

###### The Islamia University of Bahawalpur

**Bahawalnagar Campus**

## SOFTWARE REQUIREMENTS SPECIFICATION

**(SRS DOCUMENT)**

## for

#### <PROJECT NAME>

Version 1.0

***By***

**Eisha Fatima F21NINFT1M01047**

**Anam Zulfiqar F21NINFT1M01065**

**Muhammad Billal F21NINFT1M01046**

Supervisor

**Mr.Shaid Hussain**

***Bachelor of Science in Computer Science (20xx-20xx)***

**Table of Contents**

**Revision History 28**

1. **Introduction 30**
   1. Purpose 30
   2. Scope 30
2. **Overall description 30**
   1. Product perspective 30
   2. Operating environment 30
   3. Design and implementation constraints 30
3. **Requirement identifying technique 31**
   1. Use Case Diagram 31
   2. Use Case Description 31
4. **Specific Requirements 33**
   1. System feature X 33
5. **Quality attributes** .
   1. Usability 34
   2. Performance 35
6. **External interface requirements 35**
   1. User interfaces 35
   2. Software interfaces 35
   3. Hardware interfaces 35
   4. Communications interfaces 35
7. **Project Gantt chart 35**
8. **References 35**

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason for changes** | **Version** |
|  |  |  |  |
|  |  |  |  |

**Application Evaluation History**

|  |  |
| --- | --- |
| **Comments (by committee)**  **\*include the ones given at scope time both in doc and presentation** | **Action Taken** |
|  |  |
|  |  |

**Supervised by**

**<Supervisor’s Name>**

Signature

###### Introduction

With an active user base of over 368 million users per month, Twitter is one of the largest real time data sources on the internet with 500 million tweets a day. The platform has turned out to be a very powerful tool to gauge public sentiment about a wide variety of subjects: political events, product reviews, and social movements. This data can provide a wealth of value for sentiment analysis to give businesses e insight into what customers are saying about them, what the brand is saying, and their public opinion on events. The problem of analyzing twitter data is complicated by the sheer volume, the text is unstructured, and the use of language is diverse (slang, emojis, abbreviations, and hashtags). It is found that 80% of the data in social media platforms is unstructured and needs sophisticated techniques to extract meaningful insights (Agarwal et al., 2023; Kumar & Arora, 2021; Valerian Chinedum et al., 2023).

However, existing sentiment analysis models tend to have a hard time handling these challenges, and get only 60% accuracy when used on noisy social media text. Simple rule based sentiment analysis or basic machine learning approach such as SVM cannot capture the nuances of Twitter posts using informal language. Additionally, the trend of trending topics on social media is dynamic, and thus a solution that can scale to handle real time data is needed (Chioma et al., 2016; Shu et al., 2017; Vardhan et al., 2022).

To solve these problems, this research presents a machine learning based sentiment analysis system which uses both desktop and mobile applications to real time analyze Twitter data. Advanced supervised learning techniques like Support Vector Machine (SVM) and Logistic Regression, as well as deep learning models like Long Short-Term Memory (LSTM) networks and Bidirectional Encoder Representations from Transformers (BERT), will be used in the system to achieve accuracy improvement of up to 15 percent in sentiment classification tasks. Preprocessing methods such as tokenization, stemming, and feature extraction using methods like TF-IDF and Word Embedding’s will also be used to work with noisy text to enhance model performance.

The developed system attempts to integrate a dual platform approach to offer a user friendly interface for sentiment analysis, data visualization, and trend monitoring. With this comprehensive solution, businesses and organizations will be empowered to base their decisions on real time social media sentiment, and respond to public opinion trends quickly and accurately.

Purpose

Identify The purpose of the proposed sentiment analysis system is to create an advanced, scalable, and efficient solution for analyzing real-time Twitter data. By combining machine learning and deep learning techniques, the system aims to overcome the limitations of traditional sentiment analysis methods, which often fail to handle the complexities of noisy, unstructured, and informal text data prevalent on social media.

The system is designed to achieve the following key objectives:

**Accurate Sentiment Analysis:** Enhance the accuracy of sentiment classification tasks by utilizing advanced models such as Support Vector Machines (SVM), Naive Bayes, Long Short-Term Memory (LSTM) networks, and Bidirectional Encoder Representations from Transformers (BERT). These methods improve the ability to process and interpret complex linguistic patterns, including slang, mixed sentiments, and context-dependent expressions.

**Real-Time Data Processing:** Enable seamless real-time data collection and analysis through integration with the Twitter API, ensuring that businesses, researchers, and policymakers can monitor and respond to emerging trends promptly.

**Handling Noisy and Unstructured Data:** Implement sophisticated preprocessing techniques, such as tokenization, stemming, and feature extraction (e.g., TF-IDF and word embeddings), to manage the challenges posed by unstructured text data and improve model performance.

**Dual Platform Accessibility:** Provide a user-friendly, dual-platform system (desktop and mobile) for analyzing and visualizing sentiment trends, making the solution accessible and convenient for diverse user groups across devices.

**Scalability and Versatility:** Ensure that the system can handle large volumes of dynamic social media data efficiently and adapt to a wide variety of applications, including brand monitoring, political analysis, market research, and public opinion tracking.

Scope

It This sentiment analysis system is an innovative solution tailored to process and classify real-time data from Twitter, providing actionable insights for businesses, researchers, and policymakers. The system employs a hybrid approach, combining traditional machine learning models such as Support Vector Machine (SVM) and Naive Bayes with advanced deep learning architectures like Long Short-Term Memory (LSTM) networks and BERT. This approach ensures high accuracy in handling the complexities of Twitter data, which often includes noisy, unstructured, and informal language characterized by slang, abbreviations, emojis, and mixed sentiments.

To enhance model performance and reliability, the system integrates robust preprocessing techniques, including tokenization, stemming, stop-word removal, and feature extraction methods such as Term Frequency-Inverse Document Frequency (TF-IDF) and word embeddings. These preprocessing steps ensure the input data is clean and meaningful, paving the way for more accurate sentiment classification.

The system is designed as a dual-platform solution, offering both desktop and mobile applications with consistent and intuitive user interfaces. This ensures accessibility and flexibility for users across different devices. Core features include real-time tweet collection via API integration, efficient sentiment classification using cutting-edge models, and dynamic data visualization through interactive graphs and charts. By enabling users to monitor trends and sentiment patterns in real time, the system caters to a wide range of applications, such as brand reputation management, political opinion tracking, and market analysis.

**Scope and Limitations**

This initial iteration of the system focuses exclusively on text-based sentiment analysis of English-language tweets. It does not currently extend to multimedia content such as images or videos, nor does it support other languages. Future enhancements may include multilingual support and multimodal analysis to address these limitations, broadening its utility and appeal. By focusing on its current scope, the system ensures a highly optimized and efficient tool tailored to meet immediate market demands while setting a foundation for future expansion.

In summary, this sentiment analysis system combines cutting-edge technologies with practical usability, delivering a scalable, high-performance solution for real-time social media analysis. It empowers users to harness the vast potential of Twitter data for strategic decision-making and trend monitoring across industries

.

###### Overall description

Product perspective

The proposed sentiment analysis system is a new product designed to address the challenges of analyzing real-time Twitter data with a focus on accuracy, scalability, and user accessibility. Its development is rooted in the growing demand for advanced tools capable of extracting meaningful insights from the vast, unstructured, and noisy data generated on social media platforms. Unlike existing solutions that are often limited in scope or unable to handle the complexities of informal language, this system represents a novel integration of cutting-edge machine learning and deep learning techniques with real-time data collection capabilities.

Context and Origin

This product originates from the need for a comprehensive and accessible sentiment analysis system that caters to modern data requirements. While many existing tools either focus on rule-based approaches or rely on basic machine learning techniques, they often fall short in addressing challenges such as:

**Volume and Velocity:** The system is designed to handle the massive volume of tweets generated on Twitter every second. Leveraging advanced API integration, it ensures real-time data collection and processing, enabling users to analyze trends as they emerge. This capability is critical for time-sensitive applications like brand reputation monitoring or political sentiment tracking.

**Data Complexity:** Twitter data often includes informal expressions, slang, abbreviations, emojis, and mixed sentiments within a single tweet. The system uses robust preprocessing techniques like tokenization, stemming, and advanced feature extraction to clean and structure this noisy data. These steps improve the accuracy and reliability of sentiment classification, making it highly effective in understanding diverse user expressions.

**Dynamic Trends:** Trending topics and hashtags on Twitter change rapidly, requiring the system to adapt in real time. By continuously fetching and analyzing fresh data, the system ensures that users stay updated with the latest trends. This dynamic adaptability makes the tool valuable for businesses, marketers, and researchers aiming to respond promptly to emerging themes.

**Position in the Market**

The proposed system is neither an update to a mature system nor part of an existing product line. Instead, it is a standalone product tailored to fill a critical gap in the market by offering:

**Dual Platform Integration:** The system offers seamless functionality across desktop and mobile applications, catering to users who need flexibility in accessing sentiment analysis on different devices. This dual-platform integration ensures consistency in features and user experience, making the tool accessible and versatile.

**Advanced Technology Integration:** By integrating advanced NLP models like LSTM and BERT, the system provides cutting-edge sentiment analysis capabilities. These models enable the tool to understand the nuanced and contextual nature of social media text, setting it apart from traditional sentiment analysis tools.

**Real-Time Insights:** The system employs Twitter API for live data collection and analysis, enabling users to track sentiment trends in real time. This capability empowers users to make informed, time-critical decisions based on the latest social media dynamics, providing a competitive edge in fast-paced industries.

Design and implementation constraints

In the development of the proposed sentiment analysis system for Twitter data, certain constraints must be considered to ensure compatibility, scalability, and efficiency.

**Programming language** **restrictions** arise due to the reliance on pre-existing machine learning and deep learning libraries, such as TensorFlow, PyTorch, and scikit-learn, which are predominantly optimized for Python. Python's extensive ecosystem for natural language processing, including libraries like NLTK, SpaCy, and Hugging Face, makes it the logical choice for implementing the system.

**Real-time data collection** depends on integration with the Twitter API, which has robust support for Python, further solidifying its necessity. Another constraint is the reliance on existing models and frameworks like LSTM and BERT, which require significant computational resources. This necessitates the use of high-performance hardware and cloud computing platforms compatible with GPU acceleration. Furthermore, the dual-platform requirement for desktop and mobile applications imposes the need to choose frameworks like Flask or Django for backend development and Flutter or React Native for cross-platform frontend development to ensure a consistent user experience. These constraints are essential to maximize development efficiency, maintain compatibility with existing technologies, and achieve the desired system performance.

###### Data Flow Diagram (DFD)

Data Flow Diagram (DFD) for your proposed sentiment analysis system. Here's a breakdown of the flow and processes depicted:

**Data Collection:**

The first step in the system involves collecting data, likely from Twitter via the Twitter API. This is the starting point of the entire system, gathering real-time tweets for analysis.

**API Integration:**

The data is then processed through API Integration, which ensures that the data collection step communicates seamlessly with external APIs (such as Twitter's API) to fetch live data.

**Data Filtering:**

After data collection, the raw data undergoes Data Filtering. This step removes irrelevant or noisy data, ensuring that the input for further analysis is meaningful.

**Data Preprocessing:**

Once filtered, the data moves into Data Preprocessing. This stage involves transforming the data into a format suitable for machine learning and deep learning models, such as tokenization, stemming, and handling missing values.

**Feature Extraction:**

In the Feature Extraction phase, relevant features are extracted from the processed data. Methods like TF-IDF, word embeddings, and other text representations might be used here to prepare the data for modeling.

**Model Development:**

The core of the system is Model Development, where the system uses both traditional machine learning and deep learning techniques. The flow branches into:

Traditional ML Models: Including Support Vector Machine (SVM), Naive Bayes, and Logistic Regression, which are classic models often used for text classification tasks.

Deep Learning Models: Using advanced architectures like LSTM (Long Short-Term Memory) Networks and BERT (Bidirectional Encoder Representations from Transformers), which are state-of-the-art models for handling complex natural language processing tasks.

**Model Evaluation:**

After model development, the system proceeds to Model Evaluation, where the performance of the models is tested and validated to ensure accuracy and reliability in sentiment classification.

**Application Development:**

Finally, the results of the analysis are integrated into the Application Development phase. This includes building both a Desktop App and a Mobile App, providing a user-friendly interface for sentiment analysis, real-time data monitoring, and trend visualization.

This diagram provides a clear representation of how data flows through the system, from data collection to application development, highlighting the key processes and models used to achieve sentiment analysis.

Data Flow Diagram (DFD)

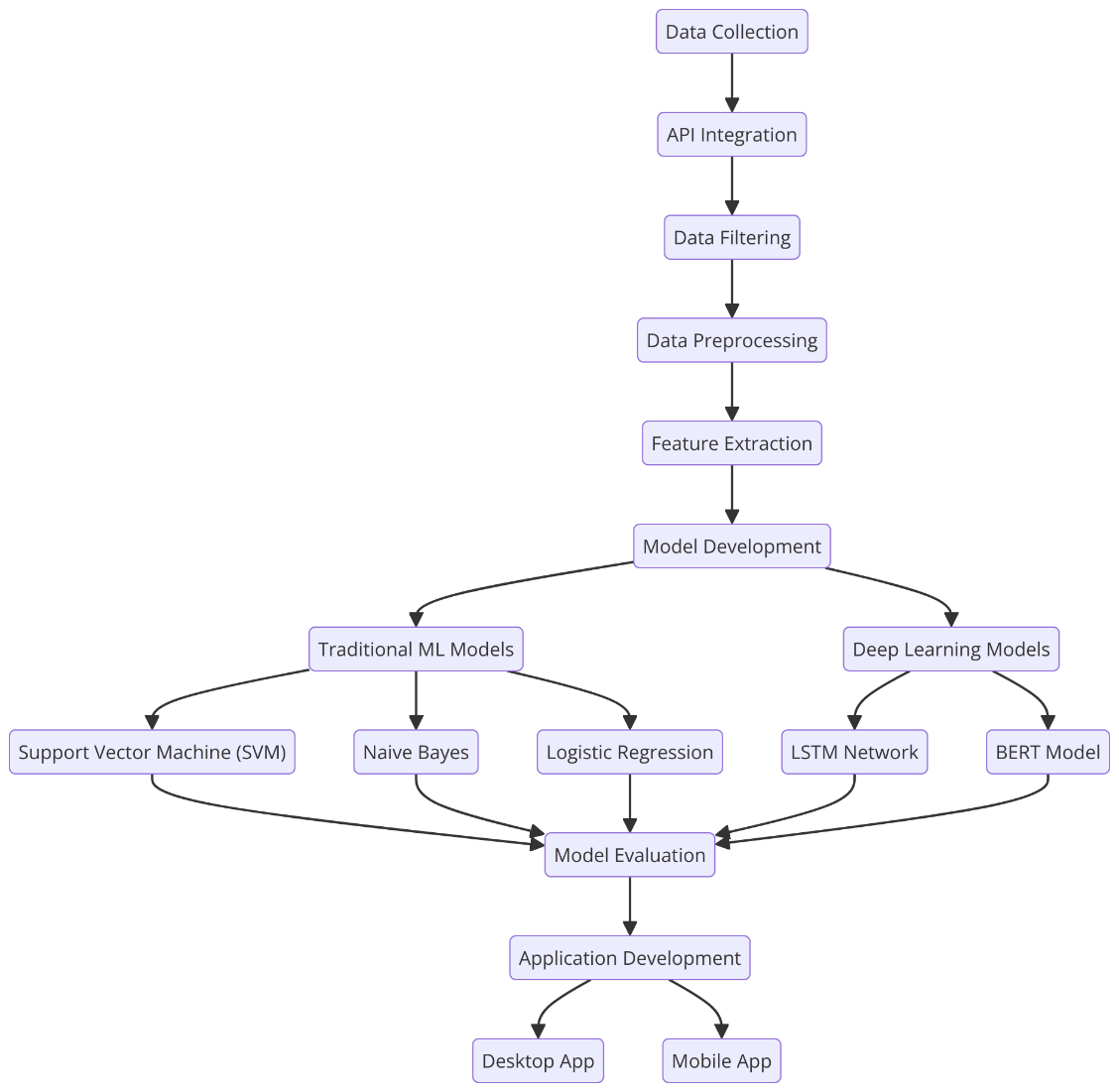


Figure 3: Proposed Study Flow Diagram

###### Problem Formulation

Automated Real-Time Sentiment Analysis of Twitter Data Using Machine Learning and Deep Learning Techniques

The problem is to design an automated system for real time sentiment analysis of Twitter data using a hybrid model of machine learning and deep learning. The goal is to classify the tweets sentiments (positive, negative, neutral) with noise data, mixed sentiments and context ambiguity.

Let represent the set of collected tweets:

For each tweet , the task is to predict its sentiment label , where:

: Positive sentiment

: Neutral sentiment

: Negative sentiment

The tweet undergoes preprocessing to extract feature vectors :

We define the sentiment classification model , parameterized by . The predicted sentiment is given by:

The objective is to minimize the classification error , defined as:

where is the cross-entropy loss function:

Objective Functions

Maximize Classification Accuracy:

Minimize Cross-Entropy Loss:

Minimize Misclassification Rate for Mixed Sentiments:

Notations

: Set of all tweets

: A single tweet

: Total number of tweets

: True sentiment label of tweet

: Predicted sentiment label of tweet

: Feature vector extracted from tweet

: Sentiment classification model

: Model parameters

: Loss function

: Indicator function

The formulated problem is to classify the sentiment of tweets in real time using a mixture of standard machine learning and state of the art deep learning models. Preprocessing the tweets stream to extract meaningful features is our input, which is a stream of tweets. The model integrates these features into a hybrid sentiment analysis model that outputs a prediction of the sentiment label (positive, neutral or negative) for each tweet.

The goal is to maximize classification errors and minimize errors, particularly with tweets that contain mixed or ambiguous sentiment. The problem formulation tackles noisy data, model scalability and a dual platform application (desktop and mobile) for real time analysis. The system optimizes the model parameters to provide accurate real time sentiment insight to support applications such as brand monitoring, political analysis and market research.

###### The Modules

**Data Collection Module:** It uses Twitter API to gather real time tweets based on keywords.

**Data Preprocessing Module**: It cleans and normalizes the tweet text for analysis.

**Feature Extraction Module:** It takes a text data and transforms it into numerical features based on word embeddings.

**Sentiment Classification Module:** It uses the machine learning and deep learning models for sentiment prediction.

**Visualization Module:** It shows sentiment trends using interactive charts and graphs.

**Application Interface Module:** It offers a user friendly desktop and mobile interface for data analysis.

Example Module (Sentiment Classification): In this module we use a hybrid approach of combining traditional (SVM, Naive Bayes) and deep learning (LSTM, BERT) models to predict sentiment of tweets and classify the tweet as positive, negative or neutral.

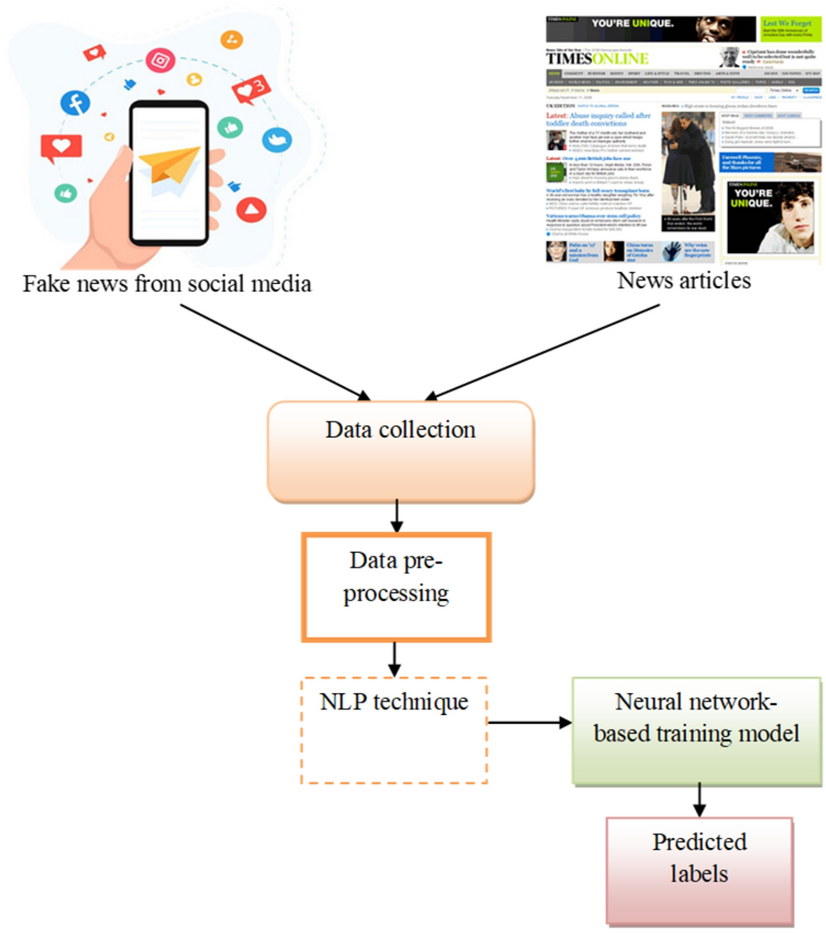


Figure 1 Fake news detection Mobile App

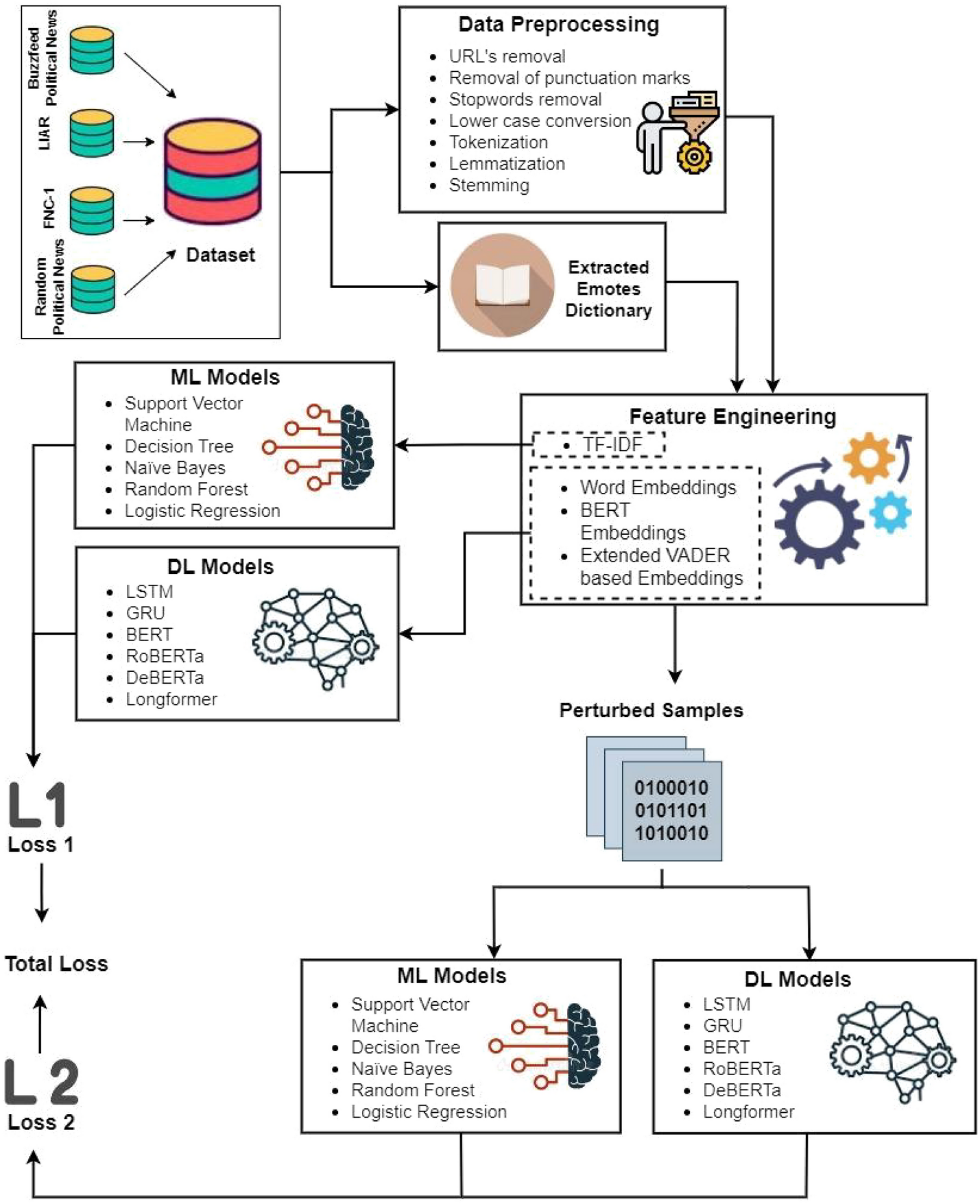


Figure 2 Fake news detection Web App

goals

###### Functional Requirements

The functional requirements of the sentiment analysis system are organized by key system features to ensure clarity and alignment with project goals

**Real-Time Data Collection** requires the system to fetch live Twitter data using the Twitter API based on user-defined keywords, hashtags, or filters.

**Data Preprocessing** involves tokenization, stemming, and noise removal to handle unstructured and informal text.

**Sentiment Analysis** necessitates the implementation of machine learning and deep learning models, such as SVM, LSTM, and BERT, to classify sentiments with high accuracy.

**Visualization Tools** are essential for presenting sentiment trends in real-time through interactive graphs, charts, and dashboards for easy interpretation by users.

**Dual-Platform Accessibility** mandates the development of both desktop and mobile applications with consistent, user-friendly interfaces.

**User Management** ensures secure login, personalized preferences, and saved analysis reports. These features collectively fulfill the system's purpose of delivering real-time, accurate, and actionable sentiment insights to its usersle.

Functional Requirement X

Itemize the specific functional requirements associated with each feature. These are the software capabilities that must be implemented for the user to carry out the feature‟s services or to perform a use case. Describe how the product should respond to anticipated error conditions and

to invalid inputs and actions. Uniquely label each functional requirement, as described earlier. You can create multiple attributes for each functional requirement, such as rationale, source, dependencies etc. The following template is required to write functional requirements.

**Table 6 Show the functional requirement template**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Identifier** | **Title** | **Requirement** | **Source** | **Rationale** | **Business Rule (if required)** | **Dependencies** | **Priority** |
| **FR-001** | Real-Time Data Collection | The system shall collect real-time Twitter data based on user-defined keywords, hashtags, or filters, with a response time of no more than 2 seconds. | Project Team | To provide up-to-date sentiment analysis based on the most recent trends. | Must adhere to Twitter's rate limits for API calls. | FR-002, FR-003 | High |
| **FR-002** | Data Preprocessing | The system shall preprocess collected Twitter data by tokenizing, stemming, and removing noise (e.g., special characters, stop words) within 5 seconds of collection. | Project Team | To ensure that the data is cleaned and ready for analysis, improving the accuracy of predictions. | N/A | FR-003, FR-004 | High |
| **FR-003** | Sentiment Analysis | The system shall apply machine learning (SVM, Naive Bayes) and deep learning models (LSTM, BERT) to classify sentiment within 10 seconds after preprocessing. | Project Team | To accurately classify sentiments from unstructured, informal Twitter text using advanced techniques. | Models must be trained on a sufficiently large dataset to ensure accuracy. | FR-004 | High |
| **FR-004** | Visualization Tools | The system shall present real-time sentiment trends using interactive graphs and charts within 5 seconds of analysis completion. | Project Team | To provide users with a clear, understandable way to interpret sentiment data in real time. | Must support multiple chart types, including bar, line, and pie charts. | N/A | Medium |
| **FR-005** | Dual-Platform Accessibility | The system shall provide a desktop and mobile application with consistent user interfaces, ensuring both versions have the same functionalities and capabilities. | Project Team | To ensure accessibility across different user devices and platforms. | N/A | FR-002, FR-003 | High |
| **FR-006** | User Management | The system shall provide a secure login feature for users, allowing them to save personalized settings and analysis reports for future access. | Project Team | To offer a personalized user experience and secure access to analysis data. | N/A | FR-005 | Medium |

###### External Interface Requirements

###### **User Interfaces**

###### Graphical User Interface (GUI):

###### The system must provide an intuitive, responsive graphical user interface (GUI) for users to interact with the system. The interface should be designed to cater to both desktop and mobile users, with easy navigation and clear visual cues.

###### The GUI must present features such as real-time data collection, sentiment classification, trend visualization, and customizable settings for filtering tweets and defining parameters for sentiment analysis.

###### Dashboard:

###### The user interface must feature an interactive dashboard that displays key sentiment metrics, including sentiment distribution (positive, negative, neutral), real-time sentiment trends, and emerging topics.

###### Users should be able to view and interact with various graphical visualizations like line graphs, pie charts, and heatmaps.

###### **Customization and Settings Panel:**

###### The system should provide users with the ability to customize settings through an easy-to-use settings panel, where they can set filters for keywords, hashtags, geographical locations, languages, and time frames for data collection.

###### **Mobile Application UI:**

###### For mobile users, the interface must be optimized for smaller screens, ensuring accessibility and ease of use. The mobile UI should support touch-based interactions, such as scrolling through sentiment trends and tapping for more detailed views.

###### Hardware Interfaces

######  Device Compatibility:

###### The system must be compatible with commonly used desktop devices, such as Windows PCs, macOS devices, and Linux systems. The desktop application should support the latest versions of web browsers for its web-based interface.

###### The mobile application must support Android and iOS devices, ensuring cross-platform compatibility. It should be optimized for modern smartphones and tablets.

###### Cloud Integration:

###### The system may rely on cloud platforms (e.g., AWS, Google Cloud, or Microsoft Azure) for data storage and real-time processing. Integration with cloud infrastructure ensures scalability and efficient handling of large datasets.

###### The cloud-based servers must be capable of supporting continuous tweet streaming, processing, and data storage while maintaining high performance and reliability.

###### Software Interfaces

###### Twitter API:

###### The system must integrate with the Twitter API to collect real-time tweets. This interface allows the system to send queries to Twitter's servers to retrieve data based on specified keywords, hashtags, or user handles.

###### The system should be capable of making authenticated API requests to retrieve tweets in real-time, ensuring compliance with Twitter’s usage policies.

###### Machine Learning and Deep Learning Frameworks:

###### The sentiment analysis system must interact with various machine learning and deep learning frameworks, including but not limited to:

###### **Scikit-learn:** For implementing classic machine learning models such as Support Vector Machine (SVM) and Naive Bayes.

###### TensorFlow or PyTorch:For training and deploying deep learning models like Long Short-Term Memory (LSTM) and Bidirectional Encoder Representations from Transformers (BERT).

###### **SpaCy and NLTK:** For natural language processing (NLP) tasks like tokenization, stemming, lemmatization, and named entity recognition (NER).

###### Database Management System:

###### The system must integrate with a database management system (DBMS) to store collected tweets, their processed data, sentiment labels, and visualization results. Supported DBMSs include **MySQL, MongoDB, or PostgreSQL**, depending on the complexity and scale of the system.

###### The database should allow for efficient querying, storing, and retrieving large volumes of tweet data and analysis results.

###### Cloud Storage Services:

###### If using cloud platforms, the system must interact with cloud storage services like Amazon S3 or Google Cloud Storage for storing large datasets, processed results, and models. The system should ensure secure access and data redundancy in cloud storage.

###### Communications Interfaces

###### RESTful API:

###### The system must expose a RESTful API that allows external applications and services to interact with the sentiment analysis system. The API will be used for retrieving sentiment analysis results, performing data collection, and customizing system settings programmatically.

###### The API should support key operations such as:

###### Retrieving sentiment classification results for specific tweets.

###### Requesting real-time sentiment trends and historical sentiment data.

###### Adding or removing keywords and hashtags for real-time data collection.

###### Non Functional Requirements

For the sentiment analysis system, several non-functional requirements are defined to ensure the system’s quality, efficiency, and usability.

**Usability:**  
The system shall provide an intuitive and user-friendly interface across both desktop and mobile platforms, ensuring ease of use for individuals with varying technical backgrounds. The interface should be designed to minimize errors, and should allow users to perform sentiment analysis, visualize trends, and access historical reports with minimal effort. The user interface shall provide clear instructions, tooltips, and contextual help to guide users in navigating through the features. Additionally, the system shall allow users to save and retrieve previous analyses with a single interaction to enhance usability and reduce repetitive tasks.

**Performance:**  
The system must provide real-time sentiment analysis with a maximum response time of 10 seconds for data collection, preprocessing, and sentiment classification. Sentiment trends shall be visualized in real time within 5 seconds after the analysis is completed. Given the volume of Twitter data, the system must handle up to 500,000 tweets per hour without significant performance degradation. Furthermore, the system shall scale efficiently to handle spikes in data volume, such as during major events or social movements, ensuring continuous operation without downtime or performance bottlenecks.

**Scalability:**  
The system must be scalable to accommodate future growth in data volume and the number of users. It should support additional features like multiple user accounts, region-based sentiment analysis, and additional data sources without requiring a major redesign. The backend infrastructure must be able to handle increasing computational loads as more complex models and larger datasets are integrated.

**Availability and Reliability:**  
The system shall ensure high availability, aiming for 99.9% uptime, and it should have reliable backup mechanisms in place to prevent data loss during failures. Data collection and analysis processes should be fault-tolerant, meaning the system must recover gracefully from failures such as network disruptions or server crashes without data corruption or significant downtime.

**Security:**  
The system shall implement robust security measures to protect user data and sensitive analysis results. It shall use encryption for data storage and transmission, ensuring secure access to user profiles, saved reports, and analysis results. User authentication will be mandatory for accessing personalized settings, and the system will incorporate role-based access control to restrict unauthorized access to administrative features.

**Other Requirements**

This section includes additional requirements that are not covered elsewhere in the SRS but are essential to the successful development, deployment, and operation of the system.

**8.1 Database Requirements**

**Type of Database:** The system will require a relational database for storing user profiles, prediction history, and sentiment analysis results. The preferred database is MySQL or PostgreSQL.

**Data Storage:** The system will store all processed tweets, their features, and sentiment analysis results in the database.

**Data Backup:** Daily backups of the database will be scheduled to ensure the protection of critical data. Backup files will be stored securely in the cloud.

**Data Retrieval:** The database must support fast retrieval of historical prediction data for users to view their past analysis results.

**8.2 Legal Requirements**

**Data Privacy Compliance:** The system must comply with data privacy laws and regulations, such as GDPR or CCPA, ensuring that user data is securely handled and not shared without consent.

**Copyright:** All data extracted from Twitter should be processed in compliance with Twitter's terms of service, ensuring that only publicly available information is used for analysis.

**Licensing:** Any third-party libraries or tools used within the project must comply with their respective licenses, and any commercial usage must be carefully considered to avoid legal conflicts.

**8.3 External Interface Requirements**

**Third-Party APIs**: The system must integrate with Twitter’s official API to collect real-time tweet data. Additionally, any other external services for language processing, sentiment analysis, or data visualization must be specified and properly licensed.

**User Authentication:** The system will integrate with OAuth for secure user authentication to access their Twitter account data.

**8.4 Internationalization Requirements**

**Language Support:** The system will be designed to support English tweets primarily, but the potential for future expansion to handle multiple languages should be considered. Language models or multilingual sentiment analysis techniques may be added for broader market usage.

**Date and Time Formatting:** The system must support various date and time formats, based on the user's region, to ensure proper interpretation of time-sensitive data.

**8.5 System Maintenance and Updates**

**Automatic Updates:** The system should include automatic update functionality for both desktop and mobile applications to ensure users receive the latest improvements, bug fixes, and features.

**Documentation and Training:** End-user documentation, including tutorials and FAQs, should be available for both desktop and mobile users. Additionally, a training module for businesses and analysts will be provided to help them fully utilize the system’s features.

**Appendix A: Glossary**

**1. Sentiment Analysis**  
The process of determining the emotional tone (positive, negative, or neutral) expressed in a piece of text. It helps in understanding public opinion or customer sentiment.

**2. Twitter API**  
A set of programming interfaces provided by Twitter, allowing developers to access and interact with Twitter’s data, including retrieving tweets, posting messages, and monitoring trends.

**3. Machine Learning (ML)**  
A subset of artificial intelligence (AI) that involves training algorithms to learn patterns from data and make predictions or decisions based on that data.

**4. Deep Learning**  
A subset of machine learning that uses neural networks with many layers (hence "deep") to learn from large amounts of data. It is particularly useful for tasks like speech recognition, image analysis, and sentiment analysis.

**5. Support Vector Machine (SVM)**  
A supervised machine learning algorithm used for classification tasks. SVM aims to find the hyperplane that best separates different classes in the data.

**6. Naive Bayes**  
A simple probabilistic classifier based on Bayes' Theorem, assuming independence between features. It is often used for text classification tasks like spam detection and sentiment analysis.

**7. Long Short-Term Memory (LSTM)**  
A type of recurrent neural network (RNN) that is particularly good at learning from sequential data and retaining information over time. LSTM networks are used for tasks like speech recognition and sentiment analysis of long texts.

**8. Bidirectional Encoder Representations from Transformers (BERT)**  
A pre-trained transformer model for natural language processing (NLP). BERT understands the context of a word in relation to all other words in a sentence, making it highly effective for tasks like sentiment analysis.

**9. Tokenization**  
The process of breaking text into smaller units called "tokens," which can be words, phrases, or even characters. Tokenization is the first step in text preprocessing.

**10. Stemming**  
The process of reducing a word to its root form. For example, "running" becomes "run," and "better" becomes "good."

**11. Feature Extraction**  
The process of transforming raw data (such as text) into a set of numerical features that can be used in machine learning algorithms. Common methods include TF-IDF and word embeddings.

**12. TF-IDF (Term Frequency-Inverse Document Frequency)**  
A statistical method used to evaluate the importance of a word in a document relative to a collection of documents (corpus). It helps identify keywords in text.

**13. Word Embedding**  
A technique for representing words as vectors in a continuous vector space. Popular models for word embeddings include Word2Vec, GloVe, and FastText.

**14. Real-Time Data**  
Data that is collected and processed instantly or with minimal delay. In the context of this system, it refers to Twitter data that is analyzed as it is posted.

**15. API (Application Programming Interface)**  
A set of tools and protocols that allows different software systems to communicate with each other. The Twitter API, for example, allows access to Twitter's data programmatically.

**16. RESTful API**  
A type of web API that adheres to REST (Representational State Transfer) principles, which include stateless communication and the use of HTTP methods like GET, POST, PUT, and DELETE.

**17. Cloud Platforms**  
Online services that provide on-demand computing resources such as data storage, processing power, and machine learning tools. Examples include AWS (Amazon Web Services), Google Cloud, and Microsoft Azure.

**18. Database**  
An organized collection of structured data. In this system, databases such as MySQL or MongoDB are used to store tweets and sentiment analysis results.

**19. Noisy Data**  
Data that contains irrelevant or extraneous information, making it difficult to analyze. In social media, this can include slang, emojis, abbreviations, and other informal expressions.

**20. Dual-Platform System**  
A software system designed to operate on both desktop and mobile platforms, offering flexibility and broader accessibility for users.

**21. API Authentication**  
The process of verifying the identity of an API user before granting access to the data. Twitter API requires OAuth for secure access.

###### References

Agarwal, A., Mishra, S., & Ahmad, S. (2023). Fake News Detection Using Machine Learning. Lecture Notes in Electrical Engineering, 1068 LNEE(3), 51–59. https://doi.org/10.1007/978-981-99-5358-5\_4

Ahmed, H. (2017). Detecting Opinion Spam and Fake News Using N-gram Analysis and Semantic Similarity. University of Ahram Canadian, 74.

Al-tai, M. H., Nema, B. M., & Al-sherbaz, A. (2023). Deep Learning for Fake News Detection : Literature Review. 34(2).

Alazab, M., Awajan, A., Alazab, A., Khreisat, A., & Saadeh, R. (2022). FAKE-NEWS DETECTION SYSTEM USING MACHINE- LEARNING ALGORITHMS FOR ARABIC-LANGUAGE CONTENT. 100(16), 5056–5069.

Berrondo-otermin, M., & Sarasa-cabezuelo, A. (2023). Application of Artificial Intelligence Techniques to Detect Fake News : A Review.

Chioma, D., John, G., Chioma Aguh, D., Obianke John, G., & Sarah Akanji, O. (2016). a Research on Detection of Fake News on Social Media Using Machine Learning Algorithms. https://www.researchgate.net/publication/366319728

Gurunathan, P. (2023). OPTIMAL FEATURES BASED FAKE NEWS DETECTION MODEL USING MACHINE. December 2022. https://doi.org/10.14704/nq.2022.20.10.NQ551051

Hosseinimotlagh, S., & Papalexakis, E. E. (2018). Unsupervised content-based identification of fake news articles with tensor decomposition ensembles. Proceedings of the WSDM MIS2: Misinformation and Misbehavior Mining on the Web Workshop, 1–8. https://doi.org/10.475/123

Jaybhaye, S. M., Badade, V., Dodke, A., & Holkar, A. (2023). Fake News Detection using LSTM based deep learning approach. 03005, 1–12.

John, A. (2022). Fake News Detection Using Unsupervised and Deep Learning Algorithms. February. https://doi.org/10.46610/JOCSES.2021.v07i02.004

Kh, S., Juzaiddin, M., Aziz, A., & Ridzwan, M. (2023). 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 fu ... Heliyon Review article A review of fake news detection approaches : Heliyon, 9(10), e20382. https://doi.org/10.1016/j.heliyon.2023.e20382

Kumar, S., & Arora, B. (2021). A Review of Fake News Detection Using Machine Learning Techniques. Proceedings of the 2nd International Conference on Electronics and Sustainable Communication Systems, ICESC 2021, 9(9), 1820–1827. https://doi.org/10.1109/ICESC51422.2021.9532796

Mertoğlu, U., & Genç, B. (2020). Automated fake news detection in the age of digital libraries. Information Technology and Libraries, 39(4). https://doi.org/10.6017/ITAL.V39I4.12483

Monti, F., Frasca, F., Eynard, D., Mannion, D., & Bronstein, M. M. (2019). Fake News Detection on Social Media using Geometric Deep Learning. 1–15. http://arxiv.org/abs/1902.06673

Noida, G., Noida, G., Noida, G., & Noida, G. (2023). FAKE NEWS DETECTION USING MACHINE. 11(4), 59–68.

Pandey, M., & Kumar, M. (2022). Fake News Detection using Deep. December.

Shahid, W., Li, Y., Staples, D., Amin, G., Hakak, S., & Ghorbani, A. (2022). Are You a Cyborg, Bot or Human?-A Survey on Detecting Fake News Spreaders. IEEE Access, 10, 27069–27083. https://doi.org/10.1109/ACCESS.2022.3157724

Shalini, A. K., Saxena, S., & Kumar, B. S. (2023). Designing A Model for Fake News Detection in Social Media Using Machine Learning Techniques. International Journal of Intelligent Systems and Applications in Engineering, 11(2s), 218–226.

Shu, K., Sliva, A., Wang, S., Tang, J., & Liu, H. (2017). Fake News Detection on Social Media. ACM SIGKDD Explorations Newsletter, 19(1), 22–36. https://doi.org/10.1145/3137597.3137600

Usha, M. (2023). “ Fake News Detection Using Machine learning .” 12(2), 192–199. https://doi.org/10.17148/IJARCCE.2023.12242

Valerian Chinedum, A., Bethran Chibuike, A., & A. Onyekachi, A. (2023). Detecting Online Fake News Using Machine Learning and Topological Data Analysis. International Journal of Engineering Applied Sciences and Technology, 7(10), 42–50. https://doi.org/10.33564/ijeast.2023.v07i10.006

Vardhan, K. V., Josephine, B. M., & Rao, K. V. S. N. R. (2022). Fake News Detection in Social Media Using Supervised Learning Techniques. International Conference on Sustainable Computing and Data Communication Systems, ICSCDS 2022 - Proceedings, 9(6), 695–698. https://doi.org/10.1109/ICSCDS53736.2022.9760961.