FAKE NEWS DETECTION SYSTEM USING ENHANCED ROBERTA MODEL AND NLP TECHNIQUES

A PROJECT REPORT

Submitted by

Jeevanantham V	2021506031
Senthil Nathan M	2021506092
Shyaam S	2021506099
Mukesh K	2021506053

Under the supervision of

Dr. E. PUGAZHENDI

In partial fulfilment for the award of the degree
of

BACHELOR OF TECHNOLOGY

in

INFORMATION TECHNOLOGY

DEPARTMENT OF INFORMATION TECHNOLOGY
MADRAS INSTITUTE OF TECHNOLOGY CAMPUS
ANNA UNIVERSITY, CHENNAI – 600044

DECEMBER 2024

ANNA UNIVERSITY: CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project report titled "**FAKE NEWS DETECTION SYSTEM USING ENHANCED ROBERTA MODEL AND NLP TECHNIQUES" is** the bonafide work of Jeevanantham V (2021506031), Senthil Nathan M (2021506092), Shyaam S (2021506099) and Mukesh K (2016506053) who carried out the project work under my supervision.

Signature	**Signature**
Dr. E. Pugazhendi	**Dr. M. R. Sumalatha**
SUPERVISOR	**HEAD OF THE DEPARTMENT**
Teaching fellow	Professor
Department of Information Technology	Department of Information Technology
MIT Campus, Anna University	MIT Campus, Anna University
Chennai – 600044	Chennai – 600044

ACKNOWLEDGEMENT

It is essential to mention the names of the people, whose guidance and encouragement made us accomplish this project.

We express our thankfulness to our project supervisor **Dr. E. Pugazhendi**, Teaching fellow, Department of Information Technology, MIT Campus, for providing invaluable support and assistance with encouragement which aided to complete this project.

We are thankful to the panel members **Dr. Dhananjay Kumar,** Professor and **Dr. J. Dhalia Sweetlin,** Associate Professor, Department of Information Technology, MIT Campus for their invaluable feedback in reviews.

Our sincere thanks to **Dr. M. R. Sumalatha**, Professor and Head of the Department of Information Technology, MIT Campus for catering all our needs giving out limitless support throughout the project phase.

We express our gratitude and sincere thanks to our respected Dean of MIT Campus, **Dr. K. Ravichandran**, for providing excellent computing facilities throughout the project.

JEEVANANTHAM V	**2021506031**
SENTHIL NATHAN M	**2021506092**
SHYAAM S	**2021506099**
MUKESH K	**2021506053**

ABSTRACT

The rapid spread of fake news across digital platforms has created a significant need for effective detection systems. This research presents a Fake News Detection System that employs two complementary approaches: a RoBERTa-based model and an NLP-based feature extraction model, ensuring high accuracy and computational efficiency. The RoBERTa model leverages advanced transformer-based techniques to analyse textual data. To improve computational efficiency, the model incorporates layer reduction and layer retention strategies. The first three reduced layers are used for word-level and semantic feature extraction, while deeper layers are selectively retained based on the dataset size to capture contextual relationships and task-specific features. This dynamic retention strategy reduces processing time while maintaining robust performance. In parallel, the NLP-based model focuses on extracting meaningful features from the input text and these features are used to observe patterns in the data, and the input is dynamically matched against extracted patterns from the trained dataset to determine its authenticity. Both these approaches ensures a multi-faceted evaluation of textual data, balancing computational efficiency, semantic understanding, and feature-based analysis. The proposed architecture offers a reliable, scalable, and efficient solution for detecting fake news, addressing a critical challenge in today's digital age.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

LLM	Large Language Model
ISOT	Information Security and Object Technology
RoBERTa	Robustly Optimized BERT Pretraining Approach
BERT	Bidirectional Encoder Representations from Transformers
NLP	Natural Language Processing
TF-IDF	Term Frequency - Inverse Document Frequency
PoS	Part of Speech
NER	Named Entity Recognition

CHAPTER 1
INTRODUCTION

1.1 OVERVIEW

The Fake News Detection System is designed to identify and classify news articles as real or fake using advanced LLM and natural language processing (NLP) techniques. With the rise of misinformation on digital platforms, there is a growing need for reliable systems that can combat the spread of false information. This project aims to address this issue by leveraging modern tools and methodologies to ensure accurate and efficient detection. The system proposes two approaches: a transformer-based RoBERTa model and an NLP-based feature extraction model. Both methods analyze input text data to identify patterns and assess the authenticity of news articles. Both these techniques enhances the overall performance, making the system robust, scalable, and suitable for real-world applications. The proposed model focuses on balancing accuracy and efficiency to provide a reliable solution for fake news detection. By evaluating the system on benchmark datasets, its performance is measured and validated against standard metrics. This approach ensures that the proposed solutions can effectively contribute to reducing the spread of fake news, ultimately promoting more trustworthy information sharing.

1.2 RESEARCH CHALLENGES

The research challenges in this fake news detection project include data quality and diversity, as inconsistent or biased datasets can impact model performance. Another challenge is feature extraction, where identifying meaningful linguistic and statistical features (e.g., TF-IDF, sentiment, PoS) requires careful preprocessing and domain knowledge. Model scalability is an issue, as increasing dataset size demands more computational resources and training time. Additionally, generalization to unseen data is difficult due to evolving fake news patterns and language usage. Lastly, ensuring interpretability of results remains critical, as understanding how specific features influence predictions is vital for trust and transparency.

1.3 OBJECTIVE

The major objective is to develop two independent Fake News Detection Systems: one utilizing a RoBERTa-based model and the other based on NLP feature extraction techniques. The RoBERTa model aims to classify news articles using deep learning, while the NLP model focuses on extracting and analysing text features to identify fake news. The goal is to evaluate the performance of both models in detecting fake news, ensuring they are scalable, efficient, and capable of handling various dataset sizes.

1.4 SCOPE OF THE PROJECT

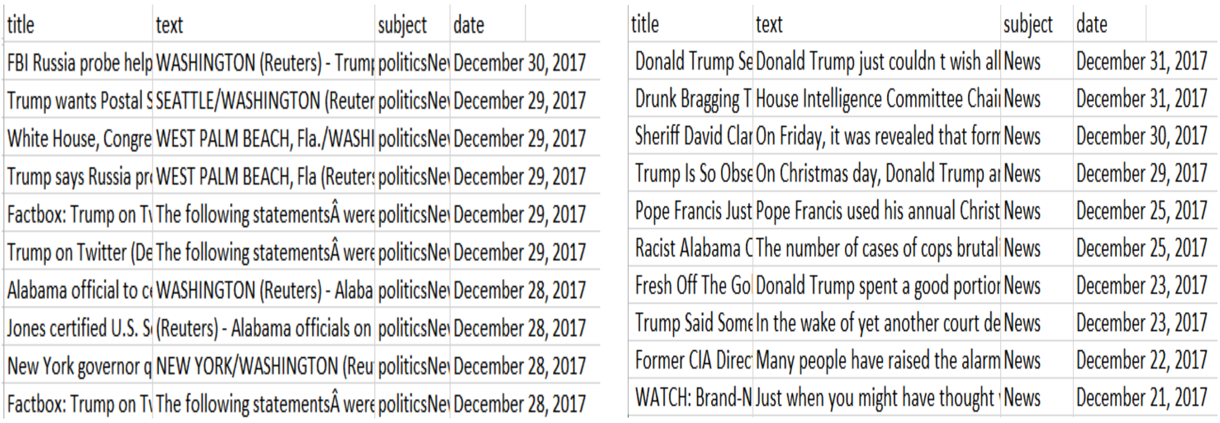
The proposed system design focuses on development of two distinct Fake News Detection Systems: one based on a RoBERTa model and the other using NLP feature extraction techniques. Each model will be evaluated independently for its effectiveness in detecting fake news, with the RoBERTa model focusing on deep learning-based classification and the NLP model utilizing feature extraction methods like TF-IDF, PoS tagging, NER, and sentiment analysis. The research work aims to provide scalable solutions for fake news detection, tailored to different dataset sizes and real-time applications.

1.5 CONTRIBUTION

The research work contributes by utilizing two advanced techniques for fake news detection: the RoBERTa-based model for deep learning-based classification and an NLP-based model for feature extraction. The RoBERTa model leverages a dynamic layer retention based on the dataset size to reduce the computational time and increase the overall efficiency of the system. The NLP model leverages TF-IDF vectorization, PoS tagging, Named Entity Recognition (NER), and sentiment analysis to analyze text and identify key features. These methods improve computational efficiency by reducing irrelevant data and enhancing accuracy through effective pattern recognition. The use of dynamic layer retention in RoBERTa optimizes computational time based on dataset size. Both techniques offer scalable, efficient solutions for fake news detection.

1.6 INPUT DATASET

The ISOT dataset is used in this project for training and evaluating the fake news detection system. It consists of several key columns: title, text, subject, and date. The title and text columns provide the content of news articles, while the source column indicates the origin of the news, and the label column categorizes each article as either real or fake. The dataset is well-structured for supervised learning tasks, making it ideal for training models to classify news articles accurately. The choice of the ISOT dataset was made due to its relevance and comprehensiveness, as it provides a substantial amount of labelled real and fake news articles. It offers diverse examples, ensuring that the models can generalize well across different types of news. Additionally, the dataset's inclusion of source information allows for deeper analysis of how source credibility may impact fake news detection. Its balanced nature and clear labelling make it a reliable and effective dataset for evaluating the performance of both RoBERTa and NLP-based models.



title	text	subject	date
FBI Russia probe help	WASHINGTON (Reuters) - Trump	politicsNe	December 30, 2017
Trump wants Postal S	SEATTLE/WASHINGTON (Reuter	politicsNe	December 29, 2017
White House, Congre	WEST PALM BEACH, Fla./WASHI	politicsNe	December 29, 2017
Trump says Russia pr	WEST PALM BEACH, Fla (Reuters	politicsNe	December 29, 2017
Factbox: Trump on Tv	The following statementsÂ were	politicsNe	December 29, 2017
Trump on Twitter (De	The following statementsÂ were	politicsNe	December 29, 2017
Alabama official to ce	WASHINGTON (Reuters) - Alaba	politicsNe	December 28, 2017
Jones certified U.S. Se	(Reuters) - Alabama officials on	politicsNe	December 28, 2017
New York governor q	NEW YORK/WASHINGTON (Reu	politicsNe	December 28, 2017
Factbox: Trump on Tv	The following statementsÂ were	politicsNe	December 28, 2017

True News

title	text	subject	date
Donald Trump Se	Donald Trump just couldn t wish all	News	December 31, 2017
Drunk Bragging T	House Intelligence Committee Chai	News	December 31, 2017
Sheriff David Clar	On Friday, it was revealed that form	News	December 30, 2017
Trump Is So Obse	On Christmas day, Donald Trump a	News	December 29, 2017
Pope Francis Just	Pope Francis used his annual Christ	News	December 25, 2017
Racist Alabama C	The number of cases of cops brutal	News	December 25, 2017
Fresh Off The Go	Donald Trump spent a good portior	News	December 23, 2017
Trump Said Some	In the wake of yet another court de	News	December 23, 2017
Former CIA Direc	Many people have raised the alarm	News	December 22, 2017
WATCH: Brand-N	Just when you might have thought	News	December 21, 2017

Fake News

Fig 1.1 Input Dataset (ISOT Dataset)

1.7 ORGANIZATION OF THE THESIS

The rest of the thesis is organized as follows. Chapter 2 presents the literature survey on approaches of machine learning and knowledge graph representation. Chapter 3 presents the architecture and system design, outlining the architecture and design of the proposed methodology 1. Chapter 4 explains the algorithms and implementation details of methodology 1. Chapter 5 presents the architecture and system design of methodology 2. Chapter 6 explains the architecture and system design of methodology 2, explaining the specifications and environment. The results achieved are presented in Chapter 7. Chapter 8 presents the conclusion and some possible avenues for future research on the topic.

CHAPTER 2
LITERATURE SURVEY

2.1 TRANSFORMER-BASED MODELS FOR FAKE NEWS DETECTION

The utilization of transformer-based models has revolutionized fake news detection by enhancing semantic and contextual understanding of textual data. Enhanced BERT models have demonstrated superior performance in distinguishing fake from real news, leveraging pre-trained embeddings and multi-layer attention mechanisms (Aljawarneh and Swedat, 2024). This approach involves layer-wise fine-tuning, enabling the model to retain deeper semantic features while adapting to task-specific data. Notably, attention-based encoders like BERT are effective in capturing short-term and long-term dependencies, making them ideal for fake news classification.

By leveraging transformers, fake news detection systems can analyse syntactic relationships and semantic nuances, achieving state-of-the-art results. These models excel in scenarios requiring contextual understanding of news content while generalizing across datasets of varying sizes and domains.

2.2 MULTI-MODAL FAKE NEWS DETECTION

Multi-modal approaches integrating textual and non-textual data, such as images, are increasingly adopted for fake news detection. One such method combines Natural Language Processing (NLP) and visual modalities to enhance prediction accuracy (Baskar et al., 2023). In these systems, word-level, phrase-level, and document-level features are extracted and fused with additional modalities to provide a comprehensive understanding of the news content.

The fusion of semantic representations with auxiliary data streams allows for improved feature learning. This multi-modal framework is particularly effective for detecting fake news across social media platforms, where text is often accompanied by visual elements. By capturing the synergy between modalities, the models address challenges posed by isolated textual or visual features.

2.3 DEEP LEARNING MODELS FOR TEXT CLASSIFICATION

Deep learning techniques, such as Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN), have been extensively employed in fake news detection. These methods facilitate feature extraction and representation learning, particularly in analysing syntactic and semantic components of news articles (Matheven and Kumar, 2022). CNNs are adept at capturing local features through convolutional operations, whereas RNNs, including LSTM and GRU architectures, excel at modelling sequential dependencies.

The hybridization of CNNs and RNNs further enhances model capabilities, enabling the detection of fake news by learning short-term and long-term relationships within textual data. These architectures have demonstrated higher performance when applied to large-scale datasets, where traditional machine learning methods often fail to generalize effectively.

2.4 OPTIMIZED LARGE LANGUAGE MODELS (LLMs) FOR RESOURCE EFFICIENCY

Large Language Models (LLMs) like BERT and its derivatives have been optimized to balance computational efficiency and performance. The retention of specific encoder layers in BERT has been shown to achieve task-specific representation without the overhead of full-layer fine-tuning (Rana et al., 2023). Layer optimization strategies involve selectively retaining layers based on dataset size to reduce computational costs while maintaining classification accuracy.

For instance, datasets with 1,000–10,000 samples utilize layers 4–8, while larger datasets (e.g., >100,000 samples) employ deeper configurations, such as layers 4–12.

This optimization ensures efficient use of resources without compromising the transformer’s ability to capture contextual and semantic features.

2.5 NLP PIPELINE-BASED MACHINE LEARNING MODELS

Traditional NLP techniques coupled with machine learning classifiers remain effective for fake news detection, particularly for smaller datasets. A standard pipeline involves text pre-processing (tokenization, stop word removal, lowercasing), feature extraction (TF-IDF, PoS tagging, Named Entity Recognition), and classification (Keya et al., 2021). Feature-based methods such as Term Frequency-Inverse Document Frequency (TF-IDF) are widely adopted for vectorization, allowing the extraction of linguistic features from textual data.

The processed features are combined and fed into lightweight classifiers, such as logistic regression or support vector machines, for binary classification. While these models lack the depth of deep learning approaches, their simplicity and interpretability make them suitable for resource-constrained environments. Sentiment analysis and lexical features further augment the classification accuracy by incorporating emotional tone and word-level semantics.

2.6 PROPOSED ARCHITECTURES FOR FAKE NEWS DETECTION

The proposed project explores two distinct approaches for fake news detection:

1. **LLM-Based Detection Using RoBERTa**:

   RoBERTa (Robustly Optimized BERT Pretraining) is employed to enhance semantic feature extraction. The model retains specific layers based on dataset size, ensuring computational efficiency while preserving task-specific representations. The architecture includes token embeddings, positional embeddings, and phrase-level semantics, followed by reduction and classification layers. Retaining layers 4–8 for smaller datasets and layers 4–12 for larger datasets ensures optimized performance.

2. **NLP Pipeline-Based Detection using Feature Extraction Techniques**: The second approach leverages a modular NLP pipeline. Text pre-processing techniques (tokenization, stop word removal, and lowercasing) are combined with advanced feature extraction, including TF-IDF vectorization, PoS tagging, Named Entity Recognition (NER), and sentiment analysis. The extracted features are aggregated and fed into a logistic regression classifier for binary classification. This method ensures interpretability and computational efficiency, making it suitable for smaller datasets or real-time applications.

2..7 SUMMARY OF THE LITERATURE SURVEY

The literature survey highlights advancements in fake news detection, emphasizing transformer-based models, multi-modal approaches, and traditional NLP pipelines. Enhanced BERT models demonstrate superior performance through contextual feature extraction, while layer optimization strategies in LLMs offer a trade-off between accuracy and resource efficiency. Deep learning architectures, such as CNNs and RNNs, are effective for sequential data analysis, capturing both local and global features.

On the other hand, feature-based NLP pipelines remain relevant for smaller datasets, offering simplicity and interpretability. The proposed project builds upon these approaches, integrating a RoBERTa-based LLM framework for semantic feature extraction and a traditional NLP pipeline with logistic regression for comparison. By combining the strengths of both methods, the project aims to provide a comprehensive and efficient solution for fake news detection.

CHAPTER 3

SYSTEM ARCHITECTURE AND DESIGN METHODOLOGY 1: ENHANCED RoBERTa MODEL

3.1 SYSTEM ARCHITECTURE

The RoBERTa-based model for fake news detection operates by first tokenizing the input text using WordPiece tokenization, which breaks the text into smaller sub word units, and converting these tokens into dense vector embeddings that capture their semantic meaning. The text is then processed through a series of transformer layers, utilizing a self-attention mechanism to understand contextual relationships and dependencies in the text.

To optimize computational efficiency, the model employs a dynamic layer retention strategy, adjusting the number of layers used based on the dataset size. The first three layers of the RoBERTa model are retained completely, as it is used to capture the word level and phrase level syntax and semantics. The other layers are dynamically reduced based on the dataset size. For smaller datasets, fewer layers are retained so the model is prevented form overfitting, while more layers are used for larger datasets to capture deeper contextual information. The output of the transformer layers is passed through a classification head, which generates a binary classification (Real or Fake) for the news article. The model is fine-tuned on a labeled dataset, like the ISOT dataset, to adapt its general knowledge to the specific task of fake news detection. This architecture allows the model to efficiently and accurately classify news articles based on their content.

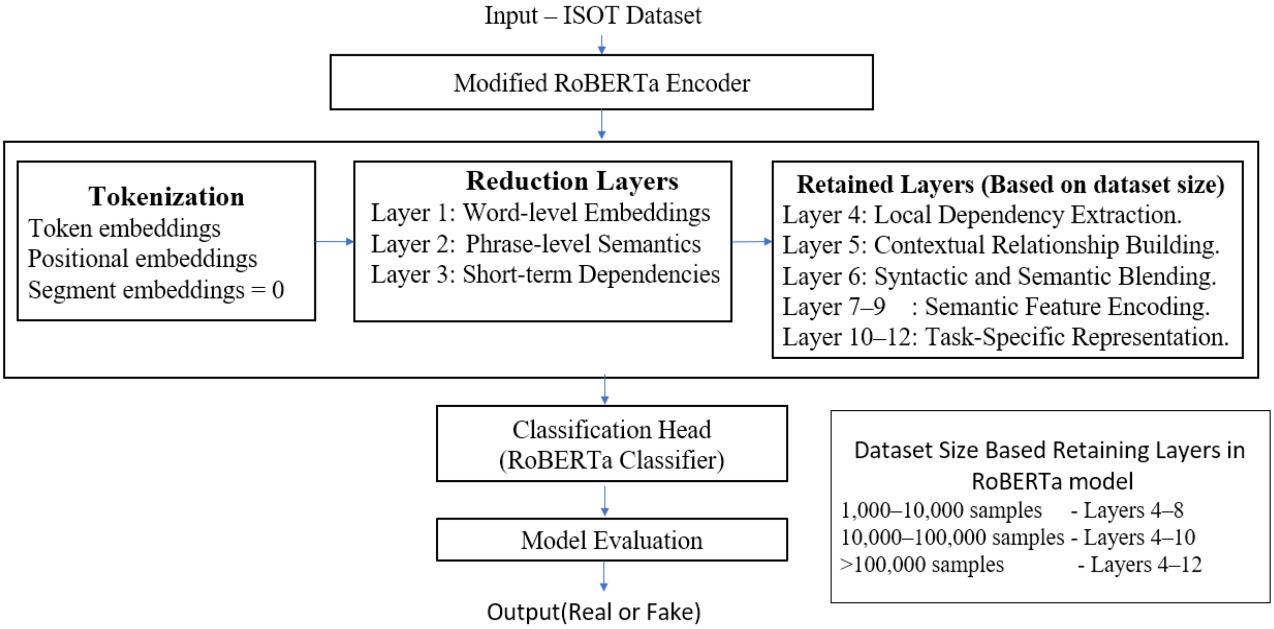

Input – ISOT Dataset
Modified RoBERTa Encoder
Tokenization
Token embeddings
Positional embeddings
Segment embeddings = 0
Reduction Layers
Layer 1: Word-level Embeddings
Layer 2: Phrase-level Semantics
Layer 3: Short-term Dependencies
Retained Layers (Based on dataset size)
Layer 4: Local Dependency Extraction.
Layer 5: Contextual Relationship Building.
Layer 6: Syntactic and Semantic Blending.
Layer 7–9 : Semantic Feature Encoding.
Layer 10–12: Task-Specific Representation.
Classification Head
(RoBERTa Classifier)
Model Evaluation
Output(Real or Fake)
Dataset Size Based Retaining Layers in RoBERTa model
1,000–10,000 samples - Layers 4–8
10,000–100,000 samples - Layers 4–10
>100,000 samples - Layers 4–12

Figure 3.1 Enhanced RoBERTa encoder methodology– System Architecture

3.2 ENHANCED RoBERTa ENCODER OVERVIEW

The RoBERTa-based model is at the core of this fake news detection system. RoBERTa (Robustly Optimized BERT Pretraining Approach) is a transformer-based model that builds upon the original BERT (Bidirectional Encoder Representations from Transformers) architecture, designed to better capture contextual relationships in text. RoBERTa's architecture consists of multiple layers of transformers, each capable of understanding and extracting complex semantic features from the text.

3.2.1 INPUT REPRESENTATION

The RoBERTa model starts by converting input text into a format that can be processed by the model. Each input text (news article, in this case) is split into tokens using a technique called WordPiece tokenization. These tokens are then mapped to embedding vectors using pre-trained embeddings from a vocabulary built during training. Each token is represented by a high-dimensional vector, where the vector values capture the meaning and context of the word in the sentence.

3.2.2 EMBEDDING LAYER

The embedding layer in the RoBERTa architecture is a critical component that transforms raw input text into a numerical representation, which can be processed by the model’s neural network layers. RoBERTa uses a sophisticated embedding strategy that involves three types of embeddings: token embeddings and position embeddings. These embeddings work together to create a rich, context-aware representation of the input text.

3.2.2.1 Token Embeddings

Token embeddings are the most fundamental type of embedding in the RoBERTa model. In the initial step, the input text is split into smaller units called tokens using **WordPiece tokenization**. Each token corresponds to a unique index in a vocabulary that the model has learned during pre-training. RoBERTa’s vocabulary is built based on sub words, which allows the model to efficiently handle unknown words and misspellings by breaking down complex words into smaller, more manageable parts. Each token in the vocabulary is associated with a high-dimensional vector (typically 768 or 1024 dimensions in base RoBERTa), and these vectors are initialized and learned during the pre-training phase. When an input text is tokenized, each token is mapped to its corresponding vector from the token embedding matrix.

3.2.2.2 Position Embeddings

Unlike models like Recurrent Neural Networks (RNNs), transformers like RoBERTa do not inherently process input data in sequential order. To address this, position embeddings are used to encode the position of each token in the input sequence. This allows the model to understand the order of tokens in a sentence or document. Position embeddings are added to the token embeddings to provide context about the relative position of tokens. The position embedding for each token is a vector that is specific to the token's position in the sequence, with the first token having a position embedding vector of a particular value, the second token having another, and so on. These position embeddings are also learned during the pre-training process, allowing the model to develop an understanding of word order and sentence structure.

3.3 TRANSFORMER ENCODER LAYERS

The core of the RoBERTa model consists of multiple **transformer encoder layers** stacked on top of each other. RoBERTa typically has up to 12 transformer layers in its base architecture, but the number of layers can vary based on the specific implementation and fine-tuning strategy. Each transformer layer consists of two main components: 1) Self-attention mechanism and 2) Feed-Forward Neural Network

3.3.1 Self-Attention Mechanism

This allows the model to focus on different parts of the input sequence when processing each word. The attention mechanism computes the relevance of every token to every other token, allowing the model to capture both local and long-range dependencies in the text. The self-attention mechanism is bidirectional, meaning it considers both the left and right context of a token simultaneously, which helps in understanding the full context of a sentence.

3.3.2 Feed-Forward Neural Network

After the self-attention layer, the output is passed through a feed-forward network consisting of fully connected layers with activation functions (usually ReLU). This allows the model to perform non-linear transformations and learn more complex patterns in the data.

3.4 DYNAMIC LAYER RETENTION

Dynamic layer retention is employed to optimize the model's computational efficiency while maintaining a high level of accuracy. The idea behind dynamic layer retention is that the deeper layers of the transformer model capture more complex and abstract representations of the text. However, these deeper layers also require more computation, which can be expensive, especially when dealing with smaller datasets or limited computational resources. To address this, the model dynamically adjusts the number of transformer layers used during processing, based on the size of the dataset.

3.4.1 LAYERS OF RoBERTa MODEL:

Layer 1: Word-level Embeddings: This layer is responsible for converting raw input text into token embeddings. Each token is mapped to a vector that represents its meaning, learned during pre-training.

Layer 2: Phrase-level Semantics: This layer processes token embeddings to understand phrase-level semantics. It captures relationships between tokens in a phrase, providing a basic understanding of sentence-level meaning.

Layer 3: Short-term Dependencies: Layer 3 focuses on learning short-term dependencies between adjacent tokens. It helps the model understand immediate contextual relationships within the input text, such as word order and simple syntax.

Layer 4: Local Dependency Extraction: At this layer, the model extracts local dependencies, which are relationships between tokens that are close to each other within a sentence. This layer helps capture more nuanced sentence structure.

Layer 5: Contextual Relationship Building: Layer 5 begins to build global relationships between words. It incorporates broader context to understand how different parts of the sentence or document relate to each other.

Layer 6: Syntactic and Semantic Blending: This layer blends syntactic structures (grammar) with semantic understanding (meaning). It refines the model’s understanding of how sentence structure influences meaning.

Layer 7–9: Semantic Feature Encoding: Layers 7 to 9 encode more complex semantic features and abstract patterns. They focus on capturing deeper relationships and conceptual information within the text.

Layer 10–12: Task-Specific Representation: The final layers are fine-tuned to extract task-specific representations. They focus on understanding the text in the context of the specific downstream task, such as fake news detection, by identifying relevant patterns and features.

3.4.2 LAYER RETENTION BASED ON DATASET SIZE:

1. Smaller Datasets (1,000 to 10,000 samples): For smaller datasets, the model uses only the 4th to 8th layers of the RoBERTa architecture. These layers capture sufficient contextual information without the need for deeper layers, thus reducing computational complexity and training time while still providing effective fake news detection. Reduces overfitting by avoiding unnecessary complexity and captures essential features.

2. Medium-Sized Datasets (10,000 to 100,000 samples): For medium-sized datasets, the model uses layers 4 to 10. This ensures a deeper understanding of the input text while balancing performance and resource usage. Balances complexity and depth to extract more nuanced features.

3. Larger Datasets (over 100,000 samples): For large datasets, the full range of layers from 4 to 12 is retained. This allows the model to extract more detailed and complex features, improving its ability to discern between real and fake news articles. This approach leverages full model capacity to maximize accuracy.

3.5 CLASSIFICATION HEAD

After passing through the transformer layers, the output representation of the input text is derived from the final hidden states of the transformer layers. Each token has a corresponding hidden state, but for text classification tasks like fake news detection, the [CLS] token (which is added at the beginning of the input) is typically used as the aggregated representation of the entire input. This hidden state serves as a summary of the input text and is passed to the classification head for generating the final prediction.

3.6 MODEL EVALUATION AND OUTPUT

The aggregated representation (from the [CLS] token) is then passed through a feed-forward neural network in the classification head, which is responsible for outputting the final binary classification: **real** or **fake**. The model uses a SoftMax activation function to predict the likelihood of the input belonging to each class. The final model is then evaluated and classification report metrics are generated, providing a comprehensive assessment of its predictive capabilities.

CHAPTER 4

ALGORITHM DEVELOPMENT AND IMPLEMENTATION METHODOLOGY 1: ENHANCED RoBERTa MODEL

4.1 DATA PREPROCESSING AND TOKENIZATION

Text data is pre-processed by cleaning and formatting it for model input, which includes removing unnecessary symbols, handling case sensitivity, and tokenizing the text. Use the tokenizer from the Hugging Face transformers library to tokenize the input text. This converts each news article into token IDs based on RoBERTa's vocabulary.

4.2 DYNAMIC LAYER RETENTION AND MODEL TRAINING

Depending on the dataset size, specific layers of the RoBERTa model are dynamically retained to optimize performance. Smaller datasets use fewer layers (e.g., layers 4-8), while larger datasets retain more layers (e.g., layers 4-12). Custom logic in the model training pipeline dynamically adjusts the number of layers used based on the input dataset size. This is achieved by modifying the model's configuration before training.

4.2.1 ALGORITHM

1. Start

2. Load the pre-trained RoBERTa model
   model = RobertaForSequenceClassification.from_pretrained("roberta-base", num_labels=2)

3. Set dynamic layer retention based on dataset size
   3.1 Small Dataset (≤ 10,000 samples): Freeze layers 0-3 and layers 8-11.
   3.2 Medium Dataset (≤ 100,000 samples): Freeze layers 0-3 and layers 10-11.
   3.3 Large Dataset (> 100,000 samples): Freeze layers 0-3 only.

4. Freeze the layers by setting their "requires_grad" attribute to "False"

```
    for layer_idx in layers_to_freeze:
    for param in model.roberta.encoder.layer[layer_idx].parameters():
        param.requires_grad = False
5. Set up Training arguments including batch size, learning rate, epochs, and evaluation settings.
6. Train the model using the Trainer class, passing in the dataset and evaluation settings
7. End
```

4.3 PROPOSED SYSTEM IMPLEMENTATION

The proposed system for fake news detection utilizes a fine-tuned RoBERTa model to classify news articles as real or fake. The system begins with tokenizing the input text using the RoBERTa tokenizer, which converts the raw text into token IDs using token embedding and position embedding techniques, applying padding and truncation to ensure consistent input lengths for the model. The RoBERTa model, pre-trained on large corpora, is adapted for sequence classification by fine-tuning it on the specific fake news dataset. The key innovation in this implementation is the dynamic layer retention strategy, which adapts the number of layers used in the model based on the dataset size. For smaller datasets (up to 10,000 samples), the first three layers are retained, while for medium-sized datasets (up to 100,000 samples), layers 4 to 10 are used. For large datasets (over 100,000 samples), the model utilizes all layers (1 to 12) to capture deeper context and improve performance. This dynamic approach reduces computational time for smaller datasets while preserving the model's accuracy and efficiency. The model is trained using the Hugging Face Trainer with specific settings for learning rate, batch size, and evaluation strategies, ensuring the best model is selected based on accuracy. This system enables efficient and scalable fake news detection across varying dataset sizes, making it adaptable to different real-world applications.

CHAPTER 5

SYSTEM ARCHITECTURE AND DESIGN METHODOLOGY 2: NLP TECHNIQUES

5.1 SYSTEM ARCHITECTURE

The proposed system architecture for fake news detection using Natural Language Processing (NLP) is designed to systematically process textual data, extract meaningful features, and classify news articles as either real or fake. The input to the system comprises raw text from the ISOT dataset, which undergoes a sequence of carefully constructed processing stages. At its core, the architecture is divided into two major components: The Data Pre-processing Module and the Feature Extraction Module, both of which are crucial for building a robust and accurate classification system.

The Data Pre-processing Module plays a foundational role in preparing the raw input text. Initially, the input data is cleaned to remove noise, irrelevant symbols, and special characters that do not contribute to semantic understanding. The cleaning phase ensures the removal of punctuation marks, unnecessary spaces, and numerical values, which could distort downstream processing. Once the text is cleaned, it is tokenized into individual units, breaking the input into words or tokens. These tokens serve as the fundamental building blocks for linguistic analysis. To maintain uniformity, the text is standardized by converting it to lowercase, ensuring that words are treated consistently regardless of their case. Stop words, such as is, and, or the, which do not provide substantial meaning, are removed to reduce the dimensionality of the data. This pre-processing pipeline collectively enhances the quality of the input text, facilitating the extraction of critical features required for model training.

The processed text is then passed into the Feature Extraction Module, where various linguistic features are systematically extracted to represent the data in a numerical form. Term Frequency-Inverse Document Frequency (TF-IDF) is first applied to assess the importance of words within the dataset. TF-IDF assigns higher weights to words that are frequent in a document but rare across the corpus, making it effective for identifying discriminative terms. This numerical representation of word importance forms the basis for further analysis.

Following TF-IDF, Part-of-Speech (PoS) tagging is performed to label words with their grammatical roles, such as nouns, verbs, and adjectives. PoS tagging captures linguistic patterns that are often indicative of manipulative or factual content in news articles. The distribution of specific grammatical elements, such as nouns and verbs, provides insights into the textual structure and helps identify patterns commonly associated with fake news.

In addition to syntactic features, Named Entity Recognition (NER) is applied to detect and classify named entities within the text. Entities such as names, locations, and organizations are identified, which helps in flagging potential inconsistencies or overuse of specific references. Fake news often exploits prominent names and events to mislead readers, and NER serves as a critical feature to capture such anomalies. The frequency of named entities is quantified, providing additional inputs for the classification process.

To complement the structural and syntactic features, Sentiment Analysis is employed to evaluate the emotional tone conveyed by the text. Sentiment polarity scores are calculated to determine whether the content exhibits positive, negative, or neutral sentiment. Fake news articles often rely on emotionally charged language to manipulate readers, making sentiment analysis a valuable indicator for detection. By analysing the sentiment polarity, the system identifies exaggerated or manipulative language that could signal the presence of fake news.

Together, these extracted features – TF-IDF weights, PoS tags, named entities, and sentiment polarity – provide a comprehensive representation of the input text. The features are then combined into a unified dataset, which serves as the input for the classification stage. The architecture ensures that the text is transformed into a structured, feature-rich format that can be effectively processed by machine learning algorithms.

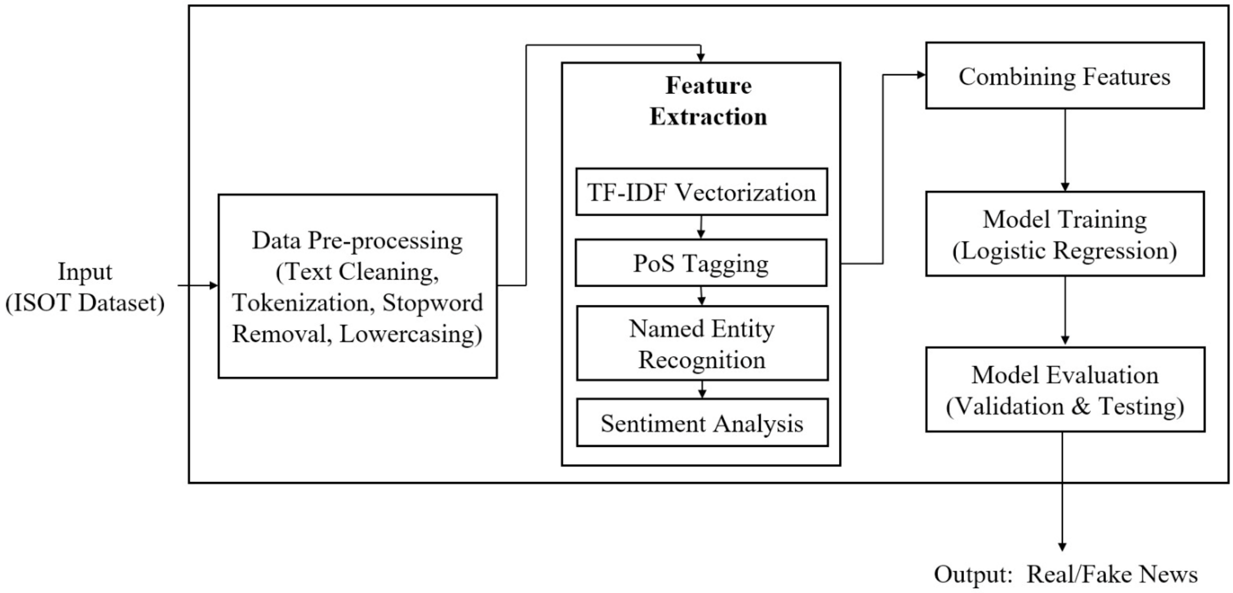


Figure 5.1 – NLP pipeline based methodology: System Architecture

5.2 DATA PREPROCESSING

The Data Preprocessing Module serves as the initial phase of the system, ensuring that the raw text data is cleaned and standardized for further analysis. The preprocessing process begins with text cleaning, where irrelevant characters, punctuation marks, and special symbols are systematically removed. This step is essential to eliminate noise that could otherwise distort the feature extraction process. Once the text is cleaned, it undergoes tokenization, where the input text is broken into individual words or tokens. Tokenization facilitates a granular analysis of the text, allowing the system to analyse and manipulate word-level features.

To ensure uniformity across the dataset, the text is standardized by converting all tokens to lowercase. Standardization helps avoid redundancy caused by case variations, ensuring that words like News and news are treated identically. In addition, stopword removal is applied to eliminate words that do not carry significant semantic value. Common stopwords, such as the, is, and in, are removed to reduce the dimensionality of the text and focus on words that are more informative. By the end of this module, the text data is transformed into a clean, tokenized, and standardized format, ready for feature extraction. The tokens are then passed onto a Feature Extraction module.

5.3 FEATURE EXTRACTION MODULE

The Feature Extraction Module is responsible for transforming the preprocessed text into numerical features that capture its semantic, syntactic, and emotional properties. The first step in this module involves the application of Term Frequency-Inverse Document Frequency (TF-IDF).

5.3.1 TF-IDF

It evaluates the importance of words by analysing their frequency in a document relative to the entire corpus. Words that are frequent within a document but rare across the dataset are assigned higher weights, enabling the system to identify discriminative terms that contribute significantly to classification. This transformation produces a numerical representation of the input text, which serves as the foundation for further linguistic analysis.

5.3.2 PART-OF-SPEECH (POS) TAGGING

It is performed to analyse the grammatical structure of the text. Words are tagged with their respective roles, such as nouns (NN), verbs (VB), and adjectives (JJ). Fake news often exhibits distinct linguistic patterns, such as an overuse of emotionally charged adjectives or specific nouns. By capturing these patterns, PoS tagging provides valuable syntactic insights that contribute to distinguishing fake news from real news.

5.3.3 NAMED ENTITY RECOGNITION

To identify named entities within the text, Named Entity Recognition (NER) is applied. NER detects entities such as names, organizations, and locations, which are often manipulated in fake news to mislead readers. The frequency and distribution of named entities are quantified, providing a critical feature for identifying articles that exploit prominent references for deceptive purposes.

5.3.4 SENTIMENT ANALYSIS

Finally, Sentiment Analysis is conducted to evaluate the emotional tone of the text. Sentiment polarity scores are computed to determine whether the content conveys positive, negative, or neutral sentiment. Fake news articles frequently employ emotionally charged language to evoke strong reactions from readers. By analyzing sentiment polarity, the system identifies exaggerated or manipulative language that may indicate the presence of fake news.

The combination of these features – TF-IDF vectors, PoS tags, NER counts, and sentiment polarity scores – creates a rich and structured representation of the input text. These features are subsequently fed into machine learning algorithms for model training, validation, and testing. The systematic extraction of linguistic features ensures that the system captures the textual, structural, and emotional characteristics necessary for accurate classification.

5.4 MACHINE LEARNING MODEL

For classifying the input news into real or fake, this methodology uses Logistic Regression machine learning algorithm. It is a simple yet effective algorithm, identifies the relationships between these features and the target labels, learning a decision boundary to separate fake news from real news. This model's interpretability allows for better understanding of which features contribute most to the classification, making it a reliable choice for detecting fake news in a structured and explainable manner.

5. 5 MODEL EVALUATION AND OUTPUT

The system architecture for fake news detection using NLP integrates data pre-processing and feature extraction modules to generate representative linguistic features. The pre-processing pipeline ensures clean and standardized text, while feature extraction focuses on TF-IDF weighting, grammatical patterns (PoS), entity recognition (NER), and sentiment polarity. These features are subsequently combined and fed into a machine learning classifier for binary classification. The modular design ensures scalability, interpretability, and computational efficiency, making the system robust for real-world fake news detection. The final model is then evaluated and classification report metrics are generated, providing a comprehensive assessment of its predictive capabilities.

CHAPTER 6

ALGORITHM DEVELOPMENT AND IMPLEMENTATION METHODOLOGY 2: NLP TECHNIQUES

6.1 DATA PREPROCESSING

Text data is pre-processed by cleaning and formatting it for input, which includes removing unnecessary symbols, removing stop words and lowercase conversion. After the input pre-processing the dataset is passed onto the Feature extraction module.

6.2 FEATURE EXTRACTION

Feature extraction plays a crucial role in transforming raw text data into meaningful numerical features for the fake news detection system. TF-IDF Vectorization converts the textual content into a matrix of numerical values, capturing the importance of words across the dataset while reducing dimensionality. PoS Tagging extracts grammatical patterns by identifying counts of nouns, verbs, and adjectives, enabling the model to understand the syntactic structure of the text. Additionally, Sentiment Analysis leverages polarity scores to capture the emotional tone of the content, providing insights into the sentiment conveyed. These extracted features are combined to build a comprehensive feature set, which enhances the Logistic Regression model's ability to classify news as fake or real effectively.

6.3 LOGISTIC REGRESSION CLASSIFIER

The Logistic Regression model serves as a simple yet effective machine learning approach for fake news classification. It is trained on extracted features such as TF-IDF scores, Part-of-Speech (PoS) tagging results, and sentiment polarity, which collectively represent the textual patterns in the dataset. The model learns to identify relationships between these features and the binary output (real or fake news). During training, the model iteratively adjusts its coefficients to minimize errors, ensuring better accuracy. By evaluating on a validation dataset, the model’s performance is measured using accuracy and classification metrics. Its simplicity, interpretability, and efficiency make Logistic Regression suitable for the task while providing competitive results.

6.3.1 ALGORITHM

1. Start
2. Load and Prepare the Dataset
    - Load the ISOT dataset.
    - Split the data into training, validation, and testing subsets:

      train_data, test_data = train_test_split(data, test_size=0.2, random_state=42)

      train_data, val_data = train_test_split(train_data, test_size=0.1, random_state=42)
3. Preprocess the Text Data
    - Convert text to lowercase to standardize it.
    - Remove unnecessary characters such as punctuation and special symbols.
    - Filter out common stopwords using a predefined stopword list (e.g., NLTK).
4. Feature Extraction
    - TF-IDF Vectorization: Convert text into numerical features using the TfidfVectorizer with a feature limit of 5000.

      ```
      tfidf_vectorizer = TfidfVectorizer(max_features=5000)
      ```
    - Parts of Speech (PoS) Tagging: Extract counts of nouns, verbs, and adjectives using NLTK's PoS tagging:

      ```
      pos_tags = nltk.pos_tag(tokens)
      nouns = sum(1 for word, tag in pos_tags if tag.startswith('NN'))
      verbs = sum(1 for word, tag in pos_tags if tag.startswith('VB'))
      ```

```
adjectives = sum(1 for word, tag in pos_tags if
tag.startswith('JJ'))
```

- Named Entity Recognition (NER): Extract counts of named entities (e.g., persons, locations, organizations) using an NER tool such as SpaCy:

```
import spacy
nlp = spacy.load("en_core_web_sm")
doc = nlp(text)
ner_count = len([ent for ent in doc.ents])
```

- Sentiment Analysis: Compute the sentiment polarity of each document using TextBlob.

5. Combine Extracted Features
   - Merge TF-IDF features, PoS features, NER counts, and sentiment scores into a single feature matrix:

```
    combined = np.hstack((tfidf.toarray(), pos_array,
np.array(ner_count).reshape(-1, 1), np.array(sentiment).reshape(-1, 1)))
```

6. Model Training
   - Initialize and train a Logistic Regression model using the extracted features:

```
model = LogisticRegression(max_iter=1000)
model.fit(X_train, y_train)
```

7. Model Evaluation
   - Predict labels for the validation and testing datasets.
   - Calculate and report the accuracy.
8. End

6.4 PROPOSED SYSTEM IMPLEMENTATION

The proposed system of the fake news detection system begins with data pre-processing, where the ISOT dataset, which includes columns like title, text, source, and label, is cleaned and prepared. This involves tokenizing the text, removing stop words, and converting the text to lowercase for consistency. Feature extraction follows, where multiple techniques are used to capture meaningful patterns. First, **TF-IDF Vectorization** is applied to convert the textual data into numerical vectors, highlighting the most important terms in the corpus. Then, **Part-of-Speech (PoS) tagging** is performed to analyse the grammatical structure of the text, capturing the frequency of nouns, verbs, and adjectives. Additionally, **Sentiment Analysis** using TextBlob is applied to gauge the sentiment of the text, which helps in distinguishing real from fake news based on emotional tone.

These features are then combined and used to train a **Logistic Regression** model, which learns to classify news as either real or fake. The model’s performance is evaluated on the validation dataset, with metrics like accuracy and classification report providing insights into its effectiveness. This structured approach combines traditional NLP techniques with machine learning to detect fake news.

CHAPTER 7
RESULTS AND DISCUSSIONS

7.1 IMPLEMENTATION ENVIRONMENT

The fake news detection system was implemented using Google Colab, a cloud-based environment that provides robust computing resources, including access to GPUs for accelerated model training. The entire implementation was carried out in a Jupyter notebook (.ipynb) file, which allowed for interactive development and easy execution of code blocks. Google Colab's seamless integration with Python libraries, especially those required for natural language processing and deep learning, enabled efficient training and evaluation of the RoBERTa model and NLP techniques. The dataset, provided in CSV format, was imported into the Colab environment and pre-processed accordingly. After pre-processing, the tokenized data was used to fine-tune the RoBERTa model for fake news classification. Also for NLP the feature extraction techniques are used to find the patterns of the news articles in the dataset.

For model storage and future reference, the trained model was saved and stored in Google Drive, ensuring that it could be easily accessed and reused for further analysis or experimentation. Google Drive's cloud storage provided a secure and scalable solution for model storage, while also enabling easy sharing and collaboration with other team members. The use of Google Colab, combined with Google Drive for model storage, created a flexible and accessible environment for implementing and refining the fake news detection system.

7.2 METHODOLOGY 1: ENHANCED ROBERTA MODEL

7.2.1 TOKENIZATION

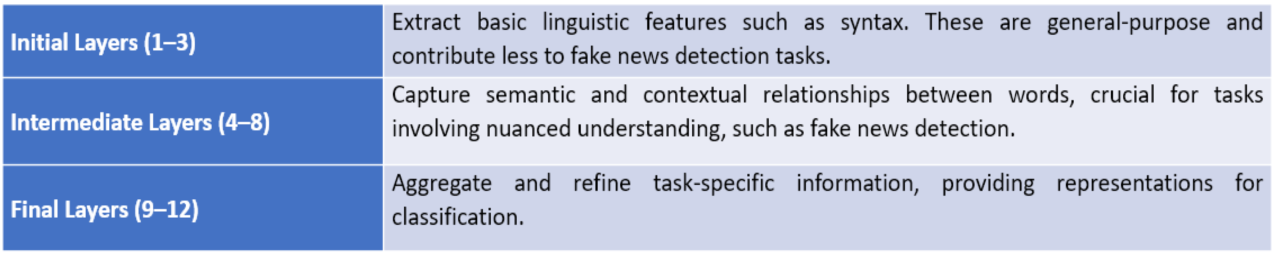
The input text is split into smaller units called tokens. These tokens are usually words or sub words, which the model can then process more efficiently. The tokenizer converts the text into a sequence of integers, where each integer corresponds to a specific token in the model's vocabulary. By utilizing the RobertaTokenizer class, the text is preprocessed, which includes truncating or padding the sequences to ensure uniformity in length, allowing the model to handle varied input sizes effectively. This tokenization step is crucial for converting human-readable text into a format that the machine learning model can understand and process.

```
tokenizer_config.json: 100%   25.0/25.0 [00:00<00:00, 582B/s]
vocab.json: 100%   899k/899k [00:00<00:00, 3.83MB/s]
merges.txt: 100%   456k/456k [00:00<00:00, 23.9MB/s]
tokenizer.json: 100%   1.36M/1.36M [00:00<00:00, 20.8MB/s]
config.json: 100%   481/481 [00:00<00:00, 36.0kB/s]
```

Fig 7.1 Tokenization of Input text

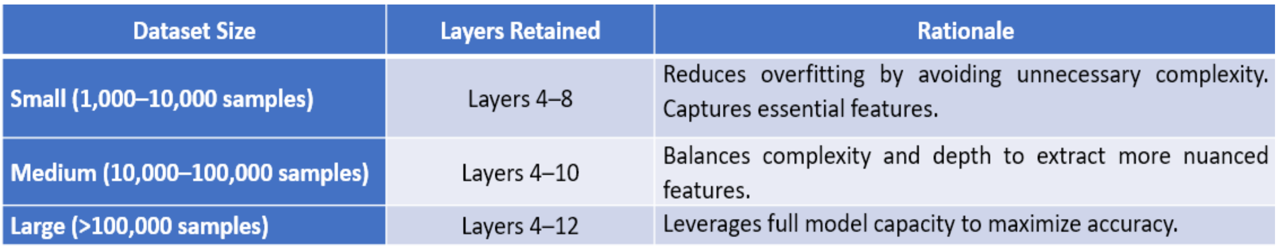
7.2.2 DYNAMIC LAYER RETENTION

Dynamic layer retention in the RoBERTa model helps optimize computational efficiency by selectively freezing layers based on the dataset size. For smaller datasets, fewer layers are retained, which reduces the model's complexity and accelerates both training and inference times. As the dataset size increases, more layers are retained to capture richer features, balancing computational cost with model accuracy. This approach minimizes unnecessary computations while ensuring high performance, improving overall efficiency. By adapting the model’s architecture based on data size, we can maintain a fast and resource-efficient system.



Initial Layers (1–3)	Extract basic linguistic features such as syntax. These are general-purpose and contribute less to fake news detection tasks.
Intermediate Layers (4–8)	Capture semantic and contextual relationships between words, crucial for tasks involving nuanced understanding, such as fake news detection.
Final Layers (9–12)	Aggregate and refine task-specific information, providing representations for classification.

Table 7.1 Layer categorization



Dataset Size	Layers Retained	Rationale
Small (1,000–10,000 samples)	Layers 4–8	Reduces overfitting by avoiding unnecessary complexity. Captures essential features.
Medium (10,000–100,000 samples)	Layers 4–10	Balances complexity and depth to extract more nuanced features.
Large (>100,000 samples)	Layers 4–12	Leverages full model capacity to maximize accuracy.

Table 7.2 Dynamic Retention Based on Dataset Size

COMPARISON OF RESPONSE TIME AFTER REDUCING LAYERS

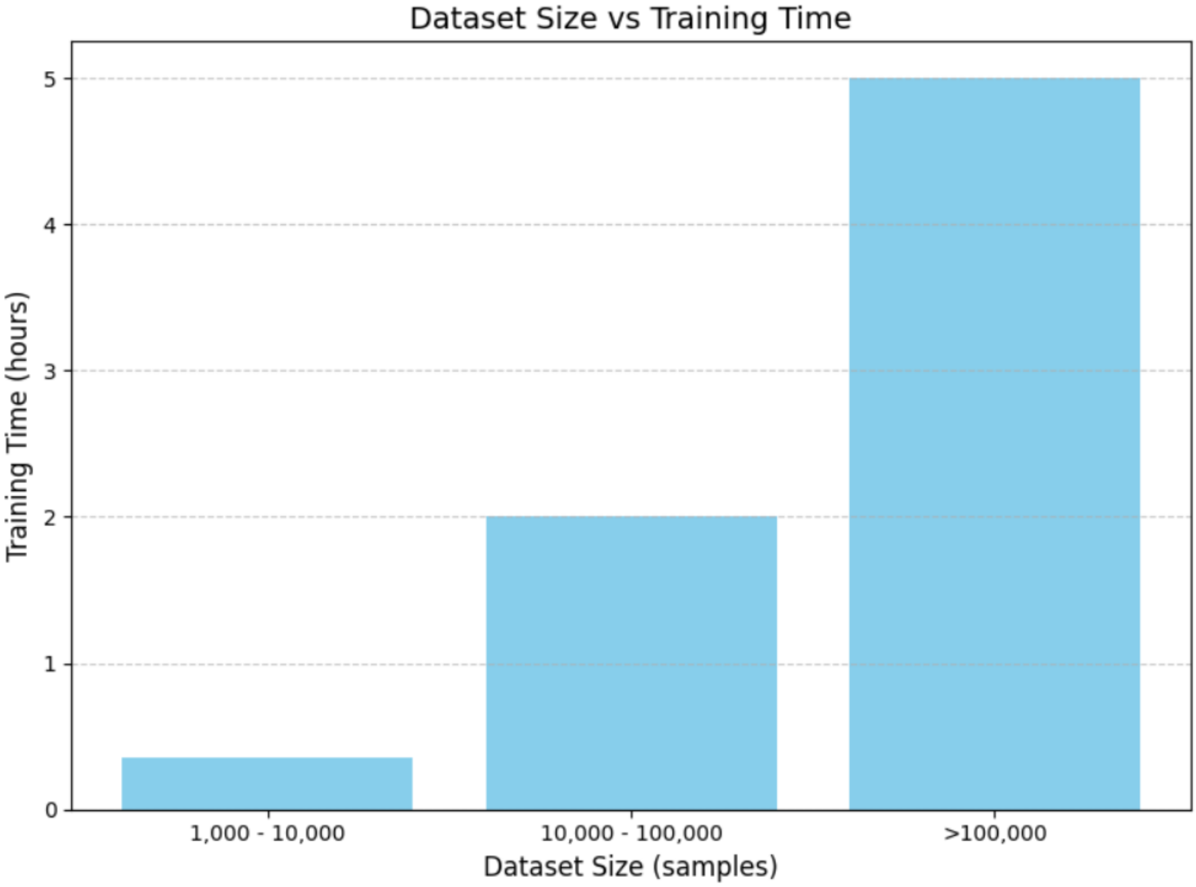


Fig 7.2 Comparison of response time for various dataset size

7.2.3 COMPARISON OF TRAINING EVALUATION

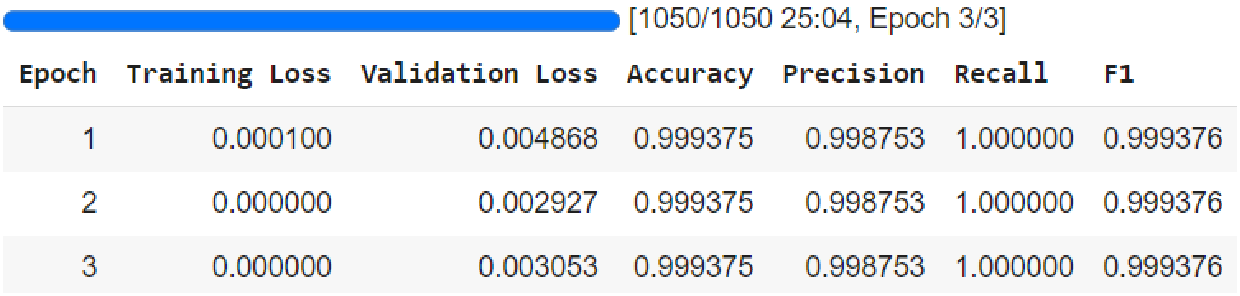
The table presents a **comparison of training evaluation** across two dataset sizes, highlighting the benefits of dynamic layer retention. For smaller datasets containing **1,000–10,000 samples**, retaining layers **4–8** achieves a **98% accuracy** with minimal training time of **0.35 hours** and a response time of **30 ms**, showcasing high efficiency. For larger datasets between **10,000–100,000 samples**, retaining layers **4–10** results in a **99% accuracy**, requiring **2.0 hours** of training and a response time of **50 ms**, balancing accuracy with computational cost.

Dataset Size	Layers Retained	Training Time (hrs)	Response Time (ms)	Accuracy After Reduction
1,000–10,000 samples	Layers 4–8	0.35	30	98%
10,000–100,000 samples	Layers 4–10	2.0	50	99%

Table 7.3 Training evaluation and Accuracy comparison

The comparison demonstrates that increasing dataset size leads to slightly higher training and response times, yet accuracy remains optimal. Layer retention effectively reduces unnecessary computations by retaining only relevant layers while maintaining performance. This approach significantly optimizes the model's training and inference efficiency for varying dataset scales.

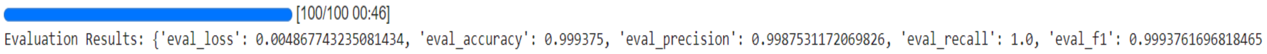
DATASET WITH 1000-10000 SAMPLES TRAINING RESULT



[1050/1050 25:04, Epoch 3/3]

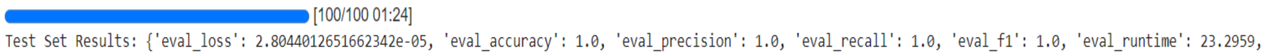
Epoch	Training Loss	Validation Loss	Accuracy	Precision	Recall	F1
1	0.000100	0.004868	0.999375	0.998753	1.000000	0.999376
2	0.000000	0.002927	0.999375	0.998753	1.000000	0.999376
3	0.000000	0.003053	0.999375	0.998753	1.000000	0.999376

Fig 7.3 Training results for smaller datasets with 1000-10000 samples



```
[100/100 00:46]
Evaluation Results: {'eval_loss': 0.004867743235081434, 'eval_accuracy': 0.999375, 'eval_precision': 0.9987531172069826, 'eval_recall': 1.0, 'eval_f1': 0.9993761696818465
```

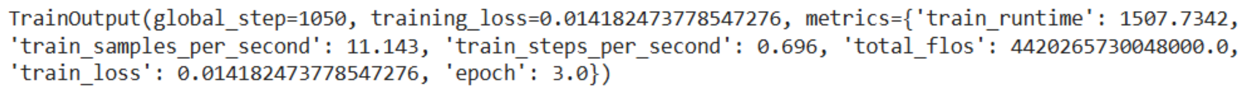
Fig 7.4 Evaluation results for smaller datasets with 1000-10000 samples



```
[100/100 01:24]
Test Set Results: {'eval_loss': 2.8044012651662342e-05, 'eval_accuracy': 1.0, 'eval_precision': 1.0, 'eval_recall': 1.0, 'eval_f1': 1.0, 'eval_runtime': 23.2959,
```

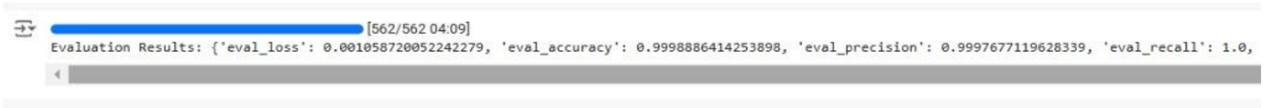
Fig 7.5 Test Results for smaller datasets with 1000-10000 samples

DATASET WITH 10000-1 LAKH SAMPLES



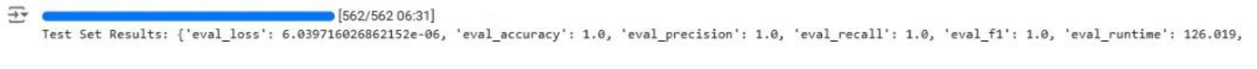
```
TrainOutput(global_step=1050, training_loss=0.014182473778547276, metrics={'train_runtime': 1507.7342,
'train_samples_per_second': 11.143, 'train_steps_per_second': 0.696, 'total_flos': 4420265730048000.0,
'train_loss': 0.014182473778547276, 'epoch': 3.0})
```

Fig 7.6 Training results for larger datasets with 10000-1 lakh samples



```
[562/562 04:09]
Evaluation Results: {'eval_loss': 0.001058720052242279, 'eval_accuracy': 0.9998886414253898, 'eval_precision': 0.9997677119628339, 'eval_recall': 1.0,
```

Fig 7.7 Evaluation results for larger datasets with 10000-1 lakh samples



```
[562/562 06:31]
Test Set Results: {'eval_loss': 6.039716026862152e-06, 'eval_accuracy': 1.0, 'eval_precision': 1.0, 'eval_recall': 1.0, 'eval_f1': 1.0, 'eval_runtime': 126.019,
```

Fig 7.8 Test Results for larger datasets with 10000-1 lakh samples

EVALUATION METRICS COMPARISON

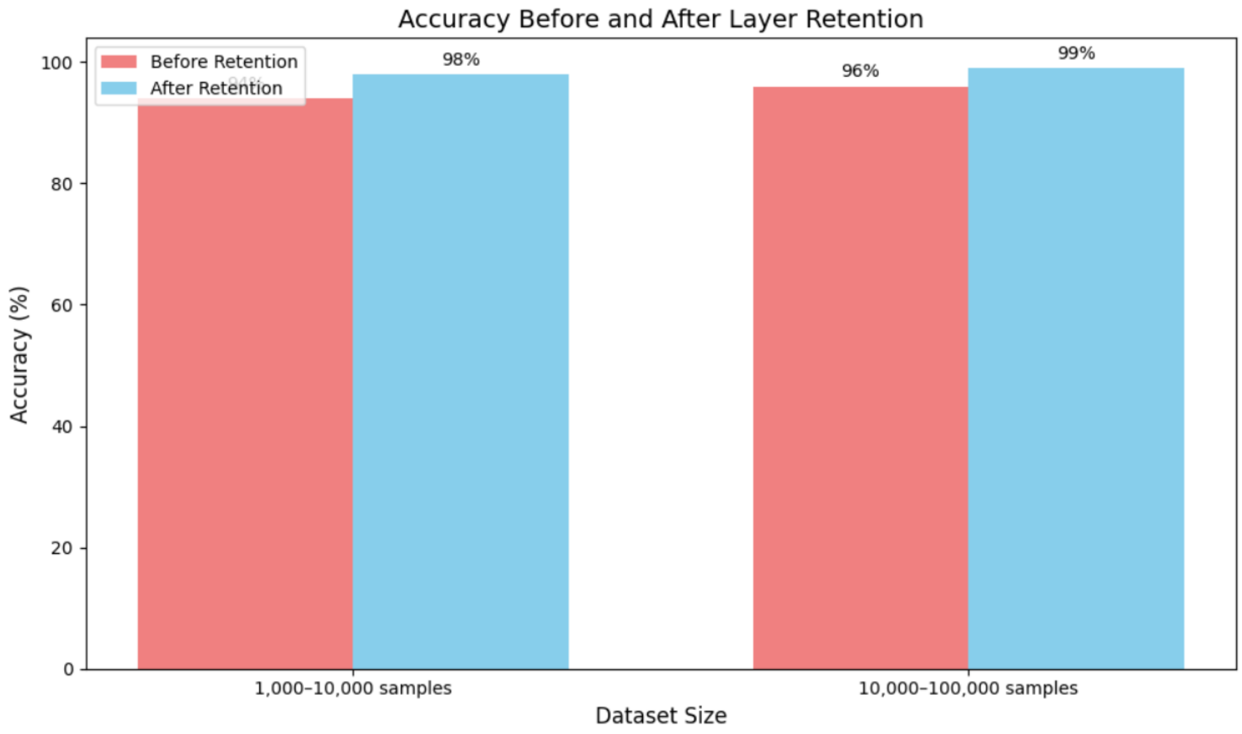


Fig 7.9 Accuracy comparison before and after layer retention

7.3 METHODOLOGY 2: NLP TECHNIQUES

7.3.1 FEATURE EXTRACTION

The system applies a combination of key NLP techniques, including TF-IDF Vectorization, Part-of-Speech (PoS) tagging, Named Entity Recognition (NER), and Sentiment Analysis, to transform raw textual data into meaningful numerical representations.

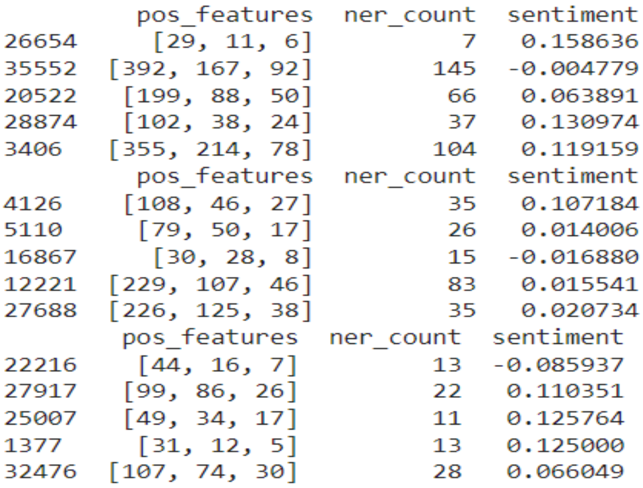
Example:

"Breaking news: Incredible earthquake devastates Japan."

Extracted Features for the above factors:

TF-IDF: breaking	0.7
TF-IDF: news	0.3
TF-IDF: earthquake	0.9
TF-IDF: japan	0.8
TF-IDF: region	0.5
POS: Nouns (NN)	2
POS: Verbs (VB)	1
POS: Adjectives (JJ)	1
NER Count	2
Sentiment Polarity	-0.85

Table 7.4 Feature extraction using NLP techniques



```
          pos_features  ner_count  sentiment
26654      [29, 11, 6]          7   0.158636
35552   [392, 167, 92]        145  -0.004779
20522    [199, 88, 50]         66   0.063891
28874    [102, 38, 24]         37   0.130974
3406    [355, 214, 78]        104   0.119159
          pos_features  ner_count  sentiment
4126     [108, 46, 27]         35   0.107184
5110      [79, 50, 17]         26   0.014006
16867      [30, 28, 8]         15  -0.016880
12221   [229, 107, 46]         83   0.015541
27688   [226, 125, 38]         35   0.020734
         pos_features  ner_count  sentiment
22216     [44, 16, 7]         13  -0.085937
27917    [99, 86, 26]         22   0.110351
25007    [49, 34, 17]         11   0.125764
1377      [31, 12, 5]         13   0.125000
32476   [107, 74, 30]         28   0.066049
```

Fig 7.10 Results of feature extraction for a Dynamic input

7.3.2 LOGISTIC REGRESSION MACHINE LEARNING ALGORITHM

The extracted features (TF-IDF, PoS counts, sentiment scores, and NER outputs) were combined into a single feature matrix to train the **Logistic Regression** model. The Logistic Regression model showed strong validation accuracy, proving its effectiveness in detecting fake news using extracted features. Its simplicity and efficiency made it ideal for handling high-dimensional text data, performing well without overfitting and ensuring reliable predictions. Additionally, the model's interpretability allowed us to identify key features like important words, grammatical structures, and sentiment scores that influenced the classification.

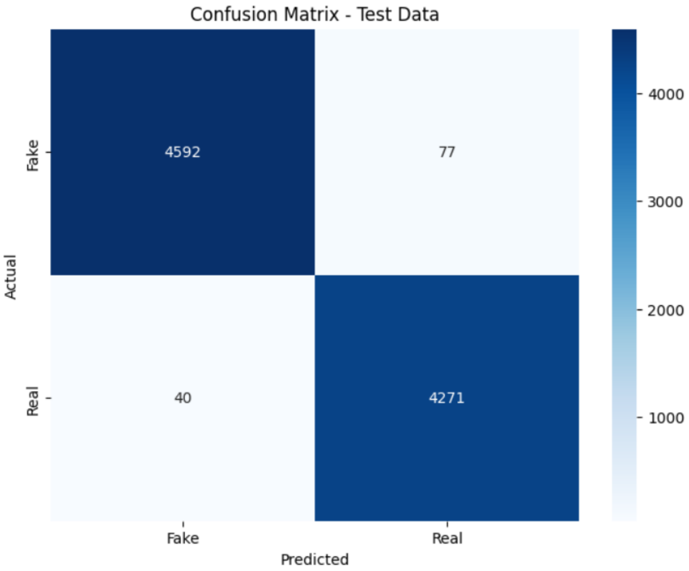
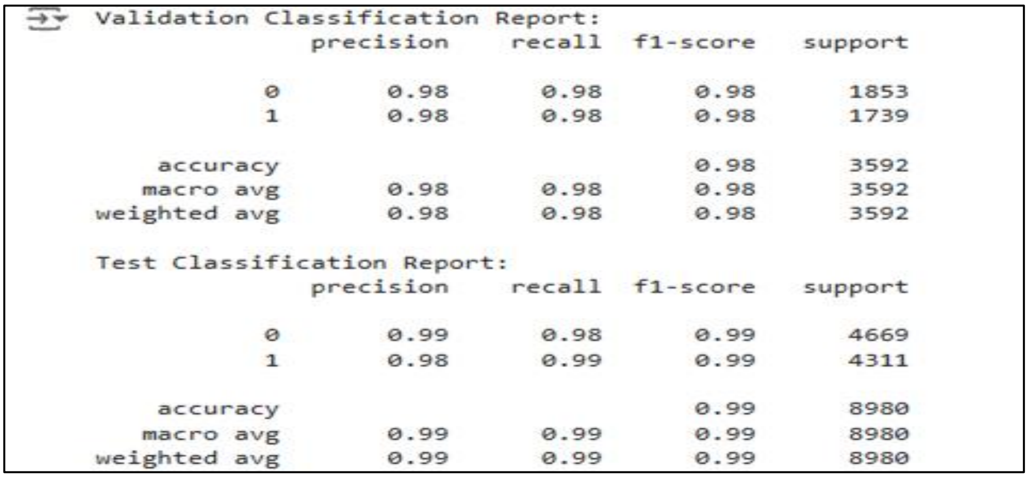


Fig 7.11 Confusion matrix of actual vs predicted labels



```
Validation Classification Report:
              precision    recall  f1-score   support

           0       0.98      0.98      0.98      1853
           1       0.98      0.98      0.98      1739

    accuracy                           0.98      3592
   macro avg       0.98      0.98      0.98      3592
weighted avg       0.98      0.98      0.98      3592

Test Classification Report:
              precision    recall  f1-score   support

           0       0.99      0.98      0.99      4669
           1       0.98      0.99      0.99      4311

    accuracy                           0.99      8980
   macro avg       0.99      0.99      0.99      8980
weighted avg       0.99      0.99      0.99      8980
```

Fig 7.12 Validation and Test classification report of Logistic regression model

7.4 KEY IMPROVEMENTS

The base paper focuses solely on Enhanced BERT, which uses deep learning to classify fake news based on semantic understanding. In contrast, the proposed system introduces two distinct methodologies: RoBERTa-based classification with dynamic layer retention and NLP feature extraction techniques. The RoBERTa model improves over BERT by leveraging advanced pre-training techniques, dynamic masking, and larger training data, while dynamic layer retention significantly reduces computational costs by retaining only the most relevant layers based on dataset size. This enhances both training efficiency and inference speed. Additionally, the NLP-based methodology uses linguistic feature extraction techniques like TF-IDF, PoS tagging, Named Entity Recognition (NER), and sentiment analysis, providing a lightweight and interpretable alternative. These improvements result in better accuracy, reduced computation time, interpretability, and flexibility.

CHAPTER 8

CONCLUSION AND FUTURE WORK

8.1 CONCLUSION

The fake news detection system presented in this research work demonstrates the effectiveness of leveraging advanced Natural Language Processing (NLP) techniques and pre-trained transformer models like RoBERTa for addressing the critical challenge of misinformation. Two distinct approaches were proposed: the first uses the fine-tuned RoBERTa model, while the second employs traditional NLP-based feature extraction techniques such as TF-IDF, Part-of-Speech (PoS) tagging, Named Entity Recognition (NER), and sentiment analysis.

The RoBERTa-based approach efficiently captures deep contextual relationships within textual data by utilizing its multi-layer transformer architecture. The introduction of dynamic layer retention in this model significantly enhances computational efficiency by adjusting the number of layers fine-tuned based on the size of the dataset. For smaller datasets, fewer layers are fine-tuned, reducing computation time, while larger datasets leverage deeper layers for richer context representation. This adaptability ensures that the system is scalable and effective across varying dataset sizes, maintaining high accuracy without unnecessary computational overhead.

The NLP-based approach complements this by extracting linguistic and statistical features to observe patterns in the text. By matching dynamic input patterns against those extracted during training, this method provides an interpretable mechanism to identify fake news based on identifiable textual characteristics. Although computationally simpler than RoBERTa, this approach offers insights into the structural and linguistic aspects of fake news detection.

Together, these methodologies address the problem of fake news detection from complementary perspectives. While the RoBERTa model excels in capturing deep, contextual representations of text, the NLP-based techniques offer an interpretable framework for analysing patterns in news articles. This dual-model strategy ensures robustness, accuracy, and scalability. The system also integrates well with practical implementation environments, using Google Colab for training and Google Drive for model storage. These tools ensure accessibility, ease of collaboration, and the ability to expand upon this work in the future.

In conclusion, the proposed fake news detection system is a comprehensive solution that balances computational efficiency, scalability, and accuracy. The use of advanced LLM, machine learning and NLP techniques ensures the system's relevance in combating misinformation in diverse and evolving contexts.

8.2 FUTURE WORK

The proposed system is implemented using RoBERTa which is primarily designed for classification tasks on a English-language dataset. Future work on the fake news detection system can focus on incorporating the T5 (Text-to-Text Transfer Transformer) model to enhance both accuracy and interpretability. Unlike RoBERTa, T5 reframes fake news detection as a text-to-text task, enabling it to generate labels or even explanations for its predictions, making the system more transparent and user-friendly. Additionally, the system can be expanded to handle multi-lingual data using models like mT5, allowing it to detect fake news across multiple languages, thus increasing its global applicability. Another promising direction is integrating multi-modal analysis by combining text processing with image or video analysis to detect fake news containing manipulated visual content, which is prevalent on social media platforms. These advancements would significantly enhance the system’s robustness and adaptability to real-world misinformation challenges.

REFERENCES

1. Shadi A. Aljawarneh , Safa Ahmad Swedat,” Fake News Detection Using Enhanced BERT”, IEEE Transactions on Computational Social Systems (Volume: 11, Issue: 4, August 2024), DOI: 10.1109/TCSS.2022.3223786

2. 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-19 pandemic: A systematic review", J. Public Health, vol. 31, pp. 1-10, Oct. 2021.

3. R. Baskar, S. Sah, S. R, K. S. Kumar, H. Patil and G. N. Reddy, "Advancements in Fake News Detection: Integrating NLP and Multi-Modal Approaches," 2023 Intelligent Computing and Control for Engineering and Business Systems (ICCEBS), Chennai, India, 2023, pp. 1-5, doi: 10.1109/ICCEBS58601.2023.10448703.

4. A. Matheven and B. V. D. Kumar, "Fake News Detection Using Deep Learning and Natural Language Processing," 2022 9th International Conference on Soft Computing & Machine Intelligence (ISCMI), Toronto, ON, Canada, 2022, pp. 11-14, doi: 10.1109/ISCMI56532.2022.10068440.

5. V. Rana, V. Garg, Y. Mago and A. Bhat, "Compact BERT-Based Multi-Models for Efficient Fake News Detection," 2023 3rd International Conference on Intelligent Technologies (CONIT), Hubli, India, 2023, pp. 1-4, doi: 10.1109/CONIT59222.2023.10205773.

6. A. J. Keya, S. Afridi, A. S. Maria, S. S. Pinki, J. Ghosh and M. F. Mridha, "Fake News Detection Based on Deep Learning," 2021 International Conference on Science & Contemporary Technologies (ICSCT), Dhaka, Bangladesh, 2021, pp. 1-6, doi: 10.1109/ICSCT53883.2021.9642565.

7. A. H. J. Almarashy, M.-R. Feizi-Derakhshi and P. Salehpour, "Elevating Fake News Detection Through Deep Neural Networks, Encoding Fused Multi-Modal Features," IEEE Access, vol. 12, pp. 82146-82159, June 2024, doi: 10.1109/ACCESS.2024.3411926.

8. Y. Liu, W. Bing, S. Ren, and H. Ma, "BC-FND: An Approach Based on Hierarchical Bilinear Fusion and Multimodal Consistency for Fake News Detection," IEEE Access, vol. 12, pp. 62738-62750, May 2024, doi: 10.1109/ACCESS.2024.3392409

9. M. Q. Alnabhan and P. Branco, "Fake News Detection Using Deep Learning: A Systematic Literature Review," IEEE Access, vol. 12, pp. 114435-114450, August 2024, doi: 10.1109/ACCESS.2024.3435497

10. D. Wang, W. Zhang, W. Wu, and X. Guo, "Soft-Label for Multi-Domain Fake News Detection," IEEE Access, vol. 11, pp. 98596-98608, September 2023, doi: 10.1109/ACCESS.2023.3313602

11. N. Seddari, A. Derhab, M. Belaoued, W. Halboob, J. Al-Muhtadi, and A. Bouras, "A Hybrid Linguistic and Knowledge-Based Analysis Approach for Fake News Detection on Social Media," IEEE Access, vol. 10, pp. 62097-62109, June 2022, doi: 10.1109/ACCESS.2022.3181184

12. J. T. H. Kong, W. K. Wong, F. H. Juwono, and C. Apriono, "Generating Fake News Detection Model Using a Two-Stage Evolutionary Approach," IEEE Access, vol. 11, pp. 85067-85079, August 2023, doi: 10.1109/ACCESS.2023.3303321

13. Ravish, R. Katarya, D. Dahiya, and S. Checker, "Fake News Detection System Using Featured-Based Optimized MSVM Classification," IEEE Access, vol. 10, pp. 113184-113196, November 2022, doi: 10.1109/ACCESS.2022.3216892

14. S. K. Hamed, M. J. A. Aziz, and M. R. Yaakub, "Improving Data Fusion for Fake News Detection: A Hybrid Fusion Approach for Unimodal and Multimodal Data," IEEE Access, vol. 12, pp. 112412-112424, August 2024, doi: 10.1109/ACCESS.2024.3443092

15. W. Jian, J. P. Li, M. A. Akbar, A. U. Haq, S. Khan, R. M. Alotaibi, and S. A. Alajlan, "SA-Bi-LSTM: Self Attention With Bi-Directional LSTM-Based Intelligent Model for Accurate Fake News Detection to Ensured Information Integrity on Social Media Platforms," IEEE Access, vol. 12, pp. 48436-48450, April 2024, doi: 10.1109/ACCESS.2024.3382832