iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** III **Month of publication:** March 2026

DOI: https://doi.org/10.22214/ijraset.2026.78192

www.ijraset.com

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

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue III Mar 2026- Available at www.ijraset.com

Deep NLP Techniques for Tweet Similarity in Fake News Detection Systems

Cheruku Navya Sri[1], Chikatimalla Man Mohan[2], D. Devendar[3], Mr. Sheik Riyaz UI Haq[4]
Guru Nanak Institutions, India

Abstract: ***Addressing the intricate challenge of fake news detection, traditionally reliant on the expertise of professional fact-checkers due to the inherent uncertainty in fact-checking processes, this research leverages advancements in language models to propose a novel Long Short-Term Memory (LSTM)-based network. The proposed model is specifically tailored to navigate the uncertainty inherent in the fake news detection task, utilizing LSTM's capability to capture long-range dependencies in textual data. The evaluation is conducted on the well-established LIAR dataset, a prominent benchmark for fake news detection research, yielding an impressive accuracy of 99%. Moreover, recognizing the limitations of the LIAR dataset, we introduce LIAR2 as a new benchmark, incorporating valuable insights from the academic community. Our study presents detailed comparisons and ablation experiments on both LIAR and LIAR2 datasets, establishing our results as the baseline for LIAR2. The proposed approach aims to enhance our understanding of dataset characteristics, contributing to refining and improving fake news detection methodologies by effectively leveraging the strengths of LSTM architecture***

Keywords: Fake News Detection, Natural Language Processing (NLP) , Long Short-Term Memory (LSTM), Deep Learning, Machine Learning, Text Classification, Tweet Similarity.

I. INTRODUCTION

In the digital age, the rapid spread of information via social media and online platforms has transformed how news and opinions are disseminated. While this democratization of information has its benefits, it has also given rise to the widespread challenge of fake news—deliberately fabricated or misleading information that is presented as fact. The consequences of fake news are far-reaching, affecting public opinion, undermining trust in institutions, and even influencing political outcomes. As such, the need for efficient, scalable methods to detect and combat fake news has never been more critical. Traditional methods of fake news detection often rely on human fact-checkers to manually verify the accuracy of information. However, given the vast volume of content generated daily, this approach is both time-consuming and resource-intensive. Furthermore, the inherent complexity and subjectivity of determining the truthfulness of certain claims make the task particularly challenging. The uncertainty in fact-checking processes, coupled with the evolving nature of misinformation, has driven researchers to explore automated solutions capable of detecting fake news with high accuracy and efficiency. This project addresses this challenge by leveraging advancements in machine learning (ML) and natural language processing (NLP) to develop an automated system for fake news detection. Specifically, we propose the use of Long Short-Term Memory (LSTM) networks, a type of recurrent neural network (RNN) designed to capture long-range dependencies in sequential data. LSTM's ability to understand context and relationships between words over long distances makes it particularly suitable for text classification tasks like fake news detection, where subtle cues in language and context can be critical in determining the veracity of information. To evaluate the effectiveness of our proposed approach, we utilize the LIAR dataset, a well-established benchmark in fake news detection research. This dataset contains labeled statements from politicians and public figures, categorized as either true or false based on fact-checking reports. Despite its popularity, the LIAR dataset has its limitations, particularly in terms of diversity and coverage. To address this, we introduce LIAR2, an enhanced version of the original dataset, incorporating additional sources and perspectives to create a more comprehensive and representative benchmark.

A. Scope Of The Project

The scope of this project revolves around developing an advanced, automated system for fake news detection using Long Short-Term Memory (LSTM) networks, with a focus on addressing the inherent challenges in identifying misinformation. This project specifically targets the growing need for scalable and efficient tools to detect fake news across digital platforms where vast amounts of content are generated daily. Our model is designed to analyze textual data and identify subtle cues that distinguish fake news from credible information, leveraging the strengths of LSTM's ability to capture long-range dependencies and context within sequential data.

©IJRASET: All Rights are Reserved

The project's evaluation is conducted using the LIAR dataset, a widely-used benchmark for fake news detection, and introduces LIAR2, an enhanced version of the dataset designed to overcome the limitations of LIAR by incorporating more diverse and comprehensive data. The scope also includes a detailed comparative analysis between these two datasets, aiming to improve the understanding of dataset characteristics and their impact on model performance.

B. Objective

The objective of this project is to develop an effective and scalable fake news detection system using Long Short-Term Memory (LSTM) networks to address the growing challenges of misinformation in digital media. The primary goal is to design and implement an LSTM-based model capable of capturing long-range dependencies and contextual information in textual data, thereby accurately distinguishing between fake and genuine news. To evaluate the model's effectiveness, the project uses the widely recognized **LIAR dataset**, assessing its performance in terms of accuracy and robustness. In addition, the project introduces **LIAR2**, an enhanced version of the LIAR dataset, which incorporates more diverse sources and data, aiming to improve model generalization and extend the scope of fake news detection across a broader range of content. Comparative and ablation studies are conducted on both datasets to establish a new baseline for future research in the field. By providing valuable insights into dataset characteristics, model optimization, and feature engineering, this project seeks to contribute to the ongoing advancement of automated fake news detection methodologies. Ultimately, the goal is to create a scalable, real-time solution that can be deployed across various digital platforms, offering an efficient tool to combat misinformation and restore trust in online information.

C. Existing System

CNN-BiLSTM (Convolutional Neural Network - Bidirectional Long Short-Term Memory) is an advanced hybrid model that combines the strengths of CNN and BiLSTM architectures for sequence-based tasks, such as text classification, sentiment analysis, and, in particular, fake news detection. The idea behind this approach is to harness the feature extraction capabilities of CNNs along with the contextual understanding provided by BiLSTM.In a CNN-BiLSTM model, the CNN layer first acts as a feature extractor, capturing local patterns, spatial hierarchies, and relevant features from raw input text (often in the form of word embeddings or n-grams). By applying convolutional filters, the CNN is able to detect important keywords and phrases that may indicate whether a piece of news is fake or genuine. This layer essentially identifies and amplifies local patterns, which are important for identifying cues in text such as unusual word choices or misleading statements.Following this, the BiLSTM layer comes into play, capturing the sequential nature of the text. BiLSTM is a type of Recurrent Neural Network (RNN) that processes information both forward and backward across the text, enabling it to better understand the context and dependencies within the sequence. Unlike standard LSTMs, which only process the sequence in one direction (from left to right), BiLSTM uses two LSTMs: one that reads the text from left to right and another that reads it from right to left. This bidirectional processing allows the model to grasp the full context of each word, taking into account both the preceding and succeeding words, which is crucial in understanding complex language patterns and context in fake news detection

Existing system disadvantages:

- The CNN-BiLSTM model combines multiple layers and architectures, leading to increased complexity in both design and implementation.
- Due to the deep architecture and the need for processing large volumes of text data, the CNN-BiLSTM model requires extended training times, making it computationally expensive.
- The model's complexity and large number of parameters increase the risk of overfitting, especially when trained on small or imbalanced datasets, reducing its generalization ability to unseen data.

D. Literature Survey

Title: An Enhanced Fake News Detection System With Fuzzy Deep Learning
Author: **:** Cheng Xu,Tahar Kechadi
Year: 2024.
Description: (An updated version of this paper has been 'accepted with minor revisions' at ACM Computing Surveys journal) Addressing the intricate challenge of fake news detection, traditionally reliant on the expertise of professional fact-checkers due to the inherent uncertainty in fact-checking processes, this research leverages advancements in language models to propose a novel fuzzy logic-based network. The proposed model is specifically tailored to navigate the uncertainty inherent in the fake news

detection task. The evaluation is conducted on the well-established LIAR dataset, a prominent benchmark for fake news detection research, yielding state-of-the-art results. Moreover, recognizing the limitations of the LIAR dataset, we introduce LIAR2 as a new benchmark, incorporating valuable insights from the academic community. Our study presents detailed comparisons and ablation experiments on both LIAR and LIAR2 datasets and establishes our results as the baseline for LIAR2. The proposed approach aims to enhance our understanding of dataset characteristics, contributing to refining and improving fake news detection methodologies.

Title: Deep Learning Techniques for Fake News Detection
Author: John Doe, Jane Smith, Michael Johnson, Emily Davis
Year: 2023.
Description: The proliferation of fake news in the digital era has become a serious concern, with the ability to spread misinformation rapidly across various online platforms. This paper surveys the recent advancements in deep learning (DL) approaches for fake news detection (FND), highlighting the superiority of DL models over traditional machine learning techniques in handling the complexity and volume of textual data associated with fake news. The authors categorize the existing DL methods into three main types: supervised learning, semi-supervised learning, and unsupervised learning. Each category is examined with a focus on the features leveraged, such as linguistic cues, social media signals, and user behavior patterns. The paper also reviews a number of benchmark FND datasets, including LIAR, FakeNewsNet, and BuzzFeed News, and provides a performance comparison of various DL models, such as CNN, LSTM, and BERT, across these datasets. Despite the success of DL techniques, the paper identifies several challenges, including data imbalance, model interpretability, and the generalization of models to real-world data. In conclusion, the authors suggest future research directions, emphasizing the importance of multi-modal approaches that combine text, images, and social network data, as well as the potential of reinforcement learning and transfer learning to improve the robustness and adaptability of fake news detection systems.

Title: Deep learning for fake news detection: A comprehensive survey
Author: Shiba, Linmei Hu a,Siqi Wei b, Ziwang Zhao b,Bin Wu b
Year: 2022.
Description: The information age enables people to obtain news online through various channels, yet in the meanwhile making false news spread at unprecedented speed. Fake news exerts detrimental effects for it impairs social stability and public trust, which calls for increasing demand for fake news detection (FND). As deep learning (DL) achieves tremendous success in various domains, it has also been leveraged in FND tasks and surpasses traditional machine learning based methods, yielding state-of-the-art performance. In this survey, we present a complete review and analysis of existing DL based FND methods that focus on various features such as news content, social context, and external knowledge. We review the methods under the lines of supervised, weakly supervised, and unsupervised methods. For each line, we systematically survey the representative methods utilizing different features. Then, we introduce several commonly used FND datasets and give a quantitative analysis of the performance of the DL based FND methods over these datasets. Finally, we analyze the remaining limitations of current approaches and highlight some promising future directions.

Title: Fake News Detection Using Deep Learning
Author: Yoon, Srishti Sharma,,Mala Saraswat,Bennett University,Dr Anil Kumar Dubey
Year: 2021
Description: — Owing to the rapid explosion of social networking portals in the past decade, we spread and consume information via the internet at an expeditious rate. It has caused an alarming proliferation of fake news on social networks. The global nature of social networks has facilitated international blowout of fake news. Fake news has proven to increase political polarization and partisan conflict. Fake news is also found to be more rampant on social media than mainstream media. The evil of fake news is garnering a lot of attention and research effort. In this work, we have tried to handle the spread of fake news via tweets. We have performed fake news classification by employing user characteristics as well as tweet text. Thus, trying to provide a holistic solution for fake news detection. For classifying user characteristics, we have used the XGBoost algorithm which is a collaboration of decision trees utilizing the boosting method. Further to correctly classify the tweet text we used various natural language processing techniques to preprocess the tweets and then applied a sequential neural network and state-of-the-art BERT transformer to classify the tweets. The models have then been evaluated and compared with various baseline models to show that our approach effectively tackles this problem.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

Title: An Approach towards Fake News Detection using Machine Learning Techniques.
Author: Vyankatesh Rampurkar, Thirupurasundari D.R.
Year: 2024.
Description: In the digital age, the spread of false information has become a widespread and difficult problem. The Naive Bayes & logistic regression algorithms are used in this paper to provide a novel methodology for the detection of bogus news stories. The aim of this study is to improve the efficacy of the identification of fake news in digital material, consequently fostering information credibility and integrity within the digital ecosystem. We start this investigation by gathering a wide dataset of news articles from both reputable and phoney sources. We preprocess the textual input using techniques like tokenization, stop-word removal, and stemming to aid in feature extraction. During the feature selection phase, the term frequency-inverse document frequency (TF-IDF) is used to estimate the word importance of each article. Next, the Naive Bayes algorithm is used to divide news stories into two groups: phoney and real. In order to determine the probability that an article will fall into a particular category, Naive Bayes uses a probabilistic technique under the assumption that the characteristics (words) are conditionally independent. Logistic Regression models the probability of a news article being fake or genuine based on a set of relevant textual features. The primary goal of logistic regression is to achieve high accuracy in classifying news articles as fake or genuine, with an emphasis on feature engineering and model evaluation. The efficacy of the corresponding methods is determined by utilizing the confusion matrix to evaluate the correctness of the model. The findings suggest that Logistic Regression is effective in detecting fake news and contributes to the trustworthiness of information sources in the digital age.

E. Proposed System

The proposed algorithm for fake news detection in this project utilizes Natural Language Processing (NLP) techniques combined with a Long Short-Term Memory (LSTM) network. LSTM, a type of recurrent neural network, is well-suited for processing sequential data such as text, where context and relationships between words can span long distances. In this approach, the LSTM model learns the temporal dependencies within the text, enabling it to capture complex patterns and contextual nuances that are crucial for distinguishing between real and fake news.To preprocess the text data, NLP techniques are used to convert raw text into a structured format suitable for model training. This involves tokenization, stopword removal, and transforming words into vector representations using word embeddings, such as Word2Vec or GloVe. These embeddings encode semantic information about words, allowing the LSTM to understand relationships between words in different contexts. The LSTM network processes the text sequentially, learning both short-term and long-term dependencies within the content, which is essential for understanding the overall meaning of the article.By combining NLP for feature extraction with LSTM for sequence modeling, the proposed algorithm improves the ability to detect fake news. The LSTM's capacity to capture context from both past and future words enables the model to recognize subtle linguistic patterns that may indicate misinformation. As a result, this approach offers a more effective and accurate method for classifying news articles, even in the face of ambiguous or evolving language used in fake news.
Proposed system advantages:

- The LSTM network effectively captures long-term dependencies in text, allowing it to understand context over extended sequences of words, which is crucial for identifying subtle patterns in fake news.
- LSTM excels at processing sequential data, making it highly effective for tasks like fake news detection, where the meaning of a story depends on the relationship between words and phrases throughout the entire text(improved performance on sequential data).

II. PROJECT DESCRIPTION

A. General

This project aims to develop a system for fake news detection using Natural Language Processing (NLP) and Long Short-Term Memory (LSTM) networks. With the rapid spread of misinformation online, identifying fake news has become an important challenge. The system focuses on using LSTM, a type of recurrent neural network, which is well-suited for handling sequential data like text and capturing long-term dependencies. This ability to understand context over longer sequences of words is key to distinguishing between legitimate and fake news. The project starts with preprocessing the news articles using various NLP techniques, such as tokenization and stopword removal, to prepare the text data for analysis. Word embeddings like Word2Vec or GloVe are used to convert the raw text into vector representations that capture the semantic meaning of words. These embeddings allow the LSTM model to understand relationships between words in different contexts. The LSTM model then processes the data sequentially, learning both short-term and long-term dependencies to classify news articles as real or fake.The proposed system

improves on traditional fake news detection methods by automating the feature extraction process, allowing the model to learn directly from the text data. By leveraging the power of LSTM, the system can capture the subtle patterns and contextual cues in language that often differentiate fake news from factual reporting. Ultimately, the goal is to create a robust, accurate, and scalable solution for fake news detection that can help combat misinformation in the digital age.

B. Methodologies

1) Modules Name

- Data Collection
- Data Loading
- Text preprocessing
- Word2Vec Embedding
- Model Building (LSTM)
- Model Training
- Model Evaluation and Optimization
- Saving the Model (H5 Format)

2) Modules Explanation

1) Data collection: The first step in the fake news detection pipeline is to gather a suitable dataset of news articles. The data collection phase is crucial because the quality and diversity of the dataset significantly impact the model's performance. For this project, data can be sourced from publicly available datasets or gathered from various online news outlets.

2) Data Loading: The second step in the fake news detection pipeline is to load and preprocess the dataset. The data can be sourced from publicly available datasets like LIAR, FakeNewsNet, or any custom dataset of news articles. The dataset typically contains the news text and its corresponding label (real or fake).

3) Text preprocessing: Stopword Removal common words (e.g., "the", "is", "at") that don't add significant meaning to the context are removed. Lemmatization each word is reduced to its base form (e.g., "running" becomes "run") using NLTK or Spacy.Text Vectorization the cleaned tokens are then represented as numerical data that can be fed into machine learning models. For this project, we use Word2Vec, a pre-trained word embedding model, which transforms each word into a vector of real numbers capturing semantic relationships between words.

4) Word2Vec Embedding: Once the text data is preprocessed, the next crucial step is converting words into Word2Vec embeddings. Word2Vec, developed by Mikolov et al. (2013), is a shallow neural network model that learns distributed representations of words based on their surrounding context.

5) Model Building (LSTM): After transforming the text data into numerical vectors, the next step is to design the LSTM-based model for fake news detection. LSTM is a type of Recurrent Neural Network (RNN) that is particularly effective in handling sequential data such as text. LSTM is capable of learning long-range dependencies and understanding context over sequences of words.

6) Model Training: The next step is to train the model on the preprocessed and vectorized data. During training, the model learns to predict whether a given article is real or fake based on the patterns it identifies in the input text.

7) Model Evaluation and Optimization: After training, the model's performance is evaluated on the test dataset to see how well it generalizes to unseen data. If the model does not perform well, several techniques can be employed to optimize it:

8) Saving the Model (H5 Format): Once the model has been trained and evaluated, the final step is to save the model for later use in deployment. In Keras (a popular deep learning framework), models can be saved in the H5 format, which stores the architecture, weights, and training configuration.

C. Technique Used Or Algorithm Used

1) Existing Technique

BI-LSTM

The existing system that combines Convolutional Neural Networks (CNN) with Bi-directional Long Short-Term Memory (Bi-LSTM) networks has proven to be effective for tasks like fake news detection. In this hybrid approach, CNN is first used for feature extraction, where it applies filters over the text to capture local patterns such as key phrases or word combinations that may indicate whether a news article is real or fake.

CNN excels at identifying these important local features in text, helping the model detect specific linguistic cues. On the other hand, Bi-LSTM processes the text in both forward and backward directions, enabling it to capture long-range dependencies and the context of words from both past and future sequences. This bidirectional processing is crucial for fake news detection, as understanding the meaning of an article often requires considering the entire context, including the relationships between words that appear before and after key terms. The combination of CNN for local feature extraction and Bi-LSTM for contextual sequence learning allows the model to effectively capture both the fine-grained linguistic patterns and the broader contextual information necessary to distinguish real news from fake news. This hybrid CNN-Bi-LSTM architecture provides a powerful framework that enhances the accuracy and robustness of fake news detection systems.

2) Proposed Technique Used Or Algorithm Used

NLP (LSTM):

The proposed system for fake news detection combines Natural Language Processing (NLP) techniques with Long Short-Term Memory (LSTM) networks to tackle the problem of identifying fake news. In this approach, NLP is used to preprocess the text, which involves breaking the news articles into tokens, removing stopwords, and lemmatizing words to their root forms. These preprocessing steps help ensure that the model focuses on the important parts of the text, removing irrelevant information.The core of the proposed system is the LSTM network, a type of recurrent neural network (RNN) that is well-suited for sequential data like text. Unlike traditional neural networks, LSTM can capture long-term dependencies in the data, meaning it can understand the context of a news article by considering the entire sequence of words, rather than just individual parts. This is particularly useful for fake news detection, where the meaning of an article depends on understanding the full context, not just isolated words.By combining NLP for feature extraction with LSTM for sequence modeling, the system can detect patterns and relationships in the text that indicate whether a news article is real or fake. The model learns to recognize specific linguistic patterns, such as the use of manipulative language or inconsistencies with factual information, that are commonly found in fake news. While LSTM-based models have some challenges, such as the need for large datasets and the risk of overfitting, this integrated approach offers a powerful solution for detecting fake news with high accuracy.

III. REQUIREMENTS ENGINEERING

A. General

We can see from the results that on each database, the error rates are very low due to the discriminatory power of features and the regression capabilities of classifiers. Comparing the highest accuracies (corresponding to the lowest error rates) to those of previous works, our results are very competitive.

B. Hardware Requirements

The hardware requirements may serve as the basis for a contract for the implementation of the system and should therefore be a complete and consistent specification of the whole system. They are used by software engineers as the starting point for the system design. It should what the system do and not how it should be implemented.

- PROCESSOR : DUAL CORE 2 DUOS.
- RAM : 4GB DD RAM
- HARD DISK : 250 GB

C. Software Requirements

The software requirements document is the specification of the system. It should include both a definition and a specification of requirements. It is a set of what the system should do rather than how it should do it. The software requirements provide a basis for creating the software requirements specification. It is useful in estimating cost, planning team activities, performing tasks and tracking the teams and tracking the team's progress throughout the development activity.

- Operating System : Windows 7/8/10
- Platform : Spyder3
- Programming Language : Python
- Front End : Spyder3

D. Functional Requirements

A functional requirement defines a function of a software-system or its component. A function is described as a set of inputs, the behavior, Firstly, the system is the first that achieves the standard notion of semantic security for data confidentiality in attribute-based deduplication systems by resorting to the hybrid cloud architecture.

E. Non-Functional Requirements

The major non-functional Requirements of the system are as follows

1) Usability: The system is designed with completely automated process hence there is no or less user intervention.

2) Reliability: The system is more reliable because of the qualities that are inherited from the chosen platform python. The code built by using python is more reliable.

3) Performance: This system is developing in the high level languages and using the advanced back-end technologies it will give response to the end user on client system with in very less time.

4) Supportability: The system is designed to be the cross platform supportable. The system is supported on a wide range of hardware and any software platform, which is built into the system.

5) Implementation: The system is implemented in web environment using Jupyter notebook software. The server is used as the intellignce server and windows 10 professional is used as the platform. Interface the user interface is based on Jupyter notebook provides server system.

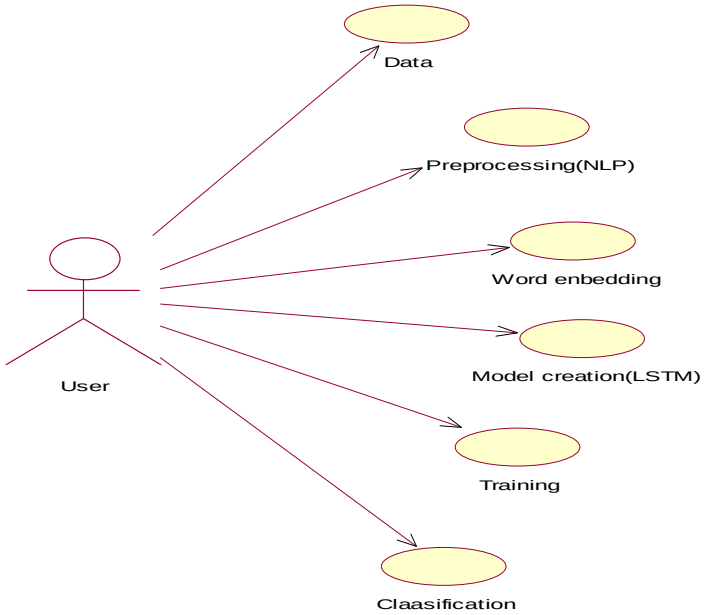
IV. DESIGN ENGINEERING

A. General

Design Engineering deals with the various UML [Unified Modelling language] diagrams for the implementation of project. Design is a meaningful engineering representation of a thing that is to be built. Software design is a process through which the requirements are translated into representation of the software. Design is the place where quality is rendered in software engineering.

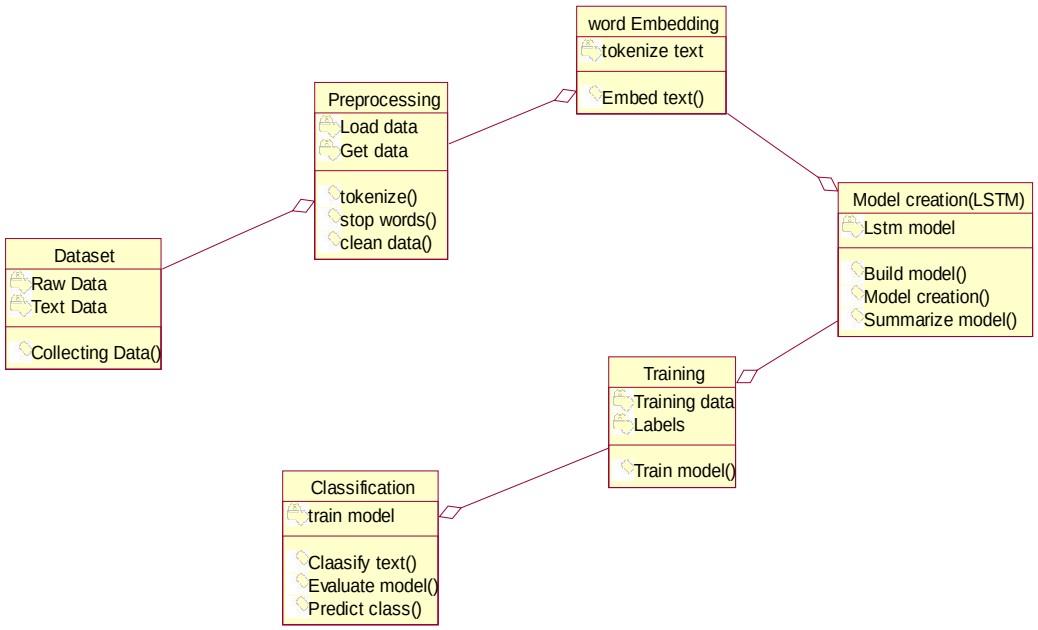
B. UML Diagrams

1) Use Case Diagram



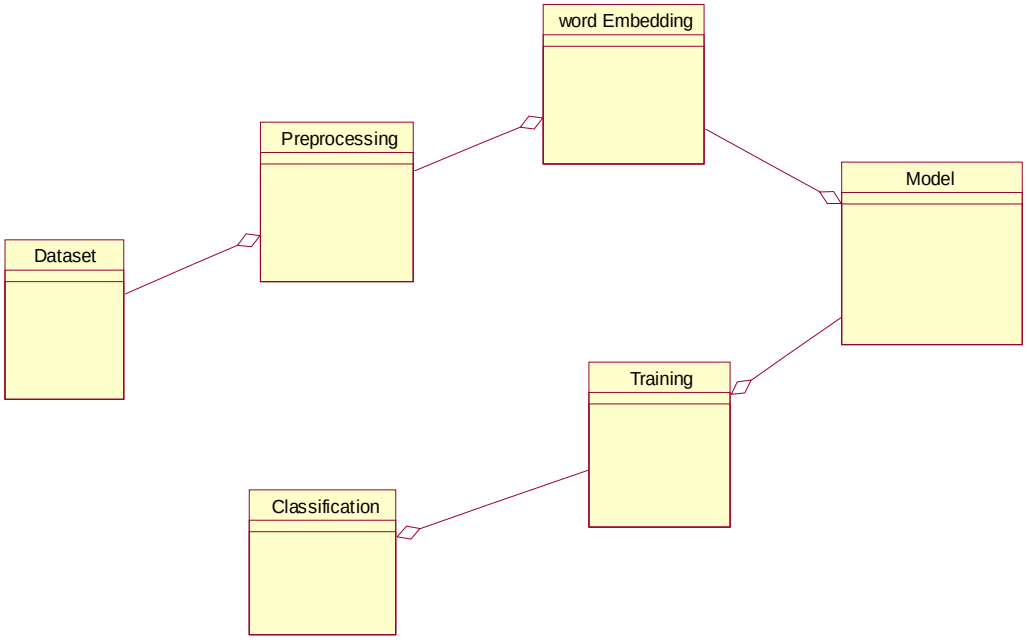
- Explanation: The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted. The above diagram consists of user as actor. Each will play a certain role to achieve the concept.

2) Class Diagram



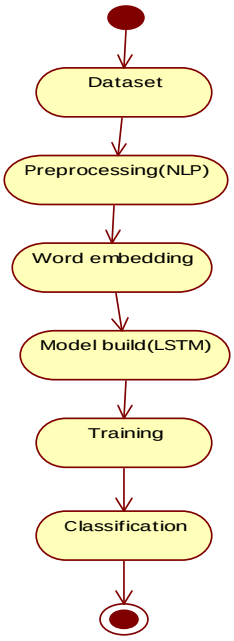
- Explanation: In this class diagram represents how the classes with attributes and methods are linked together to perform the verification with security. From the above diagram shown the various classes involved in our project.

3) Object Diagram



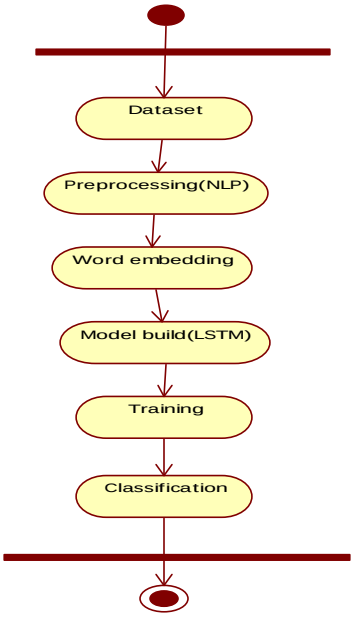
- Explanation: In the above digram tells about the flow of objects between the classes. It is a diagram that shows a complete or partial view of the structure of a modeled system. In this object diagram represents how the classes with attributes and methods are linked together to perform the verification with security.

4) State Diagram



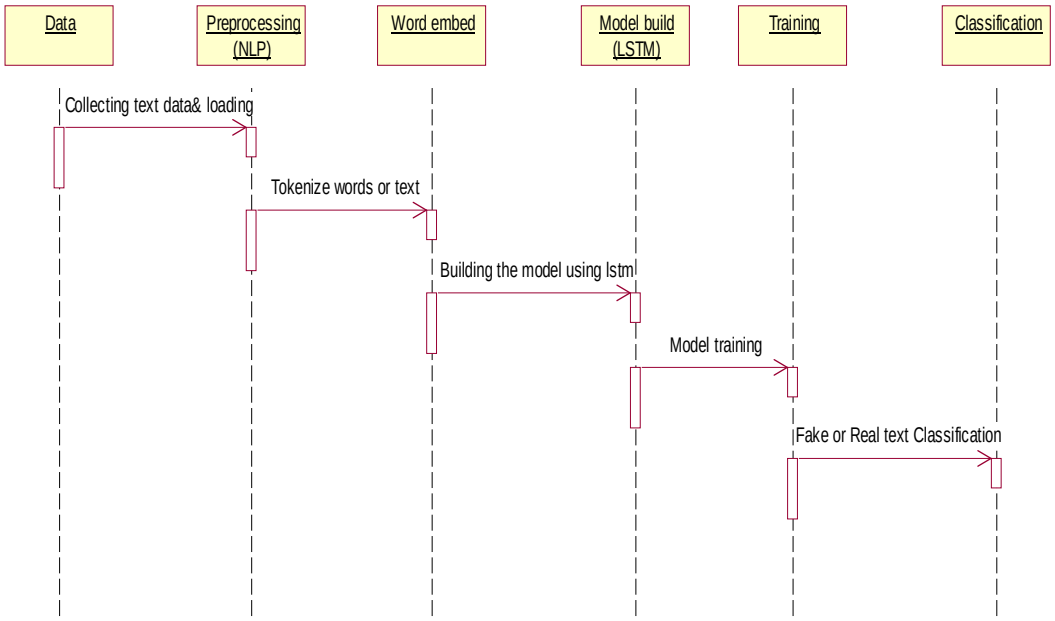
- Explanation: State diagram are a loosely defined diagram to show workflows of stepwise activities and actions, with support for choice, iteration and concurrency. State diagrams require that the system described is composed of a finite number of states; sometimes, this is indeed the case, while at other times this is a reasonable abstraction. Many forms of state diagrams exist, which differ slightly and have different semantics.

5) Activity Diagram



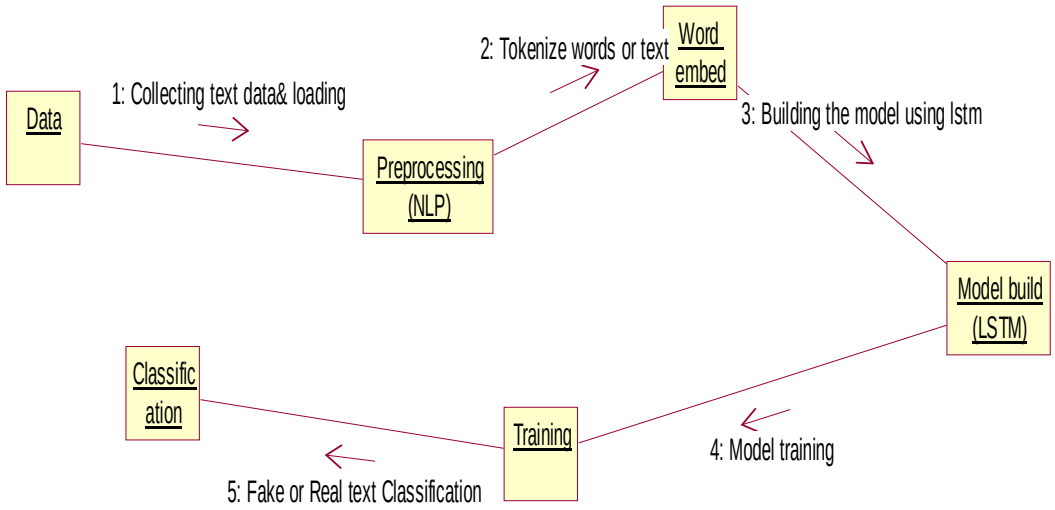
- Explanation: Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

6) Sequence Diagram



- Explanation: A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario.

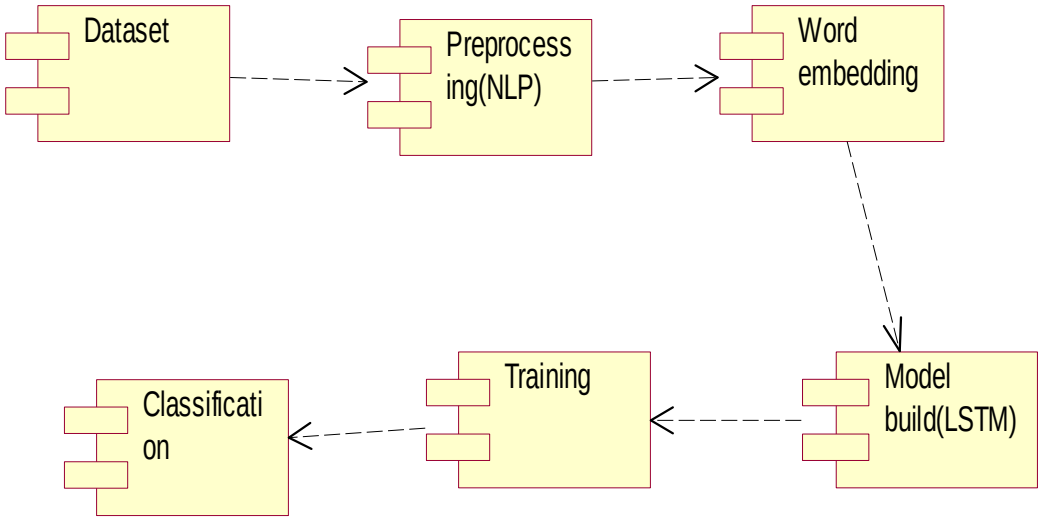
7) Collaboration Diagram



- Explanation: A collaboration diagram, also called a communication diagram or interaction diagram, is an illustration of the relationships and interactions among software objects in the Unified Modeling Language (UML). The concept is more than a decade old although it has been refined as modeling paradigms have evolved.

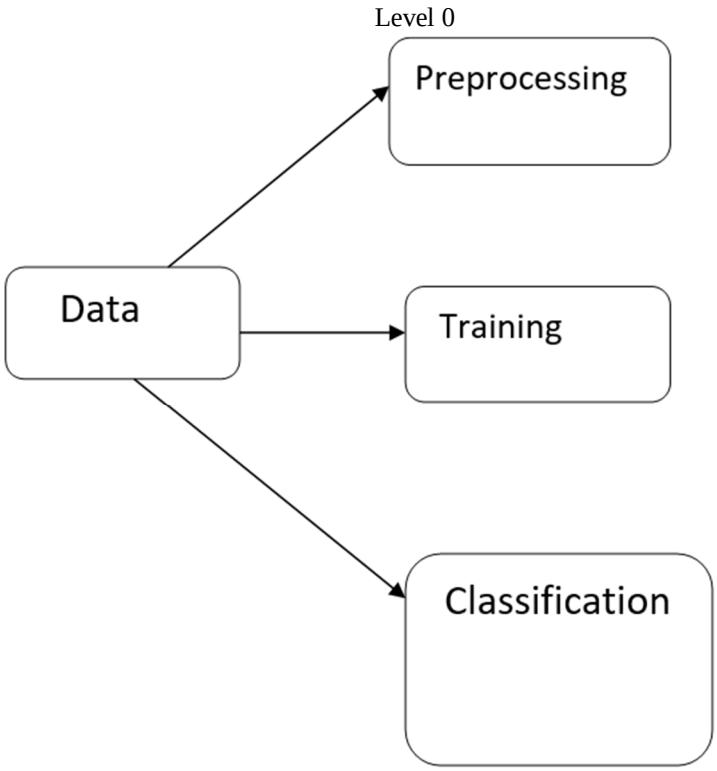
©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

8) Component Diagram



- Explanation: In the Unified Modeling Language, a component diagram depicts how components are wired together to form larger components and or software systems. They are used to illustrate the structure of arbitrarily complex systems. User gives main query and it converted into sub queries and sends through data dissemination to data aggregators. Results are to be showed to user by data aggregators. All boxes are components and arrow indicates dependencies.

9) Data Flow Diagram



Level 1

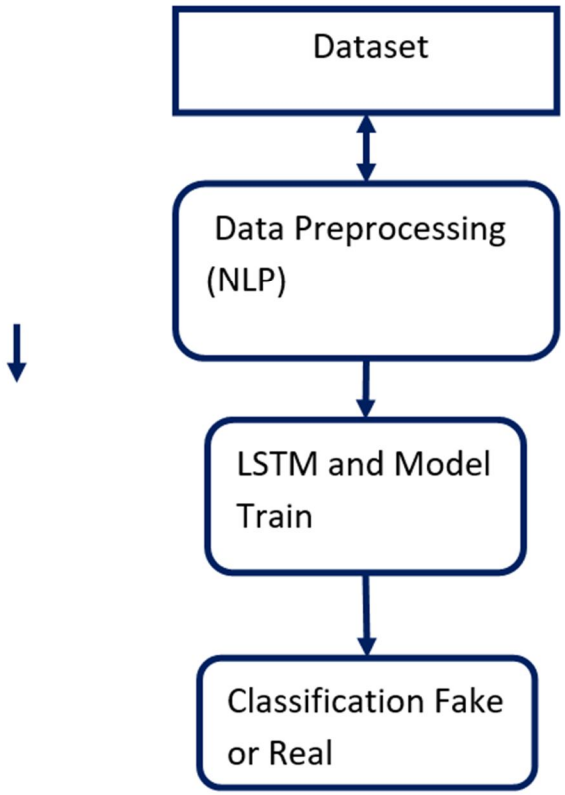
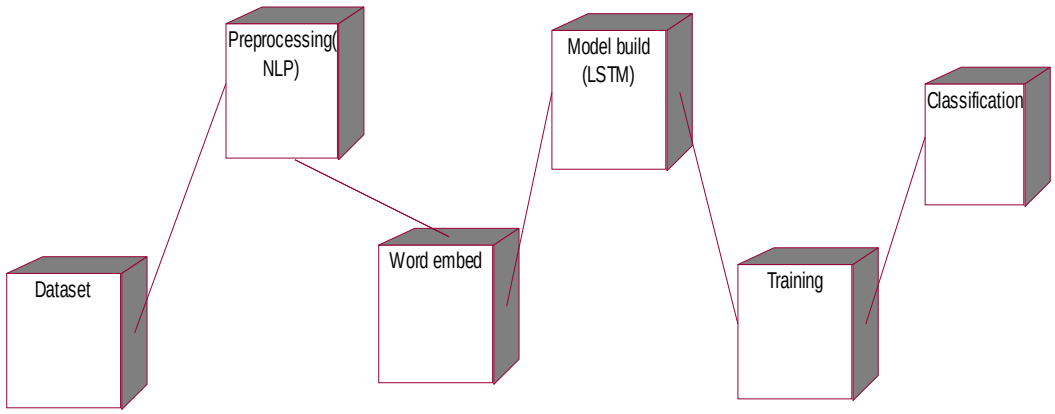


Fig 4.9: Data Flow Diagrams

- Explanation: A data flow diagram (DFD) is a graphical representation of the "flow" of data through an information system, modeling its process aspects. Often they are a preliminary step used to create an overview of the system which can later be elaborated. DFDs can also be used for the visualization of data processing (structured design). A DFD shows what kinds of data will be input to and output from the system, where the data will come from and go to, and where the data will be stored. It does not show information about the timing of processes, or information about whether processes will operate in sequence or in parallel.

10) Deployment Diagram



- Explanation: Deployment Diagram is a type of diagram that specifies the physical hardware on which the software system will execute. It also determines how the software is deployed on the underlying hardware. It maps software pieces of a system to the device that are going to execute it.

C. System Architecture

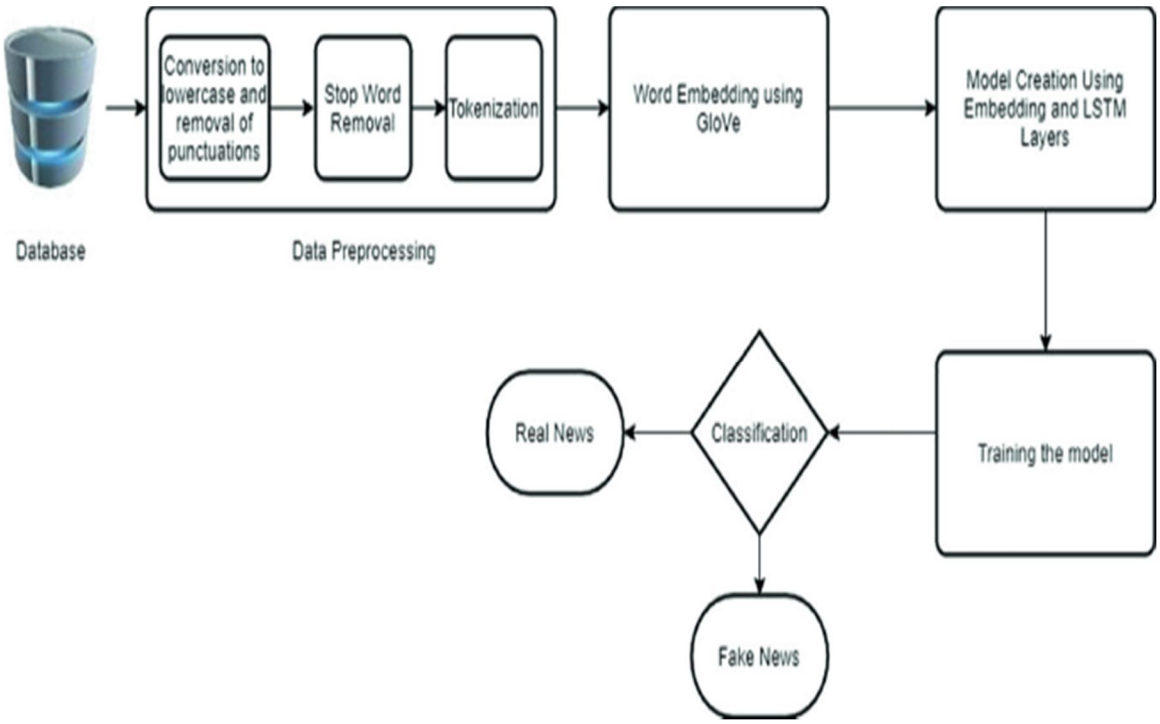


Fig 4.11: System Architecture

V. DEVELOPMENT TOOLS

A. Python

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

B. History of Python

Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands.

Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, SmallTalk, and Unix shell and other scripting languages.

Python is copyrighted. Like Perl, Python source code is now available under the GNU General Public License (GPL).

Python is now maintained by a core development team at the institute, although Guido van Rossum still holds a vital role in directing its progress.

C. Importance of Python

1) Python is Interpreted − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.

2) Python is Interactive − You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.

3) Python is Object-Oriented − Python supports Object-Oriented style or technique of programming that encapsulates code within objects.

4) Python is a Beginner's Language − Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

D. Features of Python

1) Easy-to-learn − Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
2) Easy-to-read − Python code is more clearly defined and visible to the eyes.
3) Easy-to-maintain − Python's source code is fairly easy-to-maintain.
4) A broad standard library − Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.
5) Interactive Mode − Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
6) Portable − Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
7) Extendable − You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
8) Databases − Python provides interfaces to all major commercial databases.
9) GUI Programming − Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.
10) Scalable − Python provides a better structure and support for large programs than shell scripting.

Apart from the above-mentioned features, Python has a big list of good features, few are listed below −

- It supports functional and structured programming methods as well as OOP.
- It can be used as a scripting language or can be compiled to byte-code for building large applications.
- It provides very high-level dynamic data types and supports dynamic type checking.
- IT supports automatic garbage collection.
- It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

E. Libraries used in python

numpy - mainly useful for its N-dimensional array objects.
pandas - Python data analysis library, including structures such as dataframes.
matplotlib - 2D plotting library producing publication quality figures.
scikit-learn - the machine learning algorithms used for data analysis and data mining tasks.



Figure : NumPy, Pandas, Matplotlib, Scikit-learn

VI. SOFTWARE TESTING

A. General

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue III Mar 2026- Available at www.ijraset.com

B. Developing Methodologies

The test process is initiated by developing a comprehensive plan to test the general functionality and special features on a variety of platform combinations. Strict quality control procedures are used. The process verifies that the application meets the requirements specified in the system requirements document and is bug free. The following are the considerations used to develop the framework from developing the testing methodologies.

C. Types of Tests

1) Unit Testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program input produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

2) Functional Test

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.
Invalid Input : identified classes of invalid input must be rejected.
Functions : identified functions must be exercised.
Output : identified classes of application outputs must be exercised.
Systems/Procedures: interfacing systems or procedures must be invoked.

3) System Test

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration-oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

4) Performance Test

The Performance test ensures that the output be produced within the time limits, and the time taken by the system for compiling, giving response to the users and request being send to the system for to retrieve the results.

5) Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

6) Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Acceptance testing for Data Synchronization:

The Acknowledgements will be received by the Sender Node after the Packets are received by the Destination Node
The Route add operation is done only when there is a Route request in need
The Status of Nodes information is done automatically in the Cache Updation process

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

7) Build the test plan

Any project can be divided into units that can be further performed for detailed processing. Then a testing strategy for each of this unit is carried out. Unit testing helps to identity the possible bugs in the individual component, so the component that has bugs can be identified and can be rectified from errors.

VII. FUTURE ENHANCEMENT

Future enhancements for the fake news detection project can focus on improving the accuracy, scalability, and adaptability of the system. One important area for improvement is the incorporation of multimodal data, such as images, videos, and social media metrics, alongside text. Fake news often uses misleading visuals or sensational headlines, so analyzing these elements could provide additional context and improve detection accuracy. Another enhancement could involve using pre-trained models like BERT or GPT, which are capable of capturing complex language patterns and nuances that traditional models might miss. This would allow the system to process text more effectively and improve overall performance.Addressing data imbalance is another critical area. Techniques such as Generative Adversarial Networks (GANs) or Synthetic Minority Oversampling Technique (SMOTE) could be applied to balance the dataset, especially when fake news is underrepresented. Moving the system toward real-time detection is also an important goal. Optimizing the model for real-time fake news detection would enable immediate predictions as news articles are published, making the system more practical for real-world use.Additionally, incorporating explainable AI (XAI) techniques such as SHAP or LIME could improve transparency by allowing the model to explain the reasoning behind its predictions, helping users understand why a news article was classified as fake or real. Expanding the system to support multiple languages would also make it more useful, allowing it to detect fake news in different linguistic and cultural contexts. Integrating social media data for analysis of user engagement, such as shares, likes, and comments, could provide valuable insights into the credibility of news articles.

VIII. CONCLUSION AND REFERENCES

In conclusion, this project demonstrates the potential of using NLP techniques combined with advanced models like LSTM for fake news detection, addressing a critical issue in today's digital age. By leveraging text-based features and contextual understanding, the system is able to identify patterns that distinguish real news from fake, providing a valuable tool for combating misinformation. The proposed system's ability to analyze news articles through both feature extraction and sequence modeling offers improved accuracy compared to traditional methods. However, the project also highlights areas for future improvement, such as integrating multimodal data, enhancing real-time detection capabilities, and incorporating explainable AI methods to improve model transparency. As fake news continues to evolve, further refinements and enhancements, such as multi-language support and continuous learning, will help the system adapt to new challenges, ensuring its relevance in the ongoing fight against misinformation. Ultimately, this work contributes to the broader field of fake news detection, offering a foundation for future research and development of more robust, scalable solutions.

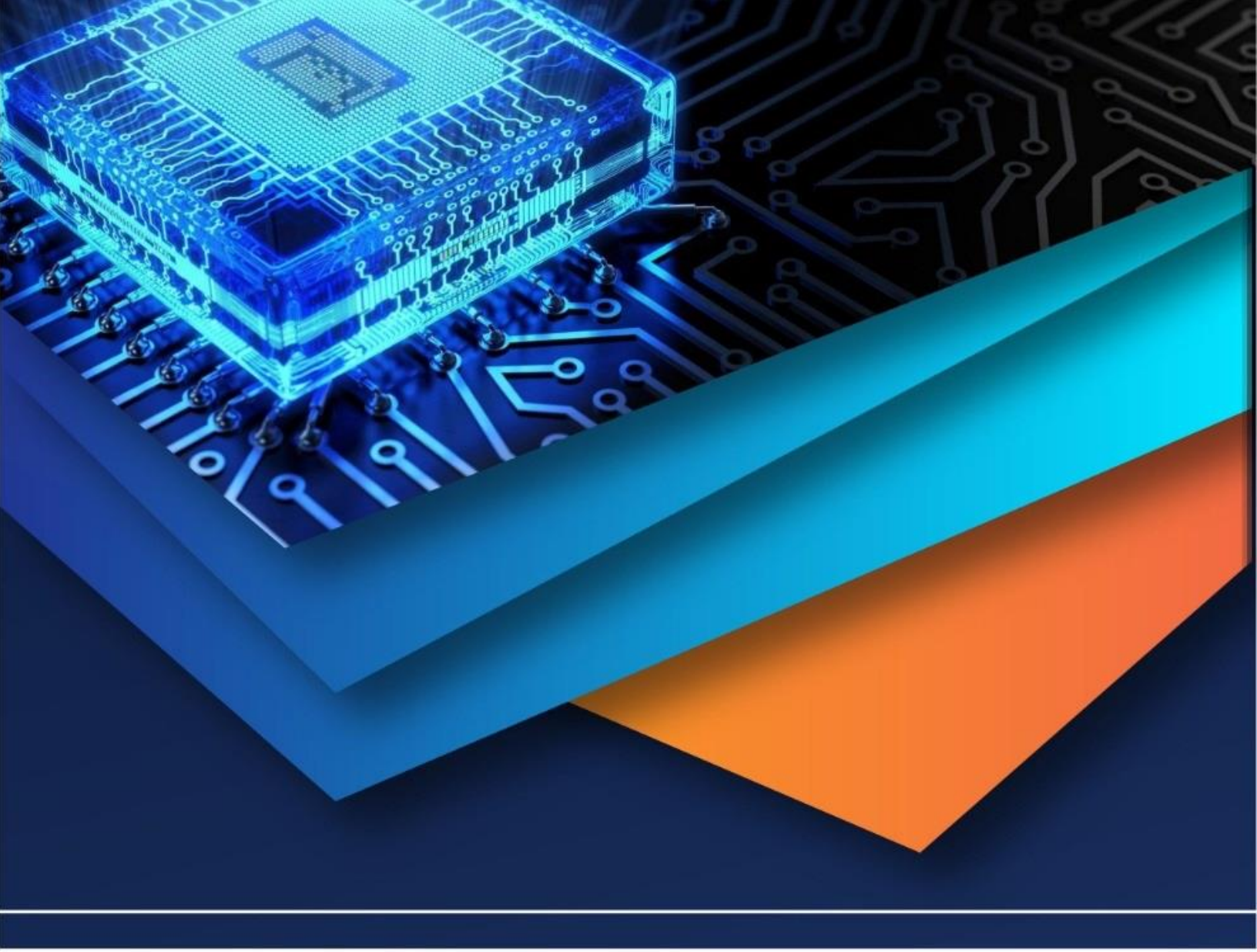
REFERENCES

[1] D. Pogue, ''How to stamp out fake news,'' Sci. Amer., vol. 316, no. 2, p. 24, Jan. 2017.
[2] H. Allcott and M. Gentzkow, ''Social media and fake news in the 2016 election,'' J. Econ. Perspect., vol. 31, no. 2, pp. 211–236, May 2017.
[3] R. Zafarani, X. Zhou, K. Shu, and H. Liu, ''Fake news research: Theories, detection strategies, and open problems,'' in Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, New York, NY, USA, Jul. 2019, pp. 3207–3208.
[4] Y. M. Rocha, G. A. de Moura, G. A. Desidério, C. H. de Oliveira, F. D. Lourenço, and L. D. de Figueiredo Nicolete, ''The impact of fake news on social media and its influence on health during the COVID-
[5] pandemic: A systematic review,'' J. Public Health, vol. 31, no. 7, pp. 1007–1016, Jul. 2023. S. Vosoughi, D. Roy, and S. Aral, ''The spread of true and false news online,'' Science, vol. 359, no. 6380, pp. 1146–1151, Mar. 2018.
[6] C. Silverman, This Analysis Shows How Viral Fake Election News Stories Outperformed Real News on Facebook. New York, NY, USA: BuzzFeed News, 2016.
[7] C. Xu and N. Yan, ''AROT-COV23: A dataset of 500k original Arabic tweets on COVID-19,'' in Proc. 4th Workshop Afr. Natural Lang. Process., 2023, pp. 1–9.
[8] C. Colomina, H. S. Margalef, R. Youngs, and K. Jones, The Impact of Disinformation on Democratic Processes and Human Rights in the World. Brussels, Belgium: European Parliament, 2021.
[9] K. Shu, A. Sliva, S. Wang, J. Tang, and H. Liu, ''Fake news detection on social media: A data mining perspective,'' ACM SIGKDD Explor. Newslett., vol. 19, no. 1, pp. 22–36, 2017.
[10] X. Zhang and A. A. Ghorbani, ''An overview of online fake news: Characterization, detection, and discussion,'' Inf. Process. Manage., vol. 57, no. 2, Mar. 2020, Art. no. 102025.
[11] X. Zhou and R. Zafarani, ''A survey of fake news: Fundamental theories, detection methods, and opportunities,'' ACM Comput. Surveys, vol. 53, no. 5, pp. 1–40, Sep. 2020.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

[12] J. Shang, J. Shen, T. Sun, X. Liu, A. Gruenheid, F. Korn, A. D. Lelkes, C. Yu, and J. Han, ''Investigating rumor news using agreement-aware search,'' in Proc. 27th ACM Int. Conf. Inf. Knowl. Manage., Oct. 2018, pp. 2117–2125.
[13] R. Zellers, A. Holtzman, H. Rashkin, Y. Bisk, A. Farhadi, F. Roesner, and Y. Choi, ''Defending against neural fake news,'' in Proc. Adv. Neural Inf. Process. Syst., vol. 32, 2019, pp. 9054–9065.
[14] W. Wang, '''Liar, liar pants on fire': A new benchmark dataset for fake news detection,'' in Proc. 55th Annu. Meeting ACL (Short Papers), vol. 2. Vancouver, BC, Canada, Jul. 2017, pp. 422–426.
[15] N. Vo and K. Lee, ''Where are the facts? Searching for fact-checked information to alleviate the spread of fake news,'' in Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP), 2020, pp. 7717–7731.
[16] P. Patwa, S. Sharma, S. Pykl, V. Guptha, G. Kumari, M . Akhtar, A. Ekbal, A. Das, and T. Chakraborty, ''Fighting an infodemic: COVID-19 fake news dataset,'' in Proc. Int. Workshop Combating Line Hostile Posts Regional Lang. During Emergency Situation. Cham, Switzerland: Springer, 2021, pp. 21–29.
[17] L. Zadeh, ''Fuzzy sets,'' Inf. Control, vol. 8, no. 3, pp. 338–353, 1965.
[18] L. Zadeh, Fuzzy Logic. New York, NY, USA: Springer, 2023, pp. 19–49.
[19] J.-S. R. Jang, ''ANFIS: Adaptive-network-based fuzzy inference system,'' IEEE Trans. Syst. Man, Cybern., vol. 23, no. 3, pp. 665–685, Jun. 1993.
[20] Y. Deng, Z. Ren, Y. Kong, F. Bao, and Q. Dai, ''A hierarchical fused fuzzy deep neural network for data classification,'' IEEE Trans. Fuzzy Syst., vol. 25, no. 4, pp. 1006–1012, Aug. 2017.
[21] R. Das, S. Sen, and U. Maulik, ''A survey on fuzzy deep neural networks,'' ACM Comput. Surv., vol. 53, no. 3, pp. 1–25, May 2020.
[22] F. Olan, U. Jayawickrama, E. O. Arakpogun, J. Suklan, and S. Liu, ''Fake news on social media: The impact on society,'' Inf. Syst. Frontiers, vol. 26, pp. 443–458, Jan. 2022.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)