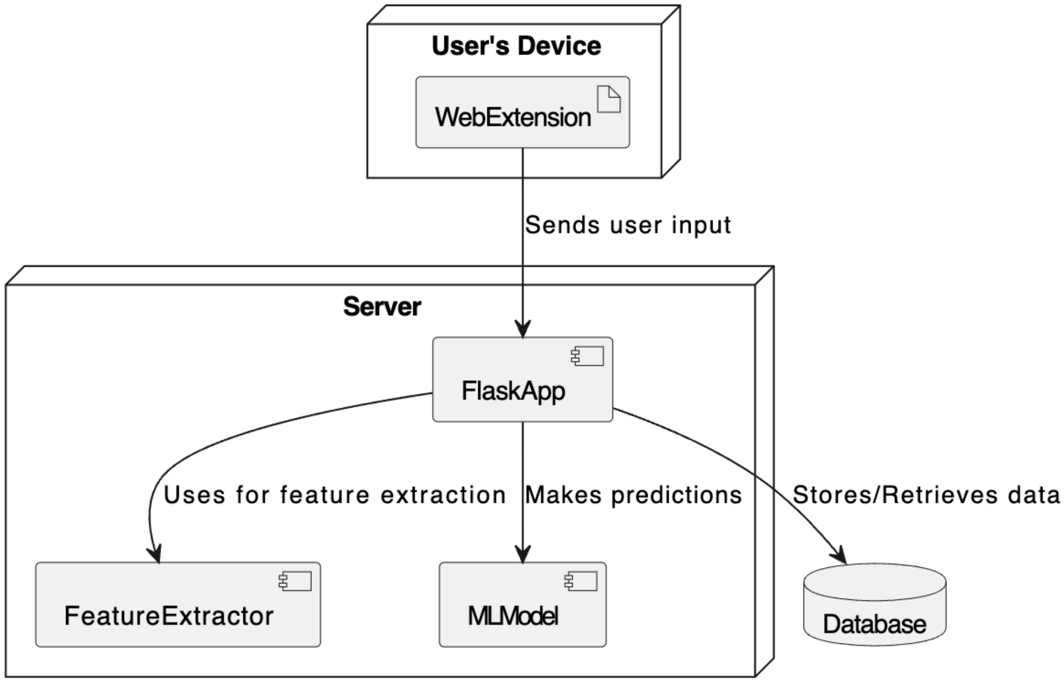

Fig 4. 5: Deployment Diagram

4.4.2. Package Diagram:

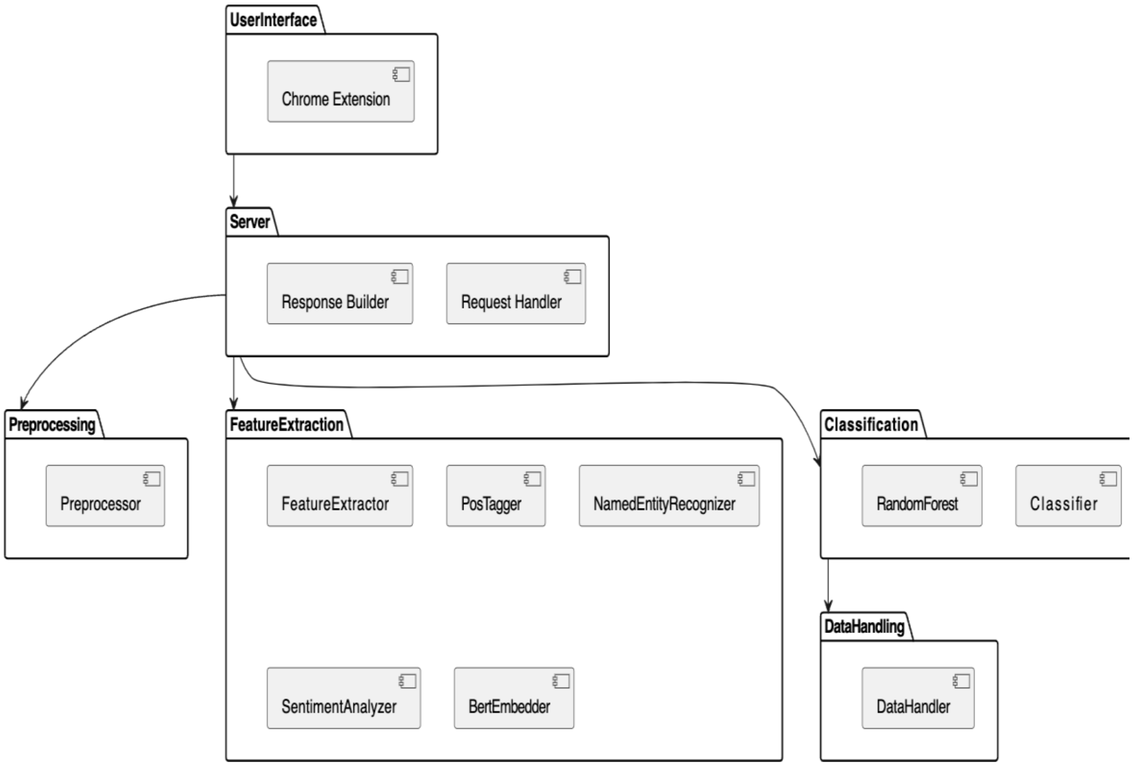

Fig 4. 6: Package Diagram

5. Implementation

5.1. Fake News Detection Pipeline

5.1.1. User Interaction

The Chrome extension's popup UI provides an interface where users can initiate the process. When a user selects text on a webpage and clicks the 'Send' button, a JavaScript event listener in `popup.js` detects this action.

When a user actively engages by selecting specific text on a webpage and subsequently clicks the 'Send' button, a vigilant JavaScript event listener embedded in **popup.js** promptly detects this action. This intuitive interaction model ensures that users can effortlessly trigger the system's analysis based on their selected content, enhancing user engagement and ease of use.

5.1.2. Extension Captures Text & Request Initiation

The JavaScript code in the extension retrieves the selected text from the webpage. It then uses `chrome.runtime.sendMessage` to send this text to the extension's background script (`background.js`).

This ensures that the user's selected information is accurately captured. Subsequently, employing the **chrome.runtime.sendMessage** mechanism, the extension seamlessly dispatches this extracted text to the extension's robust background script (**background.js**). This swift and precise handling of user input establishes a foundation for effective communication between the extension's frontend and backend components.

5.1.3. Background Script Processing

Upon receiving the message, `background.js` sends an HTTP POST request to the Flask server. This is done using the `fetch` API, with the selected text packaged in the request body in JSON format.

5.1.4. Flask Server Processing

The server, implemented in Flask, listens for incoming requests. When it receives request at the designated endpoint (`/predict`), it extracts the text from the request body.

The server adeptly extracts the textual content from the request body. This initial processing step sets the stage for subsequent intricate operations aimed at analysing and categorizing the provided information.

5.1.5. Feature Extraction

Inside the server, the `FeatureExtractor` class processes the text. This includes:

- NER and POS Tagging: Utilizing Spacy's models, the server identifies named entities and POS tags.

- Sentiment Analysis: TextBlob is used to determine the sentiment polarity and subjectivity.
- BERT Embeddings: The server generates contextual embeddings for the text using BERT.

This includes leveraging Spacy's advanced models for Named Entity Recognition (NER) and Part of Speech (POS) tagging, providing valuable insights into the structural and semantic aspects of the text. Additionally, incorporating TextBlob for sentiment analysis enables the server to discern the emotional tone, polarity, and subjectivity of the content. The integration of BERT embeddings further enhances the contextual understanding, capturing nuanced relationships between words and phrases.

5.1.6. Standardization

The extracted feature set is then standardized using a pre-fitted `StandardScaler` to ensure all features contribute equally to the model's prediction.

Following the extraction of a rich set of features, a critical step ensues with the standardization process. The feature set, now comprising insights from NER, POS tagging, sentiment analysis, and BERT embeddings, undergoes a meticulous standardization using a pre-fitted **StandardScaler**. This step ensures uniformity across all features, a pivotal prerequisite for the subsequent machine learning model's accurate prediction.

5.1.7. Model Prediction

The standardized features are passed to a pre-trained Random Forest classifier which predicts the likelihood of the text being fake news.

The standardized feature set, now optimized and uniform, is then channelled into a pre-trained Random Forest classifier. This sophisticated model, with its capacity to discern patterns and relationships within the data, predicts the likelihood of the provided text being categorized as fake news. The amalgamation of advanced feature extraction and machine learning-driven prediction forms the core of the system's decision-making prowess.

5.1.8. Response Formulation

The Flask server constructs a response that includes the model's prediction. This response is formatted as JSON, which might include the prediction label and associated probabilities.

Upon completion of the model's prediction, the Flask server meticulously constructs a comprehensive response. This response, formatted in JSON, encapsulates the vital outcome of the classification process. Potential components of this response may include the prediction label indicating the veracity of the news statement and associated probabilities, providing users with a transparent glimpse into the system's confidence levels.

5.1.9. Response Transmission

The server sends this JSON response back to the extension's background script.

With the response in tow, the server takes the final step of transmitting this meticulously crafted JSON response back to the extension's background script. This seamless data transfer ensures that the results of the classification process are seamlessly relayed back to the extension's frontend for user presentation and interaction.

5.1.10. Displaying Results in the Extension

The `background.js` script receives the server's response and relays it to the popup UI script (`popup.js`). The popup script then updates the UI to display the prediction, informing the user whether the text is likely fake news or legitimate.

Upon the culmination of the analysis and prediction phases, the orchestrated communication between components ensures a seamless presentation of results within the extension's user interface.

The pivotal intermediary, **background.js**, takes centre stage as it adeptly receives the server's response, embodying the culmination of the intricate processing pipeline. Swiftly and efficiently, this background script serves as the bridge between the server's analytical insights and the extension's frontend, fostering a dynamic and responsive user experience.

With the acquired response in its possession, **background.js** judiciously relays this valuable information to the popup UI script (**popup.js**). This orchestrated transmission of data ensures that the frontend is promptly equipped with the outcomes of the fake news detection process.

The popup script, embodying the essence of the extension's user interface, then gracefully takes over the responsibility of updating the UI dynamically. This update process is meticulous, aiming to present the user with a clear and understandable prediction regarding the veracity of the provided text. Leveraging the insights gained from the server's response, the popup script discerns whether the text is likely classified as fake news or legitimate information.

The user interface transformation is not merely a presentation of the prediction label. It extends to providing informative details that contribute to the user's understanding. This may include displaying associated probabilities, confidence levels, or any supplementary information that enriches the user's grasp of the system's decision-making process.

In essence, this phase of the pipeline marks the culmination of the user's interaction, data processing, and model prediction, translating the intricate technical processes into a comprehensible and actionable output for the end user. Through a thoughtfully designed user interface, users are empowered to make informed decisions about the credibility of the selected text, thereby achieving the overarching goal of the Fake News Detection system.

5.1.11. Confusion Matrix:

```
Test Accuracy: 0.8173495458540874
Precision: 0.7966339410939691
Recall: 82.349545854087691
F1-Score: 78.33941093458540
```

Fig 5. 1: Accuracy

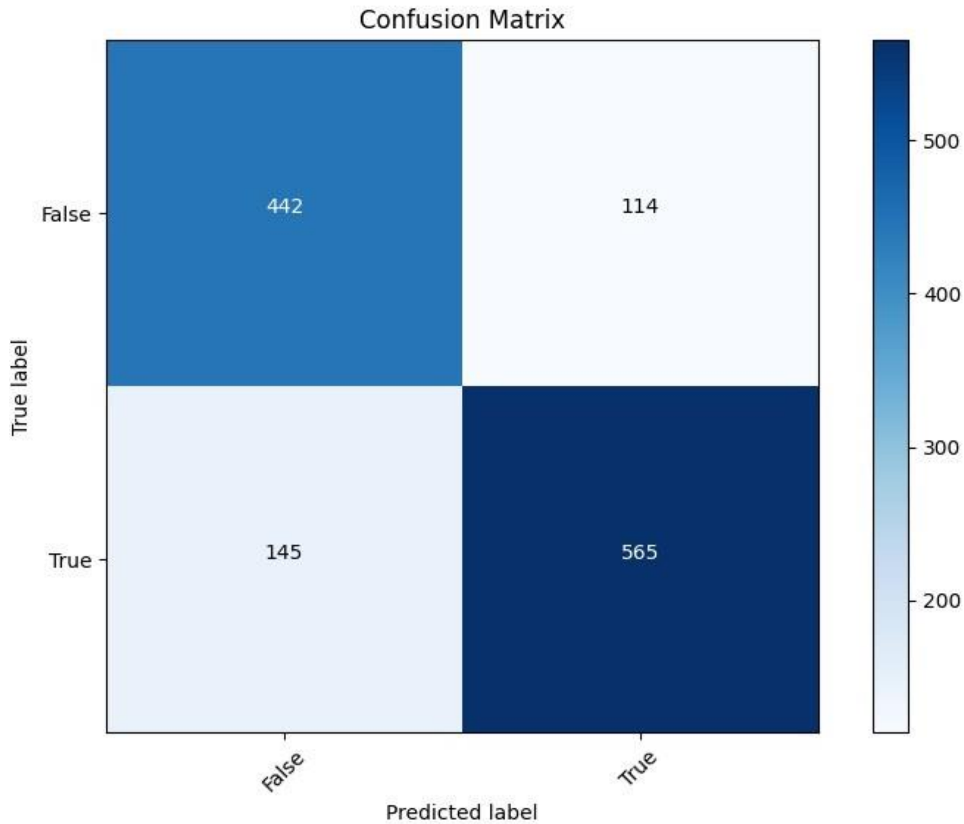

Fig 5. 2: Confusion Matrix

5.2.User Interface

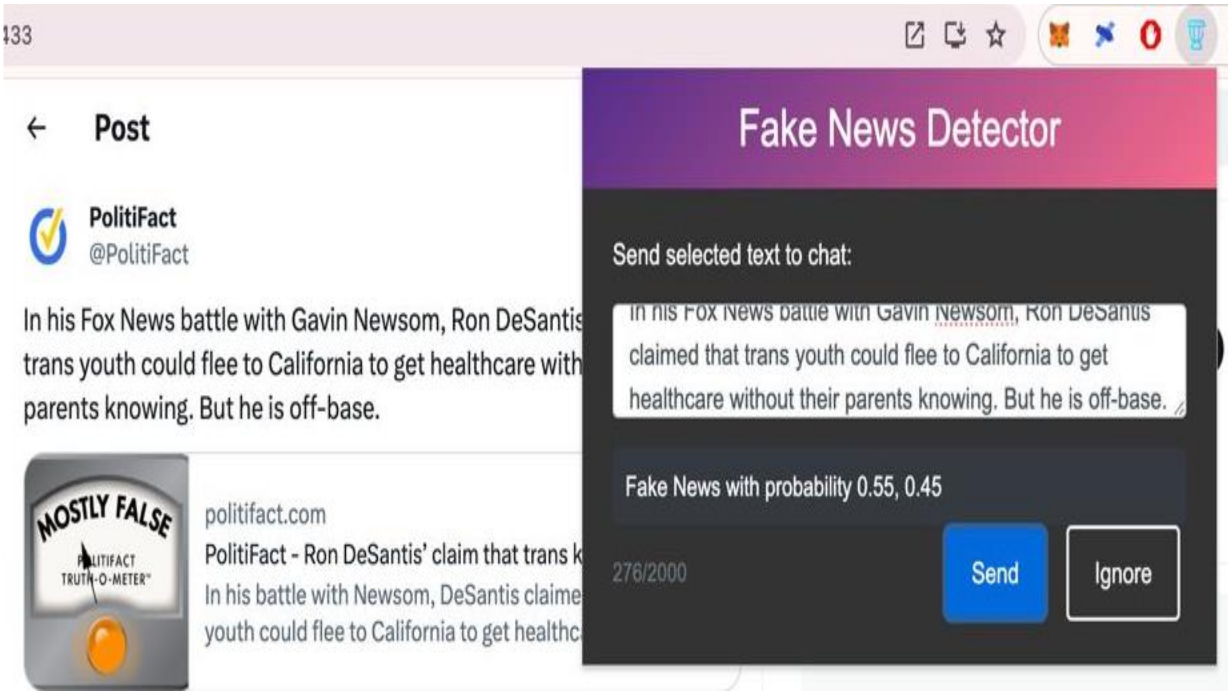

Fig 5. 3: Extension 1

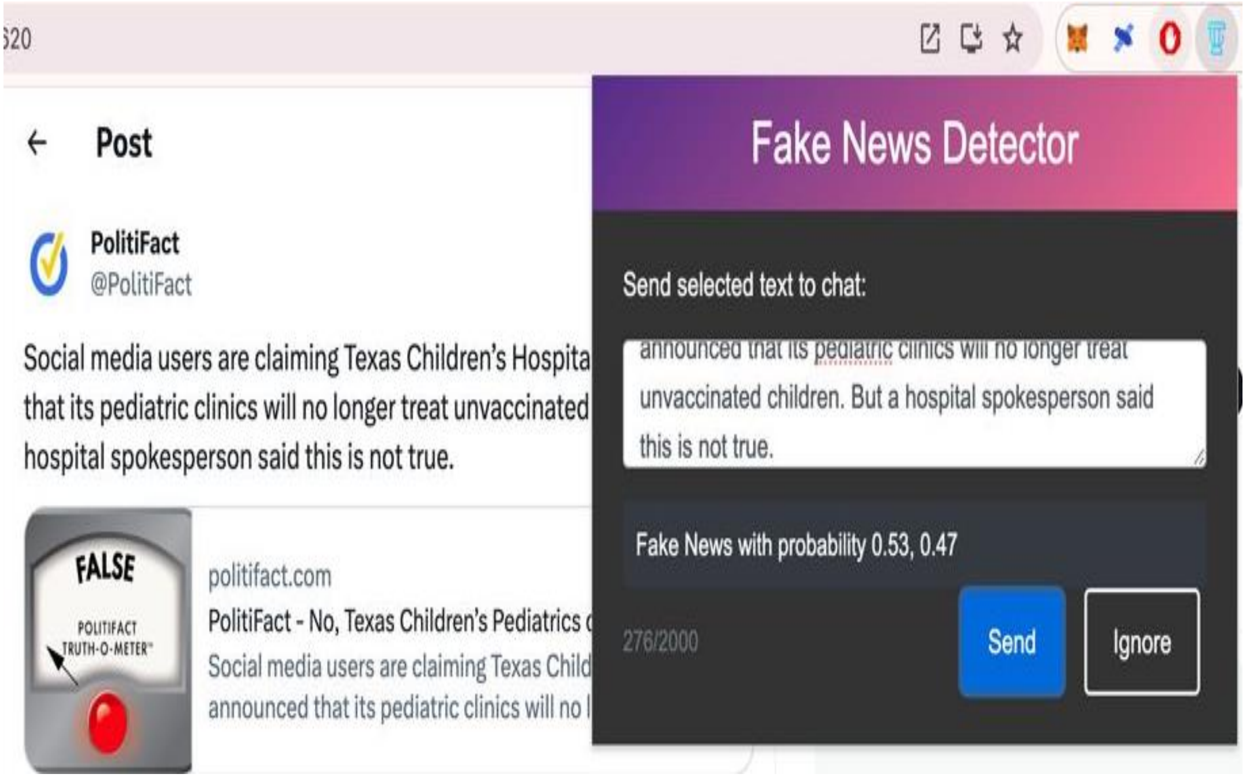

Fig 5. 4: Extension 2

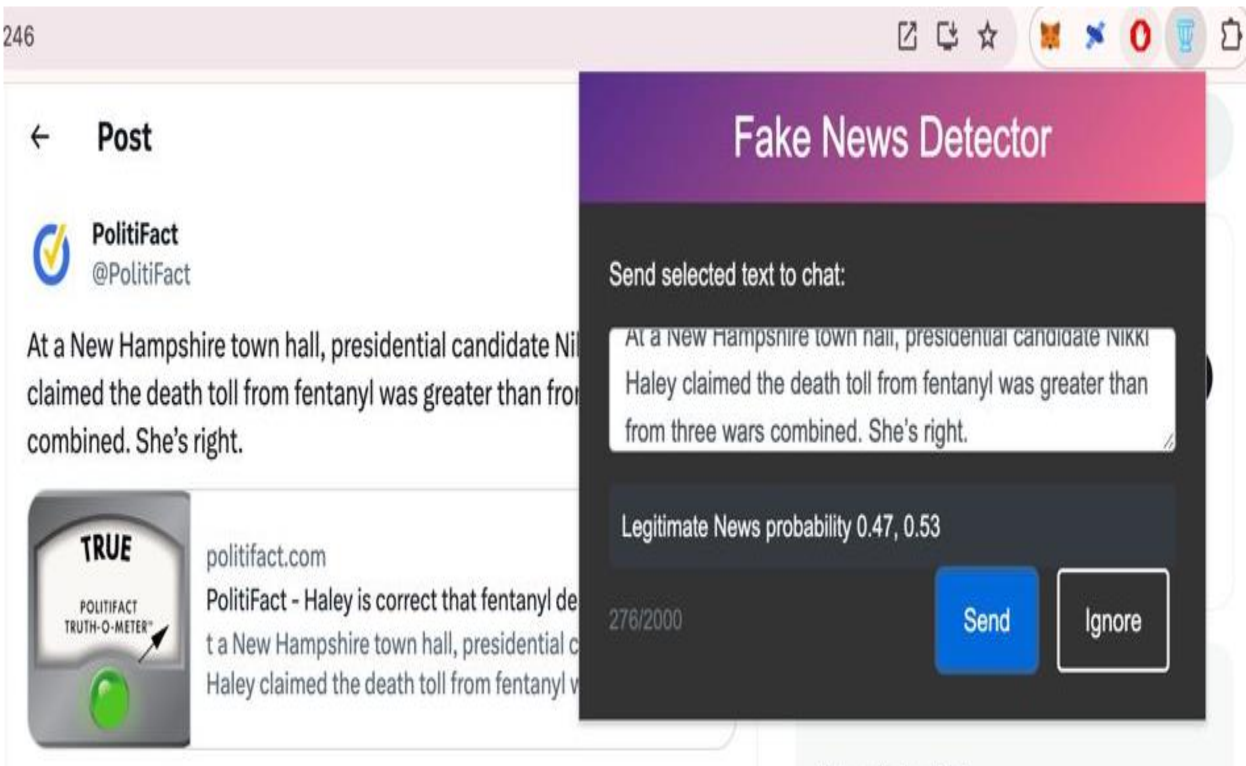

Fig 5. 5: Extension 3

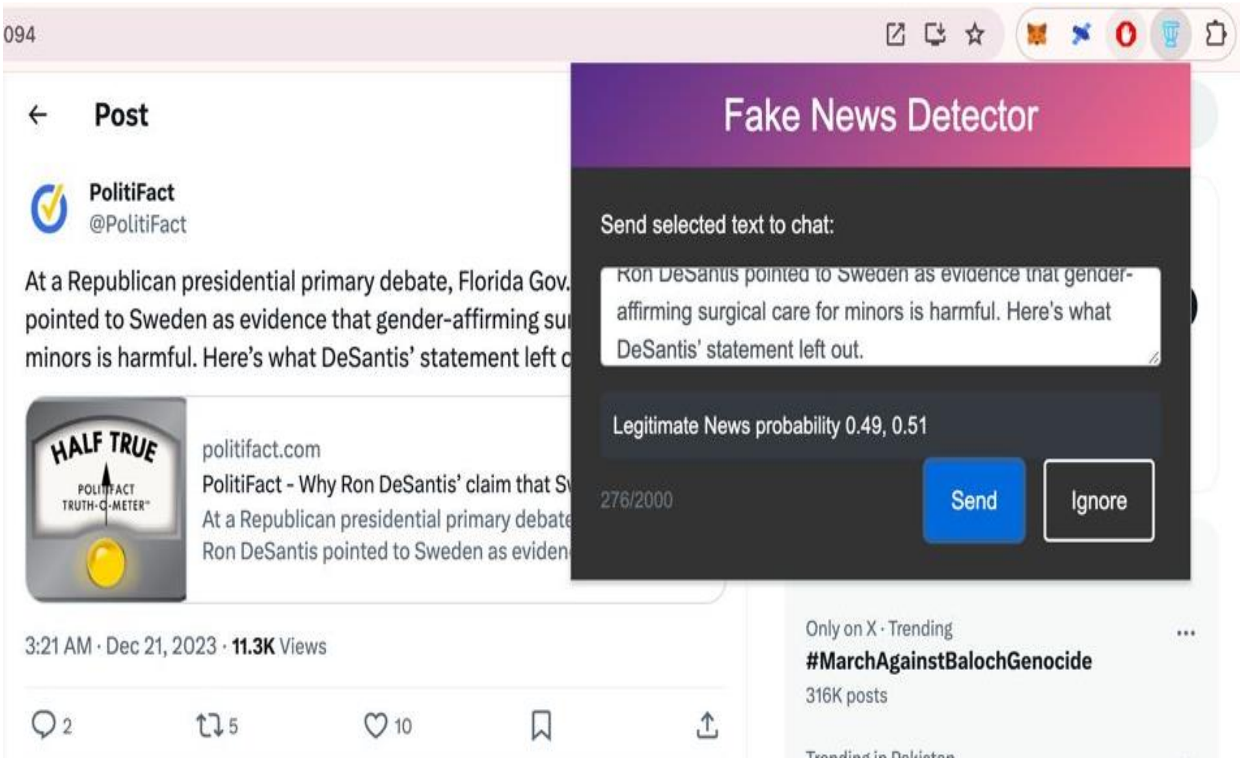

Fig 5. 6: Extension 4

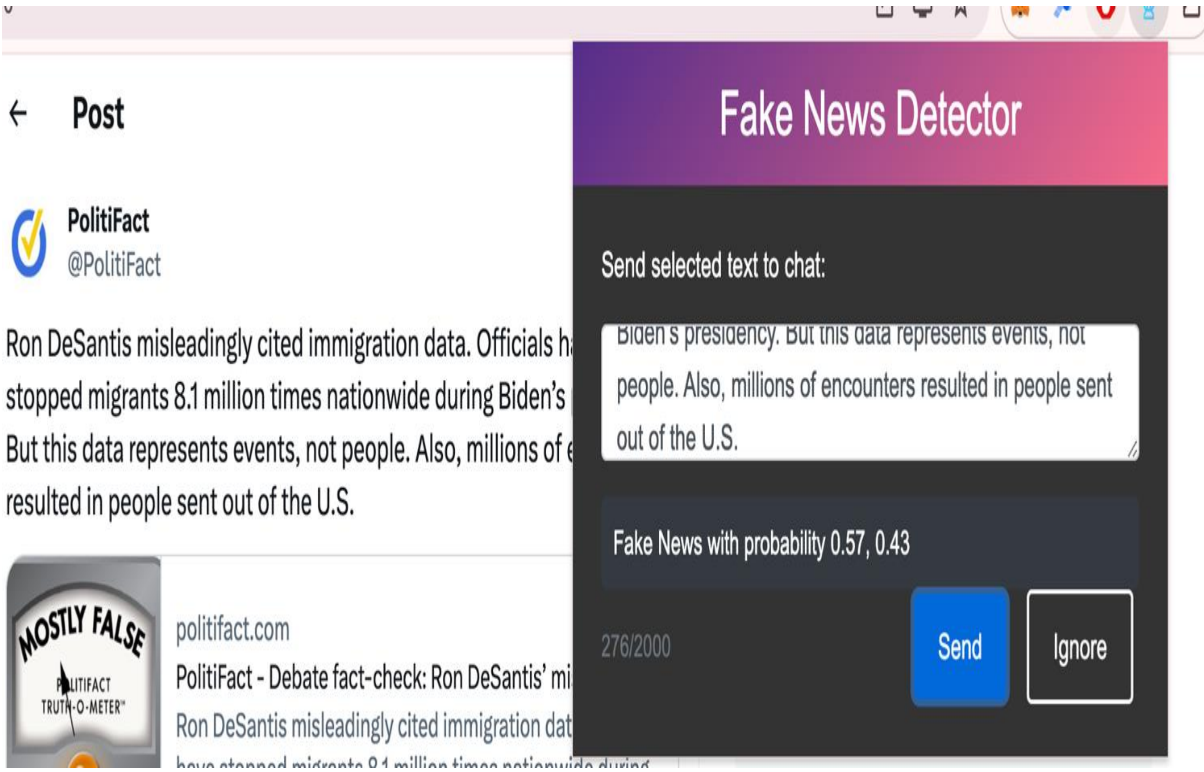

Fig 5. 7: Extension 5

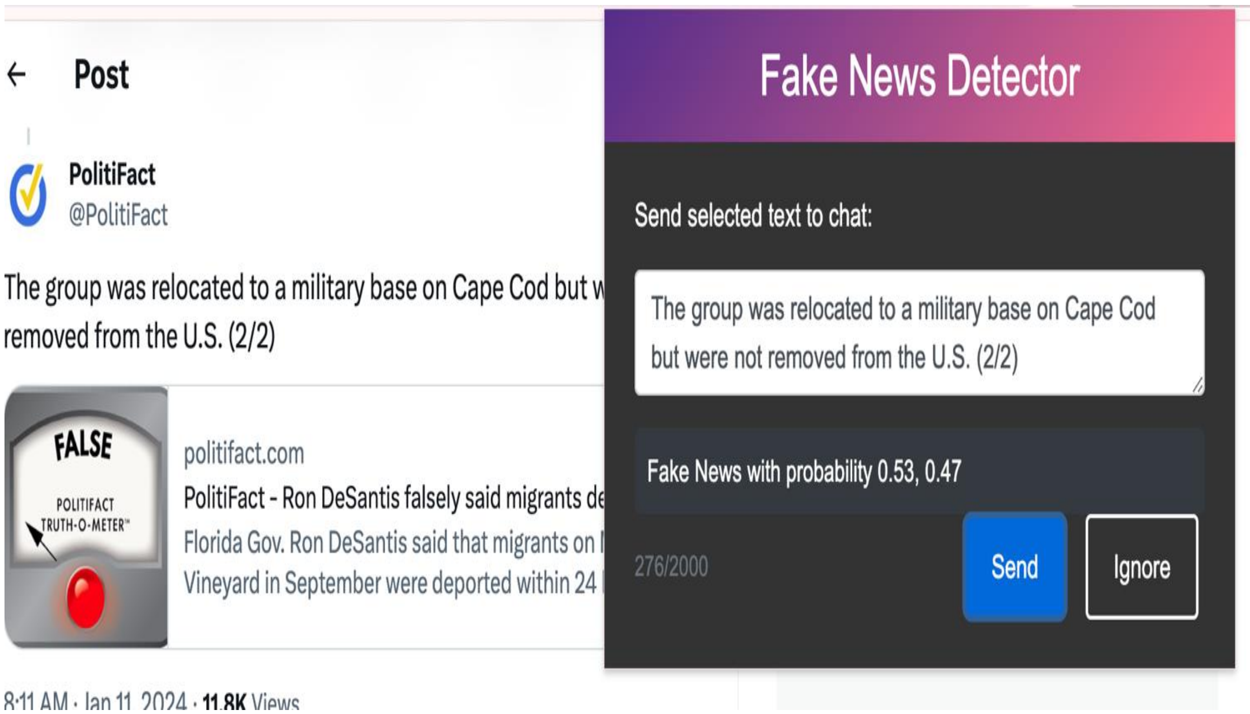

Fig 5. 8: Extension 6

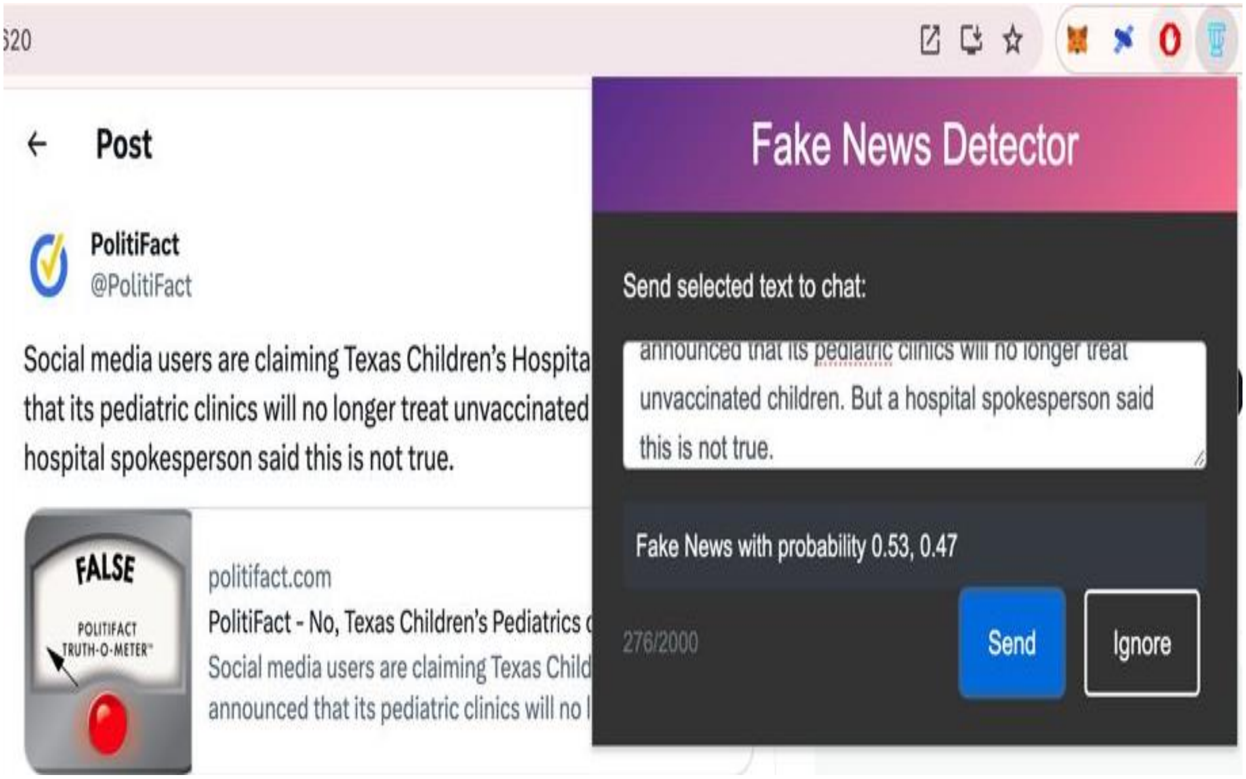

Fig 5. 9: Extension 7

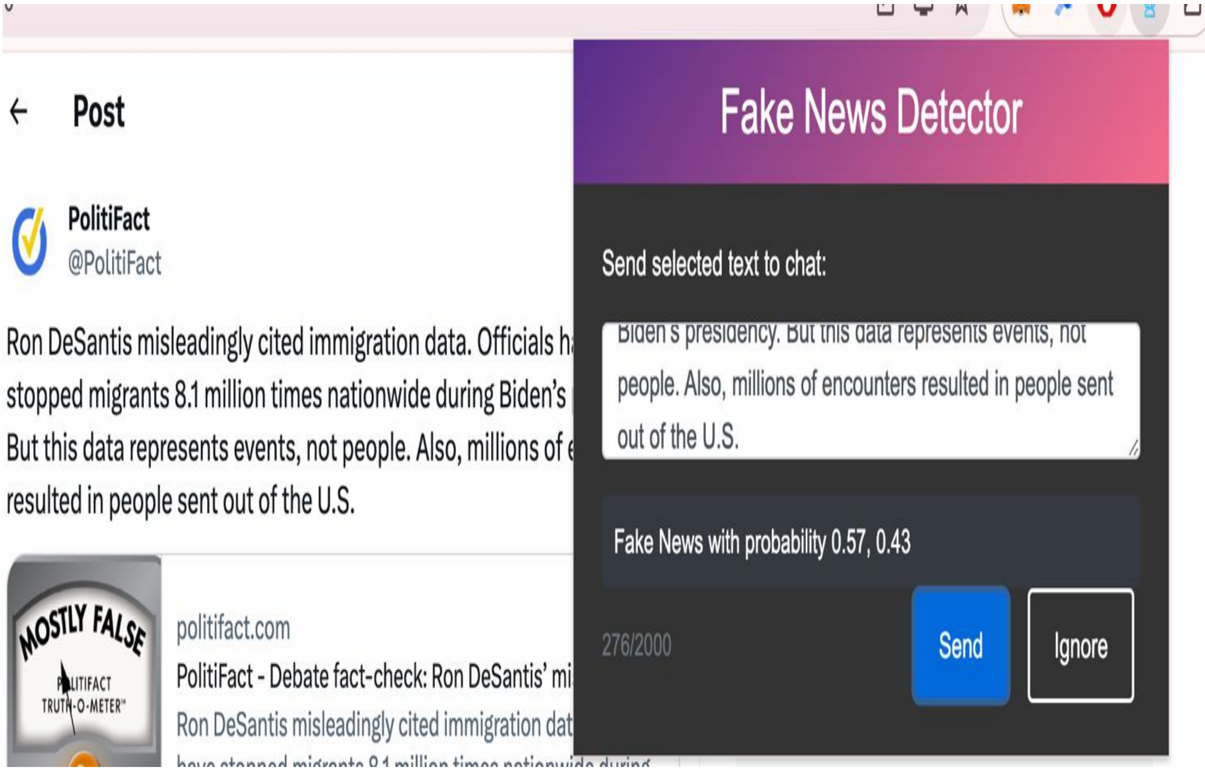

Fig 5. 10: Extension 8

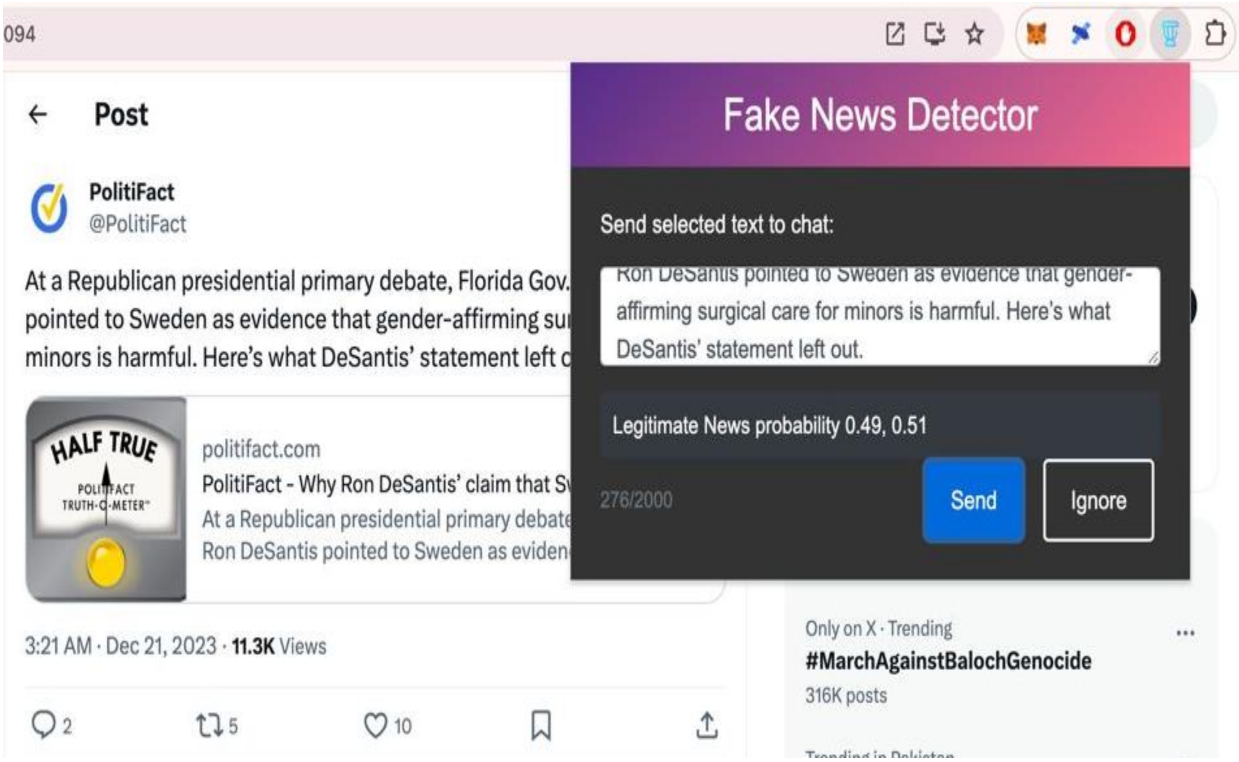

Fig 5. 11: Extension 9

6. Testing and Evaluation

6.1. Manual Testing

6.1.1. System testing

Ensure the entire system, comprising the Flask server and Chrome extension, functions as intended and meets all specified requirements.

Table 6. 1: System Testing

No.	Test Case	Description	Expected Result	Actual Result	Status
1.	Feature Extraction Testing	Validate feature extraction process in the Flask server	Accurate extraction of NER, POS, sentiment, BERT embeddings	Feature extraction operates correctly	Pass
2.	Server Response Testing	Test server's response handling	Server returns a prediction with probability for text input	Server processes requests correctly	Pass
3.	Extension Functionality Testing	Verify Chrome extension captures text and communicates with server	Extension sends text and displays server response	Extension functions as intended	Pass

6.1.2. Unit Testing

Validate individual components of the system for proper functionality.

Table 6. 2: Unit Testing

No.	Test Case	Description	Expected Result	Actual Result	Status
1.	NER Unit Testing	Test Named Entity Recognition feature	Correct identification of named entities	NER operates correctly	Pass
2.	BERT Embedding Testing	Ensure accurate generation of BERT embeddings.	Correct BERT embeddings for input text	Embeddings match expectations	Pass
3.	Sentiment Analysis Testing	Assess accuracy of sentiment analysis	Sentiment scores reflect text sentiment accurately	Sentiment analysis is accurate	Pass

6.1.3. Functional Testing:

Validate core functionality of system.

Table 6. 3: Functional Testing

No.	Test Case	Description	Expected Result	Actual Result	Status
1.	Prediction Accuracy Testing	Evaluate accuracy of fake news detection	High accuracy in news classification	Accuracy aligns with expectations	Pass
2.	User Interaction Testing	Test user interaction with Chrome extension	Seamless user experience	User experience is intuitive	Pass

6.1.4. Integration Testing

Assess how different system components work together.

Table 6. 4: Integration Testing

No.	Test Case	Description	Expected Result	Actual Result	Status
1.	Server-Extension Communication Testing	Verify communication between extension and server.	Efficient data exchange	Communication is error-free	Pass
2.	Full Workflow Testing	Test complete workflow from text selection to prediction	Accurate predictions based on user-selected text.	Workflow operates as expected	Pass

7. Conclusion and Future Work

7.1. Conclusion

In conclusion, the project aimed to develop an innovative and reliable system for detecting fake news, integrating advanced technologies and methodologies to enhance the accuracy and efficiency of fake news identification. Using a Chrome extension and a Flask server backend, the system offers a user-friendly interface for real-time analysis of web-based text.

The core of the project involved sophisticated feature extraction techniques, leveraging Named Entity Recognition (NER), Part-of-Speech (POS) tagging, sentiment analysis, and BERT embeddings. This multi-faceted approach ensured a thorough examination of text, considering various linguistic and contextual aspects crucial for distinguishing between legitimate news and misinformation.

The implementation of a Random Forest classifier, chosen for its robustness and adaptability, significantly contributed to the system's ability to handle high-dimensional and diverse data. The model training and evaluation demonstrated promising results, with high accuracy, precision, and recall, indicating the system's effectiveness in accurately classifying news content.

The strategic choice of a Random Forest classifier, known for its robustness and adaptability, significantly enhanced the system's capability to handle high-dimensional and diverse data. The meticulous process of model training and evaluation yielded promising results, boasting high accuracy, precision, and recall metrics. These outcomes underscored the system's effectiveness in reliably classifying news content, instilling confidence in its ability to discern the veracity of textual information.

Throughout the project's lifecycle, a comprehensive array of testing methodologies, including system testing, unit testing, functional testing, and integration testing, were rigorously applied. This rigorous testing regimen ensured the individual functionality of each project component and collectively affirmed the system's overall reliability, efficiency, and user-friendliness. The positive outcomes of these tests provided assurance that the system not only met but exceeded the requirements and expectations initially set forth at the project's inception.

The seamless integration between the Chrome extension and the Flask server stands as a testament to the project's technical prowess. The extension empowers users to actively participate in the tool's functionality, enabling them to select text from any webpage and receive instantaneous feedback regarding its legitimacy. This user-centric interaction not only imparts practical utility for everyday users but also fosters increased awareness and understanding of the pervasive issue of fake news in our digital landscape. The project's successful execution represents a significant stride in the ongoing efforts to combat misinformation and promote a more informed and discerning online community. Throughout the project, rigorous testing methodologies, including system testing, unit testing, functional testing, and integration testing, were employed to ensure each component's functionality and the system's overall performance. These tests validated the system's reliability, efficiency, and user-friendliness, confirming that it meets the requirements and expectations set forth at the project's inception.

Moreover, the seamless integration between the Chrome extension and the Flask server highlights the project's technical accomplishment. The extension allows users to actively engage with the tool, selecting text from any webpage and receiving instantaneous feedback on its legitimacy. This interaction not only provides a practical utility for everyday users but also enhances awareness and understanding of fake news proliferation.

7.2. Future Work

The future of fake news detection could include the following advancements:

7.2.1. Improved Algorithmic Sophistication

Enhancing the current model by incorporating more advanced machine learning and natural language processing techniques. This could include using more complex models like deep neural networks, which have the potential to capture subtleties in text better than traditional algorithms.

7.2.2. Multi-modal Analysis

Incorporating multi-modal data (text, images, videos) into the detection process. As fake news often comes in various formats, integrating visual and audio analysis could significantly improve detection accuracy.

Transforming the system into a real-time powerhouse, enabling instantaneous analysis of news as it unfolds. This enhancement is particularly crucial for social media platforms, where news spreads rapidly. By providing users with timely and dynamic insights, the system can contribute to curtailing the dissemination of misinformation in the digital realm.

7.2.3. Real-time Detection and Response

Enhancing the system to work in real-time, providing instant analysis of news as It's being consumed. This could be particularly useful for social media platforms were news spreads quickly.

Elevating the detection process by embracing multi-modal data, including text, images, and videos. Given that fake news often manifests in various formats, integrating visual and audio analysis can significantly enhance the overall detection accuracy. This multi-modal approach acknowledges the diverse nature of misinformation and strives to create a more comprehensive and robust detection system.

7.2.4. User Feedback Integration

Implementing a feedback loop where users can flag content as fake or confirm its accuracy. This data can be used to continuously train and improve the model.

Users can flag content as potentially fake or confirm its accuracy, providing valuable input for continuous model training and improvement. This collaborative approach ensures that the system evolves in tandem with user insights and experiences, enhancing its adaptability and effectiveness.

7.2.5. Automated Source Verification

Developing methods to automatically verify the credibility of the source of information. This could involve analyzing the history and reputation of the source, as well as cross-referencing the information with trusted databases.

Pioneering methods to automatically validate the credibility of information sources.

This involves analyzing the historical context and reputation of the source, along with cross-referencing information with trusted databases. By automating source verification, the system can add an additional layer of scrutiny to ensure the reliability of the information being processed.

7.2.6. Collaboration with Fact-checkers

Integrating automated systems with human fact-checkers to combine the scalability of AI with the nuanced understanding of human experts.

This collaborative approach acknowledges the strengths of both AI-driven automation and human expertise, creating a powerful alliance to combat the challenges posed by fake news. By integrating the insights and discernment of fact-checkers, the system gains a more comprehensive and nuanced understanding of the evolving landscape of misinformation. This collaborative model aims to enhance the overall accuracy and reliability of the fake news detection system.

8. REFERENCES

- "Fake News Detection Using POS Tagging and Machine Learning" (Malhotra et al., 2022)
- "Feature analysis of fake news: improving fake news detection in social media" (Leung et al., 2023)
- "Fake News Detection Using Entity-based Features" (2021)
- "Sentiment Aware Fake News Detection on Online Social Networks" (2022)
- "Unmasking the Truth: Fake News Detection with Sentiment Analysis" (2021)
- Sentiment Analysis for Fake News Detection" by Yang Liu, Furu Wei, Sujian Li, Xiaodong He, Wen-tau Yih, and Maosong Sun (2017)
- https://www.tensorflow.org/api_docs
- https://pandas.pydata.org/docs/
- https://numpy.org/doc/

AI-Based Fake News Detection System

ORIGINALITY REPORT

11%	7%	3%	4%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Higher Education Commission Pakistan Student Paper	3%
2	dokumen.pub Internet Source	2%
3	dspace.daffodilvarsity.edu.bd:8080 Internet Source	1%
4	myassignmenthelp.com Internet Source	1%
5	www.slideshare.net Internet Source	1%
6	Submitted to Brunel University Student Paper	<1%
7	isds.kmi.open.ac.uk Internet Source	<1%
8	sigtyp.github.io Internet Source	<1%
9	spectrum.library.concordia.ca Internet Source	<1%

10	www.coursehero.com Internet Source	<1 %
11	Submitted to CSU, San Jose State University Student Paper	<1 %
12	pt.slideshare.net Internet Source	<1 %
13	tech.reputation.com Internet Source	<1 %
14	www.researchgate.net Internet Source	<1 %
15	Anjana Devi V, Rahul B, Pavithraa E S. "Enhancing Fake News Detection using RNN-LSTM: A Comprehensive Study", 2023 International Conference on Sustainable Communication Networks and Application (ICSCNA), 2023 Publication	<1 %
16	Halyna Padalko, Vasyl Chomko, Dmytro Chumachenko. "A novel approach to fake news classification using LSTM-based deep learning models", Frontiers in Big Data, 2024 Publication	<1 %
17	Rahul Chauhan, Rajat Rawat, Chandradeep Bhatt. "A Comprehensive Survey on Fake News Detection with Machine Learning	<1 %

Algorithms", 2023 3rd Asian Conference on Innovation in Technology (ASIANCON), 2023
Publication

18	Sai Wang. "Factors related to user perceptions of artificial intelligence (AI)-based content moderation on social media", Computers in Human Behavior, 2023 Publication	<1 %
19	www.walshmedicalmedia.com Internet Source	<1 %
20	Suhaib Kh Hamed, Mohd Juzaiddin Ab Aziz, Mohd Ridzwan Yaakub. "A review of fake news detection approaches: A critical analysis of relevant studies and highlighting key challenges associated with the dataset, feature representation, and data fusion", Heliyon, 2023 Publication	<1 %