Deepfake-News Shield: Detecting False Articles and Manipulated Photos

A major project report submitted in partial fulfilment of the requirement for the award of degree of

Bachelor of Technology

in

Computer Science & Engineering

Submitted by

Madhav, Soha Khan, Harshit Thakur, Divyam Saini

(221030283, 221031049, 221031013, 221030070)

Under the guidance & supervision of

Dr. Deepak Gupta (SG)

Department of Computer Science & Engineering and Information Technology

Jaypee University of Information Technology,

Waknaghat, Solan - 173234 (India)

December 2025

Supervisor's Certificate

This is to certify that the major project report entitled 'Deepfake-News Shield: Detecting False Articles and Manipulated Photos, submitted in partial fulfilment of the requirement for the award of the degree of **Bachelor of Technology** in **Computer Science & Engineering,** in the Department of Computer Science & Engineering and Information Technology, Jaypee University of Information Technology, Waknaghat, is a bonafide project work carried out under my supervision during the period from July 2025 to December 2025.

I have personally supervised the research work and confirm that it meets the standards required for submission. The project work has been conducted in accordance with ethical guidelines, and the matter embodied in the report has not been submitted elsewhere for the award of any other degree or diploma.

(Supervisor Signature)

Supervisor Name: Dr. Deepak Gupta

Date: 30 Sept 2025

Designation: Assistant Professor (SG)

Place:

Department: Dept. of CSE & IT

Candidate's Declaration

We hereby declare that the work presented in this major project report entitled **'DeepfakeNews Shield: Detecting False Articles and Manipulated Photos',** submitted in partial fulfilment of the requirements for the award of the degree of **Bachelor of Technology in Computer Science & Engineering,** in the Department of Computer Science & Engineering and Information Technology, Jaypee University of Information Technology, Waknaghat, is an authentic record of our own work carried out during the period from July 2025 to December 2025 under the supervision of **Dr. Deepak Gupta.**

We further declare that the matter embodied in this report has not been submitted for the award of any other degree or diploma at any other university or institution.

Madhav	Soha Khan
221030283	221031049
Date:	Date:

Harshit Thakur	Divyam Saini
221031013	221030070
Date:	Date:

This is to certify that the above statement made by the candidates is true to the best of my knowledge.

(Supervisor's Signature)
Supervisor Name: Dr. Deepak Gupta
Assistant Professor (SG)
Department: Dept. of CSE & IT

Date:
Place:

Acknowledgement

Firstly, I express my heartiest thanks and gratefulness to almighty God for His divine blessing makes it possible for us to complete the project work successfully.

We are grateful and deeply indebted to our supervisor Dr. Deepak Gupta, Assistant Professor (SG), Department of Computer Science & Engineering and Information Technology, Jaypee University of Information Technology, Waknaghat, for his invaluable guidance throughout this project. His profound knowledge and keen interest in big data analytics, cybersecurity, machine/deep learning, and programming languages greatly motivated us to carry out this work. His endless patience, scholarly guidance, continual encouragement, energetic supervision, constructive criticism, and valuable advice—together with his effort in reviewing and correcting our drafts at every stage—made the successful completion of this project possible.

We would also generously welcome each one of those individuals who have helped us directly or in a roundabout way in making this project a win. In this unique situation, we might want to thank the various staff individuals, both educating and non-instructing, who have developed their convenient help and facilitated our undertaking.

In closing, we wish to recognize and appreciate the enduring support and patience of our parents. Their unwavering encouragement has been a source of strength throughout this journey.

With gratitude,

Madhav (221030283)
Soha Khan (221031049)
Harshit thakur (221031013)
Divyam Saini (221030070)

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

Abbreviation	Meaning
IoT	Internet of Things
BERT	Bidirectional Encoder Representations from Transformers
CNN	Convolutional Neural Network
GAN	Generative Adversarial Network
ViT	Vision Transformer
DFDC	Deepfake Detection Challenge
LSTM	Long Short-Term Memory
Bi-LSTM	Bidirectional Long Short-Term Memory
GRU	Gated Recurrent Unit
RNN	Recurrent Neural Network

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Nowadays, misleading information represents a substantial risk because false information and deepfake images can spread faster than they can be validated. This material can shape public opinion, help spark unrest, and destroy reputations. Old methods used for detection frequently fail against more sophisticated methods.

This project successfully develops text-based misinformation detection while also laying groundwork for a future multi-modal system. Currently, for text-based information, we first cleaned and preprocessed the IFND dataset using unique and customised steps. We then proceeded to follow a structured pipeline, firstly training and testing the dataset with traditional deep learning models such as LSTM and BiLSTM, then implementing hybrid and attention-based RNN architectures including stacked, residual and GRU-enhanced variants and finally advancing to transformer-based models along with hybrid transformer-RNN combinations. This organised way of moving from traditional to advanced models enabled us to provide detailed performance comparisons to identify the best model that could classify fake news.

We will then further expand this system to deepfake image detection using ViT and other advanced architectures capable of identifying inconsistencies. Additionally, a web-based interface built using Flask and Streamlit will be developed in the later stages of the project as this interface will allow users to detect fake text and images using visual analytics. A MongoDB database is also planned for storing and logging queries as part of a future framework .

In conclusion, the project provides a strong foundation for a comprehensive misinformation detection system. Its design incorporates ongoing learning, new modal integration and future features like social media input, multilingual support, and also ensures long-term adaptability and reliability in fighting the evolving misinformation and deepfake threats.

Chapter 01: INTRODUCTION

1.1 INTRODUCTION

We live in a world where information is just a click away. News spreads instantly through social media, websites, and online platforms. While this has made communication faster and easier, it has also created a serious problem — the rise of misinformation. From fake news stories designed to mislead people to deepfake images that look real but are completely fabricated, misinformation is shaping public opinion, creating confusion, and in some cases even causing harm.

The biggest challenge is that misinformation today is much harder to detect than before. Fake news articles often look like genuine reports, and deepfakes created with advanced AI tools can mimic real people with stunning accuracy. Most existing solutions deal with only one side of the problem, either fake news or deepfakes, but very few combine both. This gap makes it easier for misinformation to slip through and reach large audiences.

Our project aims to tackle this issue by building a multi-modal misinformation detection system. For the text part, we began by preprocessing our IFND dataset, then moved forward with a well thought pipeline of models: starting with traditional architectures (LSTM, BiLSTM), proceeding to hybrid and attention-based RNN methods and finally evaluating with transformer-based models along with combined transformer-RNN approaches. In the future phase of this project, we will extend the system to deepfake image detection using advanced and recent architectures and integrate the Google FactCheck API for claim verification along with a simple web interface to allow users to test text or images and view meaningful visualizations like word clouds, sentiment charts etc.

The goal of this project is not only to detect misinformation but also to make the process transparent, reliable, and user-friendly. By giving people tools to verify information for themselves, we hope to raise awareness and reduce the harmful effects of misinformation in everyday life.

1.2 PROBLEM STATEMENT

In today's digital era, the rapid spread of misinformation poses a serious challenge to society. Fake news articles, misleading claims, and manipulated media such as deepfake images circulate widely across social platforms, influencing public opinion, creating panic, and sometimes leading to harmful consequences. With the growing sophistication of text generation and image manipulation tools powered by artificial intelligence, it has become increasingly difficult for individuals to distinguish between authentic and fabricated information. While several research efforts have focused on
fake news detection using Natural Language Processing (NLP) and deepfake detection using Deep Learning (DL), most of these approaches address these problems separately. This lack of integration leaves a significant gap in building robust, realworld systems capable of detecting multiple forms of misinformation together.

The problem becomes more severe considering that misinformation does not exist in a single form. A fake article may contain manipulated claims, while an accompanying image or video may be synthetically generated, further reinforcing the false narrative. The absence of a multi-modal detection system makes it easier for misinformation to bypass detection filters, spread rapidly, and cause widespread impact. Moreover, the need for real-time claim verification through fact-checking services is still underdeveloped in current solutions. Therefore, there is an urgent requirement for a unified platform that can simultaneously analyze text for fake news, evaluate images for deepfakes, and integrate fact-checking APIs for validating claims against trusted sources. Such

a system would not only strengthen digital media literacy but also provide an effective tool to mitigate the societal risks of misinformation.

1.3 OBJECTIVES

1.3.1 To design and develop a machine learning framework for misinformation detection by combining Natural Language Processing (NLP) for fake news detection and Deep Learning models (Vision Transformer) for deepfake image analysis.

1.3.2 To provide real-time claim verification by integrating the Google FastCheck API and validating news content against trusted fact-checking sources.

1.3.3 To collect, preprocess, and integrate multi-model datasets (textual news articles, claims, and images) to create a unified system capable of handling both text-based and image-based misinformation.

1.3.4 To make all components that are available in interactive web applications using Streamlit and Flask.

1.4 SIGNIFICANCE AND MOTIVATION OF THE PROJECT WORK

Misinformation has become one of the biggest challenges of our digital era. With the rapid rise of social media platforms and online news outlets, false information— whether in the form of fake news articles or manipulated images—spreads faster than ever before. This not only misleads the public but also threatens trust in the media, fuels social unrest, and even impacts democratic decision-making.

The motivation behind this project comes from the urgent need to create a reliable system that can detect misinformation in both text and image formats. Traditional methods often focus on just one type of data—either news text or images—but misinformation today is multi-dimensional. By combining Natural Language Processing (NLP) for detecting fake news with Vision Transformers for analyzing deepfake images, this project aims to build a more comprehensive and effective framework.

The significance of this work lies in its potential to contribute to a safer digital environment. By enabling real-time claim verification through trusted fact-checking sources and integrating diverse datasets, this system can serve as a practical tool to help individuals, media organizations, and policymakers combat the spread of false information. Ultimately, the project is motivated by the vision of creating a digital ecosystem where information can be trusted, and truth has a stronger voice than misinformation.

Moreover, this project is not only academically significant but also practically relevant in today's world. As misinformation continues to evolve in sophistication, our system can serve as a foundation for future research and real-world applications. It has the potential to be integrated into news portals, social media platforms, and fact-checking tools, making it a scalable solution with long-term impact. By combining advanced AI with societal responsibility, the project aspires to bridge the gap between technology and trust in the digital age.

1.4.1 Significance of the project:

1. Provides a multi-modal detection system that can analyze both text (fake news) and images (deepfakes), making it more reliable than single-source approaches.

2. Enhances trust in media and online content by offering real-time claim verification against authentic fact-checking sources.

3. Can be used by individuals, journalists, and policymakers to quickly identify and counter false information.

4. Contributes to building a safer and more informed digital environment by reducing the harmful effects of misinformation.

1.4.2 Motivation for the project

1. The increasing spread of fake news and deepfakes inspired the need for a strong solution that addresses both simultaneously.

2. Traditional detection systems usually focus only on text or images, which motivated us to create a unified framework that handles both.

3. The misuse of advanced AI tools for generating realistic fake images and misleading content highlights the urgency to develop AI-powered countermeasures.

4. The project is driven by the vision of protecting truth and reliability in today's digital communication space.
5. A strong motivation comes from the social impact—helping people make better-informed decisions by ensuring they can trust the information they consume.

1.5 ORGANIZATION OF PROJECT REPORT

This report is organized into 3 main chapters, with each section providing a detailed description of different aspects of the project work.

1.5.1 CHAPTER 01:

1. Introduction: Provides background information on the growing issue of misinformation in the digital era, highlighting the spread of fake news and deepfake images, and introduces the concept of an AI-driven system for detection.

2. Problem Statement: Defines the key problems the project aims to address, including the limitations of existing detection methods and the urgent need for a unified, multi-modal approach.

3. Objectives: Outlines the main goals of the project, such as detecting fake news using DL, Transformer and hybrid-based models, analyzing deepfake images with Vision Transformers, integrating Google FactCheck API for real-time claim verification, and combining multi-modal datasets.

4. Significance and Motivation of the Project Work: Explains the practical importance of tackling misinformation, its societal impact, and the motivation behind building a system that enhances trust in digital information.

5. Organization of Project Report: Provides a roadmap of the chapters and sections.

1.5.2 CHAPTER 02:

1. Overview of Relevant Literature: Reviews existing work on fake news detection, deepfake image analysis, and fact-checking systems, offering insights into different machine learning and deep learning models applied in this field.

2. Key Gaps in the Literature: Identifies shortcomings in current solutions, such as their limited ability to process both text and images simultaneously, lack of real-time claim verification, and dataset constraints. These gaps provide opportunities for innovation in this project.

1.5.3 CHAPTER 03:

1. Requirements and Analysis: Specifies the requirements of the system including tools, libraries, APIs, and datasets, along with functional and non-functional specifications.

2. Project Design and Architecture: Explains the system architecture, describing how DL, Transformer and hybrid-based models are implemented for the textual part of the project.

3. Implementation: Provides details on the implementation process, including dataset collection and pre-processing, training models, and integrating different modules into the complete system.

1.5.4 CHAPTER 04

1. Testing strategy: discusses the testing strategy planned and implemented to make sure the model produces valid results. It includes various steps like unit testing, integration testing, etc.

2. Test Cases and Outcomes: describes how the system was tested to check if it works correctly. Each test case provides an input, the expected result, and the actual output.

1.5.5 CHAPTER 05:

Displays results achieved after training different models and approaches on the cleaned IFND dataset, it also provides comparison between models so that we can choose the best performing model.

1.5.6 CHAPTER 06:

1. Conclusion: this chapter summarizes the entire project, highlighting key results and explains achievements and reflects if the objectives were met.
2. Future Scope: this section explores possible improvements, advanced technology integrations and future upgrades through which our project can be expanded.

Chapter 02: Literature Survey

2.1 Feasibility Study

Deepfake-News Shield's implementation is achievable given the availability of inexpensive and open-source machine-learning frameworks, natural language processing (NLP) services, and image forensics datasets, such as FaceForensics++. Technically, previously implemented models like convolutional neural networks (CNN), recurrent neural networks (RNN), and transformers can be successfully applied to detect fake articles or media that are manipulated. Economically, the use of open source libraries as well as academic resources the cost to implement Deepfake-News Shield will be significantly reduced. Socially, Deepfake-News Shield is very relevant and covers issues of public concern, namely, the rising prevalence of misinformation and the manipulation of media that is damaging or harmful to society, political parties, politicians, etc.

Table 2.1: Literature Survey

S.No.	Title	Work done	Pros	Cons
1.	A Comprehensive Survey on Fake News Detection Using Machine Learning (2021) [1]	Reviews ML and DL methods for fake news detection, covering NLP-based feature extraction, contextual cues, datasets, and evaluation metrics.	Broad overview, multimodal focus, identifies gaps.	Only survey, no implementation, lacks validation.

2.	FaceForensics++: Learning to Detect Manipulated Facial Images (2019) [2]	Introduces FaceForensics++ dataset with 1.8M manipulated images, proposes a benchmark, and develops a CNN-based detection pipeline.	Large dataset, standard benchmark, high detection.	Face-only, drops under compression, costly training.
3.	Multimodal Fake News Detection: A Survey of Text and Visual Content Integration Methods (2022) [3]	Surveys multimodal detection combining text and images using deep learning, explains feature extraction and fusion strategies.	Covers multimodal, fusion boosts accuracy.	Costly model, weak generalization
4.	A robust ensemble model for Deepfake detection of GANgenerated images on social media (2021) [4]	Proposed VOTSTACK, an ensemble using Decision Tree, Logistic Regression, and SVM with Voting + Stacking, plus PCA-based preprocessing, achieving ~91.6% accuracy.	High accuracy, robust, scalable.	Heavy computation, not real-time.
5.	Robust manipulated media localization and detection based on high frequency and texture features (2022) [5]	Introduced RMLD-HFTF, combining frequency + texture features with attention and encoder/decoder structure for detection and localization of manipulations.	Detects & localizes edits, cross-dataset robust.	Complex model, high resource needs.

6.	Deepfake Image Detection using Vision Transformer Models (2023) [6]	Implemented a Vision Transformer (ViT) on 40,000 Kaggle images (20k real, 20k fake) to classify deepfakes, achieving 89.91% accuracy. Compared ViT performance with other detection methods.	High accuracy, fast convergence, scalable for large inputs, robust detection.	Needs large datasets, overfits on small data, computationally heavy, limited generalization.
7.	Detection of Fake News Using Machine Learning and Natural Language Processing Algorithms (2021) [7]	Developed a fake news detection system using ML (LR, SVM, DT, NB), DL (LSTM), and BERT on 26k news articles. BERT achieved the highest accuracy of 98%.	Using multiple models, strong preprocessing, BERT reached 98% accuracy.	Dataset-limited, resourcedemanding , weaker ML models underperform.
8.	Enhancing Deepfake Detection: A Multimodal Approach for Improved Accuracy (2022) [8]	Proposed a multimodal deepfake detector leveraging blur, residual noise, and facial warping artifacts. Combined visual, noise, and landmark cues for better accuracy.	Multimodal features improve accuracy, temporal inconsistencies captured, robust detection.	Complex systems require large datasets, high computational cost.
9.	The New Paradigm of Deepfake Detection at the Text Level (2020) [9]	Explored ML/DL methods for deepfake detection using CNNs, RNNs, and preprocessing techniques. Evaluated models on public datasets with accuracy, precision, recall, and F1score.	Covers multiple ML/DL approaches, strong preprocessing, good benchmark comparisons.	Dependent on dataset quality, high computation, limited generalization to new manipulations.

10.	Deepfake detection using deep learning methods: A systematic and comprehensive review (2021) [10]	Conducted a survey of DLbased deepfake detection across images, videos, and audio. Reviewed datasets, key models (CNN, RNN, GAN, hybrids), and highlighted challenges and research gaps.	Comprehensive coverage, clear taxonomy, advanced techniques explained, identifies gaps.	Focused only on recent works, DLheavy bias, requires large datasets, limited real-world validation.
11.	Deepfake Detection Challenge Dataset (2020) [11]	Created the DFDC dataset using diverse generation techniques (GAN, Deepfake, non-learned) and organized a benchmark challenge for detection models.	Large diverse dataset, boosted research on detection algorithms.	Weak generalization, models rely on dataset artifacts.
12.	Deep Fake: An Overview (2021) [12]	Reviewed deepfake techniques and security risks; proposed ECC + DNA-based encryption for securing IoT devices.	Highlights security threats, efficient ECCbased solution.	Focus on security not detection, weak generalization, vulnerable to attacks.
13.	DeepFakes and Beyond: A Survey of Face Manipulation and Fake Detection (2020) [13]	Surveyed face manipulation methods (identity, expression, attribute) and datasets like FaceForensics++ & DeepFakeTIMIT, summarizing key detection approaches.	Clear taxonomy, useful reference for researchers.	Uses older datasets, limited multimodal coverage.

14.	GazeForensics: DeepFake Detection via Gaze-guided Spatial Inconsistency Learning (2023) [14]	Introduced eye-gaze inconsistency as a deepfake cue, combining gaze estimation with CNN features; tested on FaceForensics++ and DFDC datasets.	Novel biometric-based method, better than baseline CNNs.	Drops on compressed videos, datasetspecific limitations.
15.	Detection of Fake News Using Deep Neural Networks (2022) [15]	Compared DNN, LSTM, BERT, and Hybrid LSTMBERT for fake news detection on SBFN dataset, finding BERT most effective.	Strong text-based detection, BERT, shows high accuracy.	Text-only focus, ignores nontextual features, limited scope.
16.	The Emergence of Deepfake Technology: A Review (2019) [16]	Analyzed 84 articles (2018–2019) covering deepfake uses, threats, and countermeasures across politics, business, and society.	Highlights positive applications in entertainment, healthcare, education, advertising, and AI innovation.	Exposes risks to democracy, media trust, and security, enabling disinformation and cybercrime.
17.	SpotFake: A Multimodal Framework for Fake News Detection (2019) [17]	Proposed SpotFake, combining BERT for text and VGG-19 for images, tested on Twitter and Weibo datasets.	Standalone classifier, simpler design, outperforms prior models on multiple datasets.	Limited on long articles, the fusion method is basic, leaving scope for improvement.

18.	Fake News Detection After LLM Laundering: Measurement and Explanation (2025) [18]	Evaluated fake news detectors against LLMparaphrased text, analyzing weaknesses and detection failures.	Comprehensive study, explains failures via sentiment shifts, and releases useful datasets.	Detectors weaker on LLM fakes, Pegasus hardest to detect, LLMbased detectors struggle with self-generated text.
19.	Enhanced deepfake detection with DenseNet and Cross-ViT (2025) [19]	Introduced hybrid DenseNet+Cross-ViT with a voting mechanism for multiface detection in videos.	Achieved nearperfect AUC and F1 on DeepForensics 1.0 and strong results on CelebDF.	Training time is high, resourceintensive, generalization to unseen deepfakes uncertain.
20.	DeepFakes and Beyond: A Survey of Face Manipulation and Fake Detection (2020) [20]	Surveyed face manipulation techniques and detection, categorizing synthesis, identity swap, attribute, and expression manipulation.	Useful for industries like gaming, films, cosmetics, and 3D modeling.	Enables misinformation, fake news, identity fraud, and harmful content like fake profiles or porn.

2.2 Problem Definition

In the current digital age, the proliferation of misinformation and manipulated media presents a significant risk to individuals, organizations, and society as a whole. Fake news articles can wrongly inform readers, sway public opinion, and disrupt social cohesion. Similarly, manipulated photographs and videos (deepfaking) are utilized to defame individuals, con people, or share propaganda.

Traditional approaches to fact-checking and verification of media are largely manual, and therefore slow, and not equipped to cope with the sheer volume of content available online. Additionally, any form of detection often fails when faced with cleverly executed deep-learning manipulations or deceptive narratives in text.

This emphasizes the critical need for an intelligent, automated, and scalable solution to be able to detect false news articles as well as manipulated images quickly and in real time in order to protect digital trust and secure the responsible sharing of fact-based information.

2.3 Problem Analysis

1. The rapid spread of false information: Social media platforms spread false information at extremely high speeds, reaching millions of users before factcheckers can act.

2. Advances in deepfake technology: Sophisticated AI techniques can produce extremely realistic-looking images/videos that are difficult for humans to spot.

3. Limitations of manual fact-checking: Human fact-checking is slow and laborious and cannot keep pace with the amount of information on the internet.

4. Challenges of datasets: There are few reliable, large-scale datasets of both fake/real news and fake images.

5. Cross-domain limitations: Detection processes trained to detect misinformation in a particular language, culture, or on a platform often do not translate to others.

6. Trust and interpretability: AI models should be used as "black boxes" that on the premise of little or no explainability can make predictions.

2.4 Solutions

To tackle these issues, we introduce Deepfake-News Shield, an innovative machine learning mechanism that can discover fake news articles and altered images. This proposed system will combine Natural Language Processing (NLP) and Computer Vision (CV) technologies to present a comprehensive problem-solving measure for multimedia misinformation.
Some key features include:

1. Fake News Detection (Textual Analysis): Implementation of NLP techniques (TF-IDF, embeddings, transformers) which will be applied to examining the article's content, writing style, and the credibility of its source.

2. Manipulated Image Detection (Visual Analysis): Utilization of CNN-based deep learning models trained on datasets like FaceForensics++ for photo manipulation detection.

3. Aggregate ML Models: Usage of ensemble learning combined with deep learning to improve the overall accuracy in terms of detecting text and images as fake.

4. Explainability: Added interpretable AI methods to focus on suspicious words, phrases, or image regions to build user trust.

5. Centralized Dashboard: Simple interface to provide real-time monitoring with detection reports.

6. Scalability and Automation: The proposed system is designed for the collection of a copious amount of social media/article collections, and to adapt to new manipulative behaviors.

Chapter 03: System Development

3.1 REQUIREMENTS and ANALYSIS

3.1.1 Functional Requirements

1. Data Collection & Preprocessing: To establish a diverse dataset, the system will acquire news articles and images from reliable and unreliable sources. All text data will be cleaned, tokenized, and normalized, while all image data will be resized and augmented to enhance robustness and accuracy.

2. Fake News Detection Module: The system will employ supervised ML models, such as Logistic Regression, Random Forest, and BERT, on labeled datasets (real news and fake news). The module will ideally classify articles as real articles or fake articles with high accuracy.

3. Manipulated Image Detection Module: Image data will be analyzed using CNN architectures, specifically XceptionNet and ResNet. The system will classify images as original images or manipulated images, where applicable, and indicate whether or not the original images have been tampered with.

4. Ensemble and Hybrid Integration: The predictions made from the text and image modules will be combined using ensemble

methods such as voting and stacking to increase reliability and robustness.

5. User Dashboard: A simple dashboard will give real-time results along with confidence scores alongside summaries of articles. Users will also be able to upload articles or images for verification by the system. The user dashboard will maintain a log of previous article and image detection activities made by the system for future review.

6. Reporting and Analytics: The outcomes of all detections made by the system will be retained for future analytic endeavors. The system will provide reporting on detection performance indicating the systems overall accuracy levels, types of errors, and trends of detected articles or images for future improvements.

3.1.2 Non-Functional Requirements

1. Scalability: The ability to accommodate sizable datasets and real-time inputs from various sources.

2. Accuracy & Reliability: Achieve high precision and recall in fake detection, even with noise/compression.

3. Real-Time Responsiveness: Provide low-latency detection for quickmoving social media activities.

4. Explainability: Supply interpretable results to increase user confidence in predictions made by the system.

5. Maintainability: Modular architecture that allows the model to be easily retrained through new datasets.

6. Security & Privacy: Secure sensitive user-uploaded data and ensure the ethical use of the detection results.

7. Robustness: Tolerant to adversarial manipulation of photos, dataset bias, and evolving deepfake alterations.

3.1.3 Technical Requirements (Hardware)

1) Development Environment

1. CPU: A modern quad-core or octa-core processor with a clock speed of at least 3.0GHz is required to handle compiling, debugging, and running multiple tools smoothly.

2. Ram: At least 16GB of RAM is recommended so that development tools, IDEs, and testing applications can run together without performance issues.

3. Storage: Either HDD or SSD can be used, but SSD is preferred for speed. A minimum of 500GB free space is needed for projects, dependencies, and related files.

4. OS: Compatible with macOS, Windows, or Linux, depending on developer preference and project requirements.

2) Production Environment

1. CPU: Quad-core or dual-core processor with a clock speed of at least

2.0GHz.

2. Ram: At least 4GB RAM is required to ensure stable operation and smooth handling of applications.

3. Storage: HDD or SSD with at least 100GB free space to store application files, logs, and updates.

4. OS: Supports macOS, Windows, or Linux as per deployment requirements.

5. Network: Minimum bandwidth of 10–15 Mbps is required for reliable connectivity and stable production performance.

3.1.4 Flow Chart

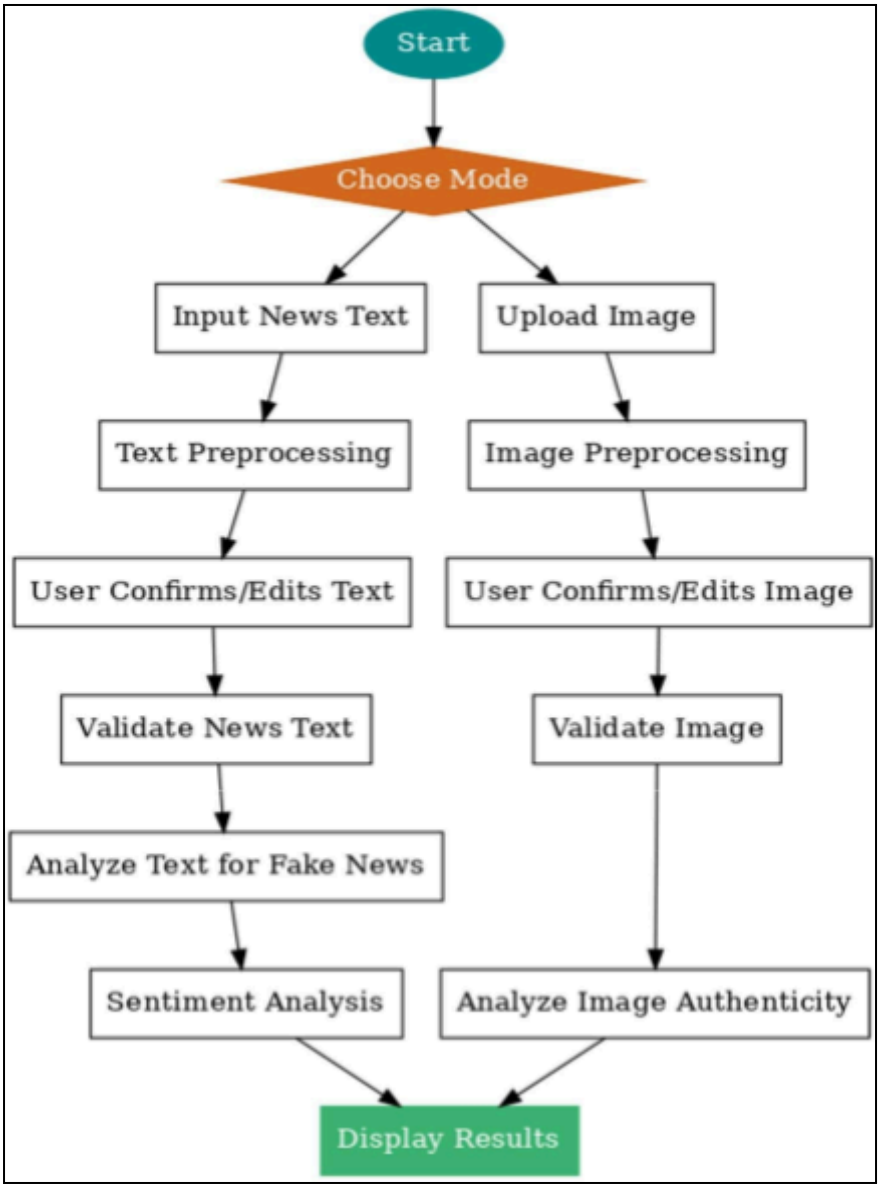
The flowchart demonstrates the functioning of the Deepfake-News Shield system. The first step in the process is for the user to select a type of input—a news story or an image.

1. In the case of the Fake News Detection path, first, the input text is preprocessed, validated, and then categorically analyzed using machine-learning models to classify the article as real or fake. Additionally, sentiment analysis can also lend more reliability in categorizations.

2. In the Manipulated Image Detection path, the uploaded image again goes through a process of preprocessing and validation, followed by analysis with deep-learning models to determine it to be authentic or manipulated.

3. Finally, both branches converge at the Display Results stage, where the system provides the outcome along with confidence scores for the user.

This ensures a unified and reliable solution for detecting both false news content and manipulated images.



[Fig. 3.1. Flow Chart of Deepfake-News Shield]

3.2 Project Design and Architecture

The project Deepfake-News Shield: Detecting False Articles and Manipulated Photos is designed to address the growing challenge of misinformation and manipulated media circulating on online platforms. The current system architecture follows a modular, scalable and multi-stage design focusing primarily on the textual part of the project while deepfake image detection and the web interface remain planned for future development.

The architecture is divided into three primary modules:

1. Fake News Detection(Completed):
   Input text undergoes preprocessing (cleaning, tokenization, and feature extraction) followed by classification through progressive, procedural pipeline of models moving from traditional > hybrid > transformer > transformer+hybrid systems. The output is a prediction label indicating whether the news content is real or fake.
   Models Implemented:
   A. Baseline Deep Learning Models
      i. LSTM
      ii. Bi-LSTM
   B. Hybrid & Attention-Based Models
      i. Attention + LSTM
      ii. Attention + Stacked LSTM
      iii. Self-Attention + LSTM
      iv. Attention + BiLSTM
      v. MHA + BiLSTM
      vi. MHA + BiLSTM + Residual
      vii. Residual BiLSTM
      viii. Scaled Dot-Product MHA + Residual
   C. RNN-GRU Hybrid Experiments

i. GRU on LSTM

ii. Stacked GRU on LSTM

iii. Bidirectional GRU on LSTM

iv. Standard Stacked RNN on LSTM

v. Stacked Bidirectional RNN on LSTM

vi. Mixed Stacked RNN on LSTM

D. Hybrid Models on BiLSTM

i. GRU on BiLSTM

ii. Stacked GRU on BiLSTM

iii. Standard Stacked RNN on BiLSTM

iv. Bidirectional GRU on BiLSTM

v. Stacked Bidirectional RNN on BiLSTM

vi. Mixed Stacked RNN on BiLSTM

vii. BiLSTM + Attention + 1layer GRU

viii. BiLSTM + Attention + 2layer/Stacked GRU

E. Advanced Transformer Models

i. RoBERTa

ii. DeBERTa

iii. T5 (Text-to-Text Transformer)

F. Fusion Model

i. RoBERTa + (BiLSTM + Attention + 1-Layer GRU)

All these models were evaluated using Accuracy, ROC-AUC and F1-Score and the best performing ones form the foundation of the final text-detection pipelines.

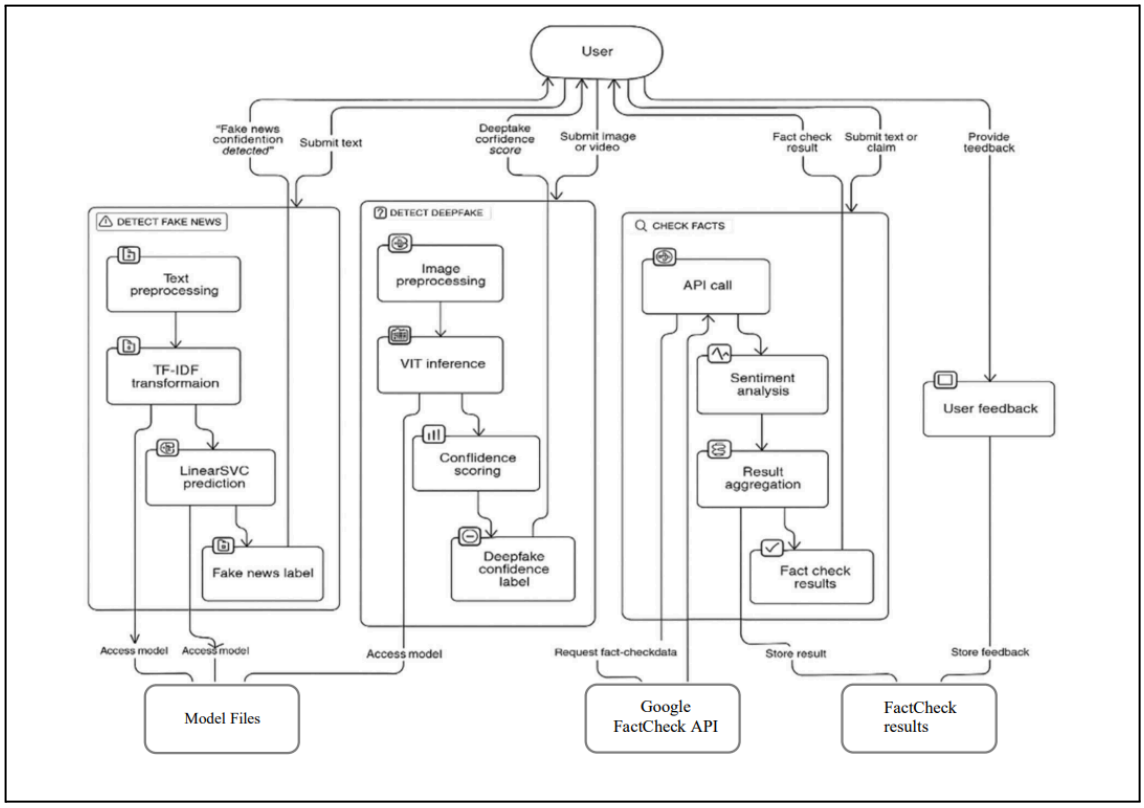
2. Manipulated Face Detection(Future Work):

   Uploaded images will be preprocessed and analyzed using deep learning-based models such as Vision Transformers (ViT) or CNN variants. The system will generate a confidence score and label the image as original or manipulated.

3. Fact-Checking & Feedback(Future Work):

A fast-checking component integrates external APIs (e.g., Google FactCheck API) to cross-verify claims, while user feedback is stored for further improvement of the system.

The overall architecture ensures clarity, modularity, and scalability, allowing future integration of more advanced detection models and larger datasets. The flowchart below illustrates the interaction between modules, starting from user input to final result display, emphasizing both automation and user transparency.



[Fig. 3.2. Project Design and Architecture]

3.3 Data Preparation

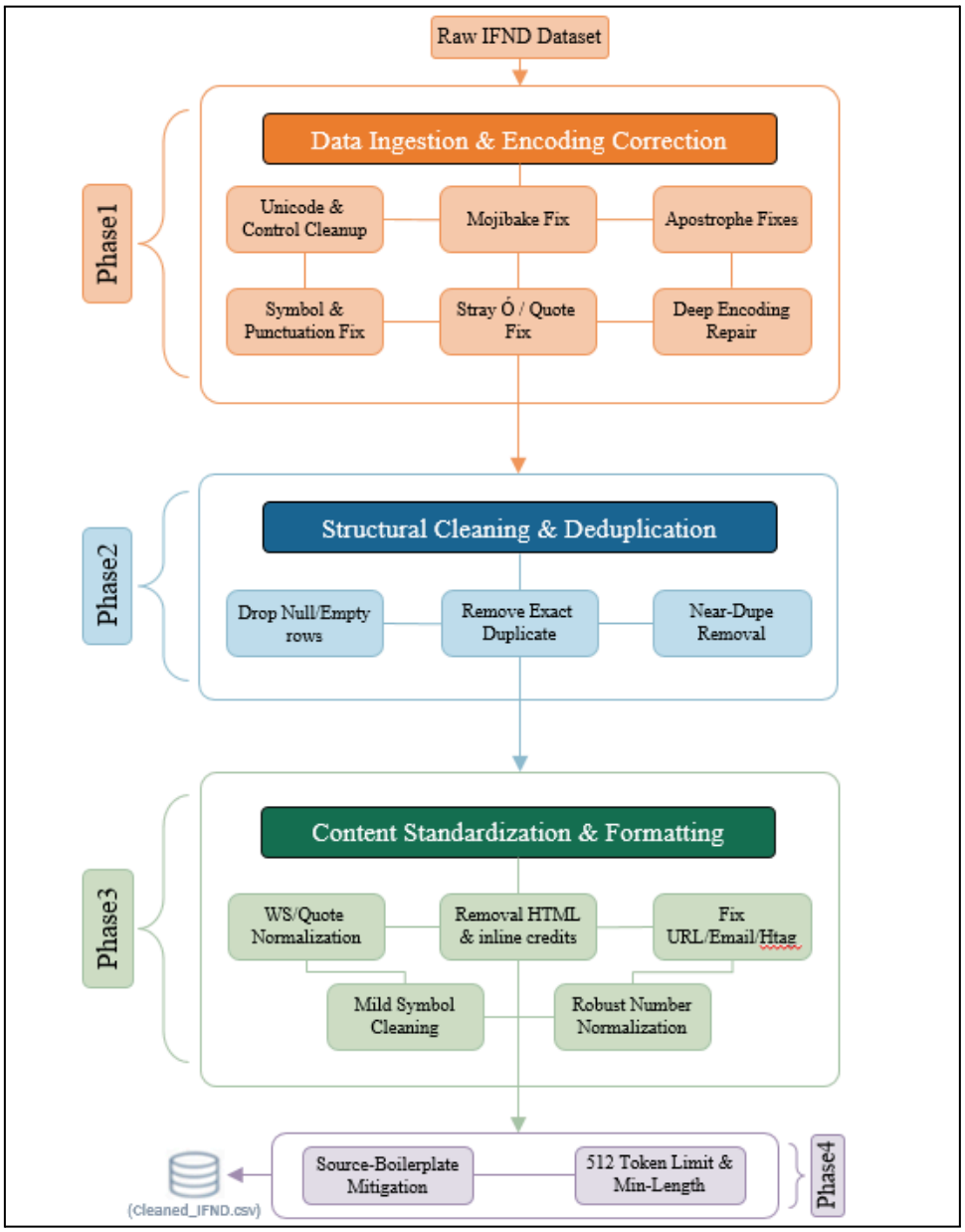
3.3.1 Dataset Description

1. Dataset used: IFND (Indian Fake News Dataset)

2. Multimodal Dataset as it contains both text and images covering news from 2013-2021.
3. The total size of the dataset is 56,868 news items.
4. It contains a total of 7 columns.
5. Raw data contained broken characters, HTML leftovers, encoding errors, near-duplicates, and punctuation inconsistencies, which required an extensive cleaning pipeline.

3.3.2 Data Cleaning

1. The IFND dataset needed a full cleaning pipeline because the raw text contained encoding issues, noisy characters, duplicates, HTML fragments, and uneven punctuation.
2. The cleaning process followed around 16 small but important steps, grouped into phases for encoding fixes, structural cleaning, content formatting, and number normalization
3. Clean data helped the models learn real linguistic patterns instead of noise, which ultimately improved stability and accuracy in testing.



[Fig. 3.3. Data Cleaning Pipeline]

3.3.3 Train-Test Split

After cleaning, the dataset was split into:

1. 80% Training (42,203 samples)
2. 10% Validation (5,275 samples)
3. 10% Testing (5,276 samples)

3.4 Implementation

3.4.1 Model Implementation

1. Dataset Splitting + RoBERTa Pipeline

   The IFND dataset was split in an 80/10/10 stratified ratio to ensure the TRUE and FALSE labels remained balanced.

   The RoBERTa tokenizer converted the text into input IDs and attention masks, creating a model-ready format.

   The model was trained using AMP, gradient clipping, a warmup scheduler, and early stopping to maintain stable learning.

2. Hybrid Model (Bi-LSTM + Attention + GRU)

   The Keras tokenizer converted the text into padded integer sequences with a maximum length of 128.

   The hybrid architecture used Bi-LSTM, an attention mechanism, and GRU layers to extract deep contextual features.

   Training was performed with class-weighted loss, gradient clipping, learning-rate scheduling, and early stopping.

3. Feature-Level Fusion (RoBERTa + Hybrid)

   The best RoBERTa and Hybrid models were loaded in frozen feature-extractor mode, meaning only the fusion classifier was trainable.

   RoBERTa's 768-dim CLS embedding and the Hybrid model's 128-dim GRU output were extracted and combined into a single fused vector.

   A lightweight MLP classifier was trained on this fused representation, producing the final fake-news predictions.

Code Snippets:

Snippet 1: (Dataset Splitting)

```
train_df, temp_df = train_test_split(
    data, test_size=0.2, stratify=data["Label"], random_state=42
)
val_df, test_df = train_test_split(
    temp_df, test_size=0.5, stratify=temp_df["Label"], random_state=42
)
```

[Fig. 3.4.1., Splitting of dataset]

Snippet 2: (RoBERTa Dataset Loader)

```
class TextClsDataset(Dataset):
    def __getitem__(self, idx):
        enc = tokenizer(str(self.texts[idx]),
                        max_length=128,
                        truncation=True,
                        padding="max_length",
                        return_tensors="pt")
        item = {k: v.squeeze(0) for k,v in enc.items()}
        item["labels"] = torch.tensor(self.labels[idx])
        return item
```

[Fig. 3.4.2., Loader of dataset]

Snippet 3: (Training Core)

```
with torch.cuda.amp.autocast():
    outputs = model(**batch)
    loss = F.cross_entropy(outputs.logits, labels, weight=class_weights)
scaler.scale(loss).backward()
torch.nn.utils.clip_grad_norm_(model.parameters(), 1.0)
```

[Fig. 3.4.3., Training of dataset]

Snippet 4: (Early Stopping)

```
 if val_f1 > best_f1:
     best_f1 = val_f1; wait = 0
 else:
     wait += 1
     if wait >= PATIENCE:
         break
```

[Fig. 3.4.4., Early Stopping]

Snippet 5: (Keras tokenizer to convert text into padded integer sequences)

```
tokenizer.fit_on_texts(train_df["clean_text"])
seq = tokenizer.texts_to_sequences(texts)
X_train = pad_sequences(seq, maxlen=128, padding='post')
```

[Fig. 3.4.5., Conversion of text into sequences]

Snippet 6: (Hybrid Architecture Forward Pass)

```
lstm_out,_ = self.bilstm(x)
attn_vec,_ = self.attn(lstm_out)
gru_input = attn_vec.unsqueeze(1)
gru_out,_ = self.gru(gru_input)
logits = self.fc(gru_out[:,-1,:])
```

[Fig. 3.4.6., Forward Pass]

Snippet 7: (Training Core)

```
logits = model(X)
loss = criterion(logits, y)
loss.backward()
torch.nn.utils.clip_grad_norm_(model.parameters(), 1.0)
optimizer.step()
```

[Fig. 3.4.7., Training]

Snippet 8: (LR Scheduler + Early Stopping)

```
scheduler.step(val_f1)
if val_f1 > best_f1:
    best_f1 = val_f1; wait = 0
else:
    wait += 1
    if wait >= PATIENCE:
        print("Early stopping.")
        break
```

[Fig. 3.4.8., LR scheduler and early stopping]

Snippet 9: (Load Best Saved Models)

```
roberta_state = torch.load(ROBERTA_BEST_PATH, map_location=device)
roberta_model.load_state_dict(roberta_state)

checkpoint_hyb = torch.load(HYBRID_BEST_PATH, map_location=device)
hybrid_model.load_state_dict(checkpoint_hyb['model_state_dict'])
```

[Fig. 3.4.9., loading best saved models]

Snippet 10: (Freeze Encoders)

```
for p in roberta_model.parameters(): p.requires_grad = False
for p in hybrid_model.parameters():  p.requires_grad = False
```

[Fig. 3.4.10., Freezing encoders]

Snippet 11: (Extract Feature Embeddings)

```
cls_vec = outputs.hidden_states[-1][:,0,:]
gru_vec = hybrid_model.encode(hyb_seq)
```

[Fig. 3.4.11., Feature embedding extraction]

Snippet 12: (Concatenate Features)

```
fused = torch.cat([cls_vec, gru_vec], dim=1)   # [B, 896]
```

[Fig. 3.4.12., Feature Concatenation]

Snippet 13: (Train Fusion Classifier)

```
logits = fusion_classifier(fused)
loss = criterion(logits, labels)
loss.backward()
optimizer.step()
```

[Fig. 3.4.13., Fusion Classifier Training]

Tools and Techniques

1. Python as the core language for implementing each stage of the pipeline.
2. PyTorch is used for building the RoBERTa model, Hybrid LSTM-GRU models, and the final fusion classifier.
3. TensorFlow/Keras used specifically for tokenization and sequence preparation for Hybrid model training.
4. Matplotlib for ROC curves, evaluation graphs, and accuracy visualizations.
5. Google Colab GPU for full training pipeline with mixed precision.
6. Feature-level Fusion for combining embeddings from RoBERTa and Hybrid models.
7. Class Weighting to handle dataset imbalance and improve fairness.
8. MLP classifier + Softmax for final decision layer.

3.4.2 Frontend Implementation

At this stage in the project, only the front-end of the system has been completed. A simple, user-friendly website was created to serve as an interface for the Deepfake-News Shield.

The homepage has two main features:

1. Input a news article in text form for verification.
2. Upload an image to check for manipulation.

The design ensures clarity and easy navigation for users. Once the input is provided, it is intended to pass through preprocessing and validation steps, as illustrated in the flowchart in Chapter 2.

Implementation Stack:

The frontend of the system was created with the following technologies:

HTML: Utilized for constructing the structure and layout of the web pages.

CSS: Used for the styling and visual design as well as making the interface responsive for different devices.

JavaScript: Added interactivity, enabling dynamic behaviour such as form submission and handling user inputs.

These technologies were chosen for their simplicity, compatibility, and efficiency in creating lightweight and responsive web applications.

Code Snippets:

1. HTML (Snippet)

```
<div class="panel-body" id="panel-text">
    <label for="newsText">Enter News Article</label>
    <textarea id="newsText" placeholder="Paste or type news text
here..."></textarea>

    <div class="controls">
        <div style="color: var(--muted); font-size: 13px;">
            Tip: Paste short article (200–1000 words)
        </div>
        <button class="btn" id="analyzeBtn">Analyze News
→</button>
    </div>
</div>
```

[Fig. 3.4.2.1. This snippet creates the main input area where users can paste a news article and click the button to analyze it.]

2. CSS (Snippet)

```
html, body {
    height: 100%;
    margin: 0;
    background: linear-gradient(180deg, var(--bg) 0%, #041226 100%);
    color: #e6f6f5;
}
```

[Fig. 3.4.2.2. This snippet defines the dark gradient background and global text styling for the website.]

```
.btn {
    padding: 12px 20px;
    border-radius: 8px;
    border: 0;
    cursor: pointer;
    font-weight: 700;
    background: linear-gradient(90deg, var(--accent), var(--accent-2));
    color: #042226;
    box-shadow: 0 8px 18px rgba(6, 182, 212, 0.12);
}
```

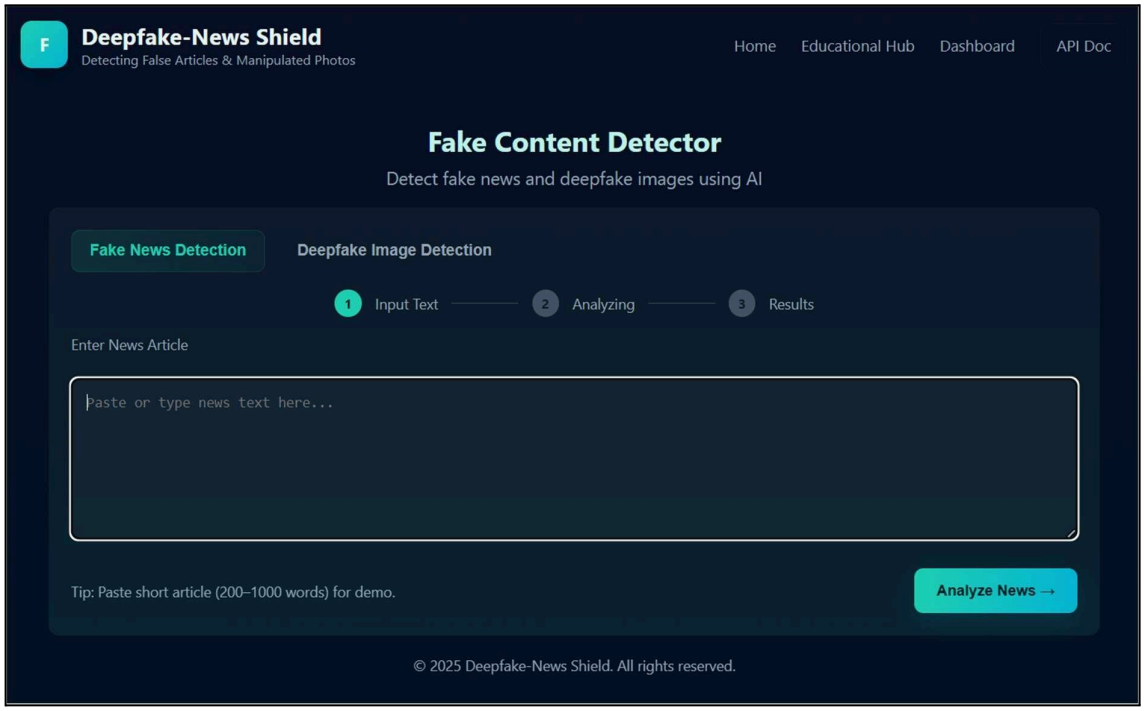
[Fig. 3.4.2.3. This snippet creates the modern gradient buttons used for actions like “Analyze News.”]

3. JS (Snippet)

```
analyzeBtn.addEventListener('click', () => {
    const txt = document.getElementById('newsText').value.trim();
    if (!txt) {
        alert('Please paste some news text to analyze (demo).');
        return;
    }

    analyzeBtn.textContent = 'Analyzing...';

    setTimeout(() => {
        const fakeScore = Math.random();
        let verdict =
            fakeScore > 0.6 ? 'Likely Fake' :
            (fakeScore > 0.25 ? 'Possibly Manipulated' : 'Likely Real');

        alert('Demo result: ' + verdict);
        analyzeBtn.textContent = 'Analyze News →';
    }, 1200);
});
```

[Fig. 3.4.2.4. This snippet shows the core logic for analyzing news text in demo mode, randomly generating a fake/real result.]

Result



[Fig. 3.4.2.5. Website's Home Page]

3.5 Key Challenges

1) Handling Noisy and Inconsistent data: the IFND dataset contained HTML leftovers, encoding issues, punctuations etc which were important to remove to get the best possible results. Maintaining semantic meaning while removing these inconsistencies was challenging.
2) Class Imbalance: the TRUE and False labels were not perfectly balanced and hence appropriate measures like careful splitting, validation checks were applied.
3) Managing Multiple Model Architectures: the project experimented with numerous models involving LSTM, BiLSTM, GRU, Transformer models etc and hence ensuring consistency across all models, saving and comparing them increased complexity.
4) Training Complexity: bigger models like RoBERTa, DeBERTa required high GPU, gradient clipping etc and hence maintaining stability without overfitting was a major challenge.

5) Fusion Architecture: combining outputs of RoBERTa and Hybrid LSTM-GRU embeddings required careful fusion and freezing encoders while training needed focused implementation.
6) Ensuring Generalization: since news varies by language, tone, source, models needed to learn beyond keyword-based patterns to show reliable predictions, which required the use of more advanced models.

Chapter 04: TESTING

4.1 Testing Strategy

The testing strategy for this project follows a multi-layer validation framework. The strategy is divided into the following levels:

4.1.1 Unit Testing

1) **Text Cleaning Module:** removal of HTML Tags, URLs, special characters, fixing encoding issues and ensuring consistent casing.
2) **Tokenization:**Keras tokenizer correctly converts text to sequences, RoBERTa tokenizer producing accurate input_ids and attention_mask. Maximum sequence length of 128.
3) **Embedding Generation:** RoBERTa returns a 768-dim CLS embedding and hybrid model returns 128-dim learned embeddings ensuring no dimension mismatch.
4) **Fusion Pipeline:** Unit testing checked if dense layer shape is correct, dropout was activated during training only and softmax produced valid probability distribution.

4.1.2 Integration Testing

This stage ensured that all components flowed seamlessly from input to final prediction.

1) **Text → Tokenizer → Hybrid Model:** verified sequence lengths passing, ensured attention layer receives proper shaped matrices and also kept track of GPU outputs alignment with fusion input.

2) **Text → RoBERTa → Embedding Vector:** No truncation or misalignment in attention masks and validation of embedding stability.
3) **Hybrid Embedding + RoBERTa Embedding → Fusion Vector:** Combined vector must be 896 dimensions and testing of missing and empty embeddings' handling.
4) **Fusion Output → Softmax Classifier:** verified consistent mapping: if **index 0 - TRUE and if index 1 - FALSE**.

4.1.3 System Testing

Implemented for validation of full pipeline:

User Input → Preprocessing → Tokenization → Embeddings → Fusion MLP → Softmax → Prediction

i. handling of very long news articles

ii. ensuring machine handles fake but well-written propaganda

iii. testing of instances where only title is provided

4.1.4 Performance and Accuracy Testing

The effectiveness of each model was tested on the test split:

1) Metrics used: Accuracy, F1-Score, ROC-AUC
2) Ensured that models have low false-positive rate, better generalization.

4.2 Test Cases and Outcomes

4.2.1 Test Cases

A. Text Preprocessing and Tokenization Test Cases

Table 4.1 Test Case A

S.no	Input	Expected Output
1.	Clean text with normal sentences	tokens generated, no HTML, no emojis
2.	Text containing emojis, special characters	emojis removed, punctuation normalized

3.	articles with 2000+ input	truncated/padded t0 128 length
4.	empty text input	error message

B. RoBERTa Encoder Test Cases

Table 4.2 Test Case B

S.no	Input	Expected Output
1.	normal articles	768-dim CLS embedding returned
2.	indirect news or sarcastic news	embedding still generates meaningful vector

C. Hybrid Model Test Cases

Table 4.3 Test Case C

S.no	Input	Expected Output
1.	normal text	128-dim GRU output embedding
2.	very short text	model still produces stable embedding

D. Fusion Model Test Cases

Table 4.4 Test Case D

S.no	Input	Expected Output
1.	valid 896-dim fused vector	output shape (2) with softmax probabilities
2.	Random noise vector	valid probab output
3.	missing RoBERTa embedding	error

E. End-to-End Fake News Prediction Test Cases

Table 4.5 Test Case E

S.no	Input	Expected Output
1.	real news from dataset	Predicted label = TRUE
2.	fake news from dataset	Predicted label = FALSE
3.	unseen fake message	model generalizes and FALSE predicted
4.	well written propaganda	softmax balanced and correct probab

4.2.2 Expected Outcomes:

1) The system should maintain high accuracy on unseen news.
2) Fusion models must outperform plain RoBERTa and hybrid models.
3) The pipeline should remain intact under noise and long inputs.

Chapter 05: RESULTS & EVALUATION

5.3 Results

Initially, multiple deep-learning models were trained and tested on the IFND dataset to compare their performance, stability, and generalization ability.

A. Baseline LSTM and BiLSTM models:

Table 5.1 Baseline Models

Model	Accuracy	F1	Precision	Recall	F1	ROC_AUC
LSTM	0.9544	0.9700	0.9500	0.9900	0.97	0.9800
BiLSTM	0.95555	0.9665	0.9600	0.9853	0.97	0.9800

B. Attention-Based Mechanism on variations of LSTM and BiLSTM:

Table 5.2 Attention-Based Mechanisms

Model	Accuracy	ROC-AUC	F1-Score
Attention + LSTM	0.9671	0.98335	0.9665
Attention + BiLSTM	0.9719	0.98863	0.9665
Attention + Stacked LSTM	0.9600	0.96000	0.9600
MHA + BiLSTM + Residual	0.9726	0.98867	0.9671
Residual BiLSTM	0.9726	0.98820	0.9668
Scaled Dot-Product MHA + Residual	0.9726	0.98905	0.9673

C. GRU-based Hybrid Models:

Table 5.3 GRU-Based Hybrid Models

Model	Accuracy	ROC-AUC	F1-Score	Time (min)
LSTM	0.9544	0.9800	0.9700	1.41
GRU on LSTM no early stopping	0.9371	0.9600	0.9500	4.55
GRU on LSTM	0.9546	0.9800	0.9700	1.07
Stacked GRU on LSTM no	0.9396	0.9600	0.9600	5.51
Stacked GRU on LSTM	0.9535	0.9800	0.9700	1.43
Bidirectional GRU on LSTM no	0.9380	0.9600	0.9700	1.51

D. Transformer Models:

Table 5.4 Transformer Models

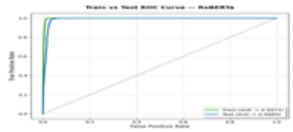
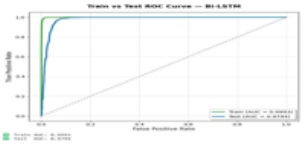
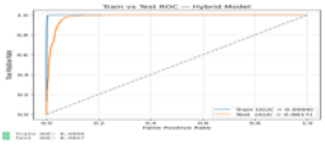
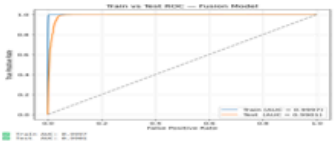
Model	Accuracy	ROC-AUC
DEBERTA-V3	0.9722	0.9817
T5	0.9678	0.9748

From all the experiments, the two strongest models — **RoBERTa** and the **Hybrid Bi-LSTM + Attention + GRU model** — consistently achieved the highest accuracy, macro-F1, and ROC-AUC scores. These top models were finally combined using a **feature-level fusion approach**, and their fused output was evaluated through detailed metrics and visual graphs to validate performance improvement.

The complete comparison of all models results are in the following table, where their test accuracy, AUC scores, and ROC curves are clearly presented:

Final Comparison Table:

Table 5.5 Final Comparison Table

Model	Test Accuracy	Test AUC	ROC Curve
RoBERTa	0.9795	0.9889	
Bi-LSTM	0.9621	0.9794	
Hybrid1 (BiLSTM+Attn+2layer/Stacked GRU)	0.9547	0.9791	
Hybrid2 (Bi-LSTM+Attn+1layer GRU)	0.9623	0.9817	
Fusion (RoBERTa + Hybrid2)	0.9797	0.9901	

Chapter 06: Conclusions and Future Scope

6.1 Conclusion

The project Deepfake- News Shield: Detecting False Articles and Manipulated Photos successfully demonstrates an end-to-end pipeline for detecting misinformation using advanced deep learning models. A large and noisy IFND dataset was preprocessed, cleaned and standardised using unique steps to make it appropriate for model training. One of the key developments achieved through this project was progression from traditional deep learning to hybrid architectures and advanced transformer models. Significant improvements and learnings were observed through this approach on how to design, train and evaluate deep learning architectures. Working with these models provided practical experience in managing tokenization, sequence modeling, class imbalance handling and early stopping. One of the most important outcomes was the successful implementation of feature-level fusion where RoBERTa's findings were combined with a hybrid model to produce a more robust and accurate classifier.

Key Findings

1. Cleaning and preprocessing significantly improved the quality of the dataset, resulting in stable and accurate performance.
2. Transformer models consistently outperformed baseline LSTM-based architectures.
3. Hybrid models captured sequential and contextual patterns effectively.
4. Feature-level fusion of RoBERTa and hybrid models produced the most stable performance across accuracy, ROC-AUC and F1-score.

Limitations

1. The current system focuses only on text so misinformation contained by images, video or audio cannot yet be detected.
2. Transformer models require high computational power, limiting experimentation with more advanced and larger versions.
3. The IFND dataset is India-specific, hence it may not fully generalize to global misinformation.
4. The web interface currently is not connected to backend and operated in demo mode due to which real-time results are not yet available.

Contributions to the Field

1. Provides a modular architecture for fake news detection and combines deep learning, attention mechanism and transformer modes together.
2. Introduces feature-fusion framework merging embeddings of transformers with hybrid sequence-based features for improved accuracy, robustness and generalization.
3. Demonstrates a data cleaning pipeline with 16 unique steps appropriate for the dataset.
4. Offers a web based interface for real time misinformation detection.
5. Lays the foundation expanded the project into multimodal detection.

6.2 Future scope

1. **Adding image based detection using ViT or EfficientNet B3:** This can help the system catch fake news that contains edited or misleading images. Models like ViT or EfficientNet B4 can understand visual patterns that the text model cannot see.
2. **Building a multi modal fusion model:** A fusion model that mixes text, images, and metadata can provide a more complete view of the content. This usually helps the classifier keep stronger and more accurate.

3. **Deploying the final system using Flask:** The model can be turned into a simple web application so users can test it in real time. Deployment also helps check how the system performs on new and unseen data.
4. **Incorporating Video and Audio Deepfake Detection:** The system can be expanded to detect deepfake videos and audios using models like XceptionNet, 3D CNNs, Wav2Vec.
5. **Integrating External Fact-Checking APIs:** we can connect the system to APIs such as Google FactCheck, NewsAPI to automatically verify claims and also display evidence or sources of the input.

REFERENCES

[1] S. Kaliyar, A. Goswami, and P. Narang, “FakeBERT: Fake news detection in social media with a BERT-based deep learning approach,” Multimedia Tools and Applications, vol. 80, no. 8, pp. 11765–11788, 2021.

[2] M. Kaur, S. Kumar, and S. Bawa, “Rumor detection on social media: A data mining perspective,” Information Systems Frontiers, vol. 24, no. 5, pp. 1485–1506, 2022.

[3] A. Rossler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Nießner, “FaceForensics++: Learning to detect manipulated facial images,” in Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 2019, pp. 1–11.

[4] H. Nguyen, J. Yamagishi, and I. Echizen, “Use of a capsule network to detect fake images and videos,” arXiv preprint arXiv:1910.12467, 2019.

[5] M. H. A. Khan and F. Algarni, “Deepfake detection using convolutional neural networks for improved security on social media,” IEEE Access, vol. 9, pp. 143825–143837, 2021.

[6] B. Ghita, I. Kuzminykh, A. Usama, T. Bakhshi, and J. Marchang, “Deepfake Image Detection using Vision Transformer Models,” IEEE/academic publication, 2024.

[7] N. N. Prachi, M. Habibullah, M. E. H. Rafi, E. Alam, and R. Khan, “Detection of Fake News Using Machine Learning and Natural Language Processing Algorithms,” Journal of Advances in Information Technology, vol. 13, no. 6, pp. 652–661, Dec. 2022, doi: 12720/jait.13.6.652-661.

[8] Karthikeyan A., Monniesh B., Kishorekumar V., and Niveshkumar S., “Enhancing Deepfake Detection: A Multimodal Approach for Improved Accuracy,” TIJER – International Research Journal, vol. 11, no. 7, July 2024, pp. 583–587.

[9] C.-M. Rosca, A. Stancu, and E. M. Iovanovici, “The New Paradigm of Deepfake Detection at the Text Level,” Appl. Sci., vol. 15, art. 2560, Feb. 27, 2025, doi: 10.3390/app15052560.

[10] A. Heidari, N. J. Navimipour, H. Dag, and M. Unal, “Deepfake detection using deep learning methods: A systematic and comprehensive review,” WIREs Data Mining and Knowledge Discovery, vol. 14, no. 2, Feb. 2024, Art. no. e1520.

[11] 10N. Shakya and P. Poudyal, "Detection of Fake News Using Deep Neural Networks," Kathmandu University Journal of Science, Engineering and Technology, vol. 16, no. 2, pp. 110–119, 2022.

[12] R. Tolosana, R. Vera-Rodriguez, J. Fierrez, A. Morales, and J. Ortega-Garcia, "DeepFakes and Beyond: A Survey of Face Manipulation and Fake Detection," Information Fusion (Elsevier), 2020.

[13] Z. He, et al., "GazeForensics: DeepFake Detection via Gaze-guided Spatial Inconsistency Learning," arXiv preprint, 2023.

[14] B. Dolhansky, J. Bitton, B. Pflaum, J. Lu, R. Howes, M. Wang, and C. C. Ferrer, "The DeepFake Detection Challenge (DFDC) Dataset," arXiv preprint arXiv:2006.07397, 2020.

[15] B. B. Gupta, S. Kumar, Shubham, and A. Jaiswal, "Deep Fake: An Overview," in Smart and Innovative Trends in Engineering and Technology (SITET-2020), 2021.

[16] M. Westerlund, "The Emergence of Deepfake Technology: A Review," Technology Innovation Management Review, vol. 9, no. 11, pp. 39–52, Nov. 2019.

[17] S. Singhal, R. R. Shah, T. Chakraborty, P. Kumaraguru, and S. Satoh, "SpotFake: A Multi-modal Framework for Fake News Detection," in Proc. IEEE Int. Conf. Multimedia Big Data (BigMM), 2019.

[18] R. K. Das, "Fake News Detection After LLM Laundering: Measurement and Explanation," arXiv preprint arXiv:2501.18649, 2025.

[19] F. Siddiqui, J. Yang, S. Xiao, and M. Fahad, "Enhanced deepfake detection with DenseNet and Cross-ViT," Expert Systems With Applications, vol. 267, p. 126150, 2025.

[20] R.Tolosana,R.Vera-Rodriguez,J.Fierrez,A.Morales,andJ. Ortega-Garcia,"DeepFakes and Beyond: A Survey of Face Manipulation and Fake Detection," arXiv preprint arXiv:2001.00179, 2020.