Digital Threat: Misinformation Detection using LSTM in TensorFlow

By
Bikrom Adatya Roy
ID: 213-15-4561

Mir Saleh Ahmed
ID: 213-15-4244

FINAL YEAR DESIGN PROJECT REPORT

This Report Presented in Partial Fulfillment of the Requirements for the **Degree of Bachelor of Science in Computer Science and Engineering**

Supervised by

Dr. Abdus Sattar
Associate Professor and
Director of MSc in CSE
Department of Computer Science and Engineering, Daffodil International University

Co-Supervised by

Mohammad Jahangir Alam
Assistant Professor
Department of Computer Science and Engineering, Daffodil International University

DAFFODIL INTERNATIONAL UNIVERSITY
Dhaka, Bangladesh

September 16, 2025

APPROVAL

This Project titled **"Digital Threat: Misinformation Detection using LSTM in TensorFlow"**, submitted by **Bikrom Adatya Roy**, ID No: **213-15-4561** and **Mir Saleh Ahmed**, ID No: **213-15-4244** to the Department of Computer Science and Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Engineering and approved as to its style and contents. The presentation has been held on **16 September, 2025**.

BOARD OF EXAMINERS

Dr. Arif Mahmud — **Chairman**
Associate Professor and Associate Head
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Dr. Md. Ali Hossain — **Internal Examiner**
Associate Professor
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

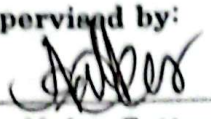
Mr. Amir Sohel — **Internal Examiner**
Assistant Professor
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Dr. Md. Manowarul Islam — **External Examiner**
Professor
Department of Computer Science and Engineering
Jagannath University

DECLARATION

We hereby declare that this project has been done by us under the supervision of **Mr. Abdus Sattar, Assistant Professor**, Department of Computer Science and Engineering, Daffodil International University. We also declare that neither this project nor any part of this project has been submitted elsewhere for the award of any degree or diploma.

Supervised by:

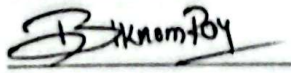


Dr. Abdus Sattar
Assistant Professor, Director of M.Sc. in CSE
Department of Computer Science and Engineering, Daffodil International University

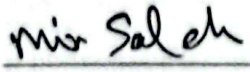
Co-Supervised by:

Mohammad Jahangir Alam
Assistant Professor,
Department of Computer Science and Engineering, Daffodil International University

Submitted by:



Bikrom Adataya Roy
Student ID: 213-15-4561
Department of Computer Science and Engineering, Daffodil International University



Mir Saleh Ahmed
Student ID: 213-15-4244
Department of Computer Science and Engineering, Daffodil University

ACKNOWLEDGEMENTS

This work would not have been possible without the support and contributions of many individuals over the past two semesters. We are deeply grateful to everyone who has assisted us in one way or another. First, we express our heartfelt thanks and gratefulness to the almighty for His divine blessing, making it possible for us to complete the **Final Year Design Project (FYDP)** successfully. We are grateful and wish our profound indebtedness to **Supervisor: Mr. Abdus Sattar, Assistant Professor**, Department of Computer Science and Engineering, Daffodil International University, Dhaka, Bangladesh. Deep knowledge and keen interest of our supervisor in the field of **Deep Learning** to carry out this project. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts, and correcting them at all stages have made it possible to complete this project.

We would like to express our heartfelt gratitude to the Head of the Department of Computer Science and Engineering, for his kind help in finishing our project and to other faculty members and the staff of the Department of Computer Science and Engineering, Daffodil International University.

We would like to thank our entire course mates at Daffodil International University, who took part in this discussion while completing the coursework.

Finally, we must acknowledge with due respect the constant support and patience of our parents.

We are sincerely thankful to Almighty Allah for granting us the strength, patience, and wisdom to carry out and complete this project.

ABSTRACT

In today's digital world, where false information spreads swiftly via social media and jeopardizes public trust, misinformation has become a significant issue. Despite the existence of some traditional detection methods, they are barely able to keep up with the evolving deception techniques and sophisticated manipulation techniques. In order to solve these kinds of problems, every paper develops a progress detection system based on bidirectional long short-term memory networks and their attention tool. Our system's coverage of both real and fake news was very wide because it was built on two large data sets that included roughly 45,000 news articles from numerous sources. We used a rigorous data preprocessing step to process our data, which involved tokenization, text removal, and the implementation of a custom LSTM architecture. In order to improve each article's ability to distinguish between real news and false information, the attention mechanism also helps the model focus on the most relevant portion. The outcomes were striking. Our model achieved a precision and recall score of over 99.8% and an accuracy of 99.88% on test data. A score of 1.0000 under the AUC-ROC indicates high discrimination of news categories, and the confusion matrix shows a few instances of misclassification. According to this data, deep learning techniques—particularly bidirectional LSTMs with attention—can also offer effective ways to combat false information. With references to automated fact-checking systems and social media monitoring, this model's high performance can be interpreted as a sign that it can be applied in practical situations. The current study provides practical solutions for maintaining information integrity in a world that is becoming more interconnected and where news accuracy must be confirmed.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Our work on misinformation detection is summarized in this chapter, with a focus on its significance in the digital era. Fake news threatens decision-making, safety, security, and trust because it spreads more quickly than the real thing.

1.1 Introduction

Fake information is one of the most dangerous problems in the current digital era. Most of it comes from social media and online news sources. It's considerably easier to reach a broad audience instantly. Detection systems, such as rule-based or single-model classifiers, often don't work well when faced with complicated manipulations and changing language patterns.

We use Deep Learning models that can work with more than one classifier to make them more reliable and accurate. We also use LSTM techniques to get linguistic and semantic features. Our goal is to make it easier to find false information in different types of digital content and to promote safer, more reliable online communication.

1.2 Motivation

Countries don't just fight with direct combat, heavy artillery, missiles, or drones anymore. Even though traditional military tactics still work, the digital age has brought about a new kind of war in which propaganda and false information are used as weapons to weaken enemies without fighting, destabilize societies, and sway public opinion. Recent events have shown how digital manipulation can change people's feelings, causing unrest and making diplomatic efforts harder. For instance, the war between Russia and Ukraine has shown how fake news and altered images and videos are spread on social media on purpose to change people's minds and weaken the other side. China and Taiwan are also dealing with advanced online deception efforts that are trying to change the stories about safety, security, and sovereignty. Fake news has also made things worse in

conflicts in the region, like the one between India and Pakistan. During Operation Shindoor, reports that were not true about military activities spread quickly online, causing confusion and distrust among the people of both countries and affecting the situation around the world.

Disinformation has an impact on both central and local conflicts, with propaganda and false news shaping public opinion and changing how people see the world. Thailand and Cambodia have been fighting over the sacred Shiva Temple in Southeast Asia for a long time. Also, false stories are used to change people's minds at home and put pressure on people from other countries. These efforts to spread false information give short-term strategic advantages, twist the truth, and make tensions between groups or countries worse. Digital propaganda differs from traditional weapons because it can be deployed at a low cost, impact millions instantly, and exploit citizens' emotional biases, as they rely on social media and online news for information. The extensive spread of misinformation degrades institutions, creates hysteria, and makes societies vulnerable to manipulation, demonstrating that misinformation is not merely a social or political issue anymore but also a necessary digital threat.

From a technological and research perspective, the issue of handling misinformation is compounded by its velocity, volume, and increasingly high sophistication. Classical fact-checking does not serve the purpose because misinformation is crafted to be plausible-sounding and emotionally appealing. Worldwide dissemination of content leaves little time for manual verification, and automated methods are necessary. LSTM, TensorFlow, and Deep Learning algorithms offer hopeful approaches to detect, classify, and hinder the spread of fake news in real-time. Inspiration for this study is therefore guided by the imperatives of developing robust frameworks for misinformation detection that can safeguard democratic processes, national security, and citizens from digital deception. In responding to this key challenge, the research plays a role in contributing towards international efforts to counter propaganda and preserve information integrity during crises and conflicts.

1.3 Research Objectives

Primary objectives for the project are:

- To identify & understand the challenges caused by fake or spam news.
- To apply LSTM techniques for extracting meaningful linguistic and semantic features from text data.

1.4 Methodology

A Kaggle dataset with classified fake and genuine news was utilized by us. Preprocessing of the data involved vectoring it with TF-IDF as well as purifying the text (removing punctuation, stop words, and lowercasing). Popular terms for both groups have been identified through Word Cloud visualizations. Eighty per cent of the dataset was used for training, and twenty per cent was used for testing. The binary losses of cross-entropy and an Adam optimization method were employed for training a bi-directional LSTM-based model that included 1,088,614 parameters. A confusion matrix, ROC curve, and probability distributions were used to assess the model's performance. For an actual application, the model with the greatest efficiency was saved with Jabil and then used in a web-based design.

1.5 Research Question

Based on this thesis paper, the aim answered the following Questions:

Research Question (RQ1) 01: How can deep learning frameworks, specifically LSTM and Bi-Directional LSTM (Bi-LSTM), be used for classification and to detect misinformation in online news portal articles, social media comments, and effectively?

Research Question (RQ2) 02: What combination of model architecture, optimization strategy, and evaluation metrics creates the strongest framework for detection and classification of misinformation, and how can the best model be implemented in a particular web-based system?

1.6 Project Outcome

The creation of LSTM within TensorFlow and deep learning that can precisely differentiate among true and false information constitutes the primary outcome of this research-based project. This dataset's execution algorithm is excellent, supporting its dependability and efficacy in spotting false news.

- Kaggle offers for the implementation of a highly accurate identification and classification method to identify false news data. to guarantee that each user can simply specify the text data that the suggested model will use for classification and detection.
- This project's computational structure for misinformation detection using a bidirectional LSTM in TensorFlow is comprehensive and clearly defined. This framework can serve as a starting point for additional research or development.
- Subsidence of digital threats: The project's accomplishments demonstrate that deep learning, and more especially LSTMs, can be an effective barrier against the dissemination of false information, which is a serious online danger. The result is an automated content control technique that has been scientifically verified as effective.
- Establishing a process pipeline for data processing. After preprocessing and augmenting a dataset of misinformation detection and classification, a strong training set that reflects real-world variances was produced. High-performance metrics indicate that the models utilized in the Bi-directional LSTM core logic sound, based on the results on a particular dataset. This offers a solid foundation for further advancement, such as honing the model on a more intricate and varied dataset to see how well it generalizes to various forms of false information.

1.7 Organization of the Report

- **Chapter 1 Introduction:** Introduces the background, motivation, objectives, research scope, and specific outcome of the strong defense against digital misinformation by enabling accurate detection, and the result is a reliable method for automatic content moderation.
- **Chapter 2 Background:** Discusses the history of long- and short-term memory (LSTM) in TensorFlow, identification, and execution approaches in news formation classification-related deep learning applications and identifies the research gaps addressed by this project.
- **Chapter 3 Research Methodology:** Provided an in-depth description of the system design dataset collection from Kaggle and true and fake data to detection, preprocessing, and augmentation strategies, and proposed

methodology model architectures, explainable with the deep learning approach, and evaluation methods.

- **Chapter 4 Implementation and Results:** Details the experimental environment setup, model training phase, testing strategies, performance evaluation metrics, confusion matrix (real - 4278 and fake - 4689), and ROC-AUC analysis. and a comparative analysis of Bi-directional with LSTM TensorFlow models.
- **Chapter 5 Engineering Standard and Design Challenges:** Describes compliance with software and hardware standards, communication protocols, ethical considerations, and environmental and societal impacts, sustainability planning, and complex engineering problem mapping associated with the project.
- **Chapter 6 Conclusion:** Summarizes the achievements of the project, highlights the limitations faced during development, and outlines future directions for improvement, scaling, and real-world application for device deployment.

Chapter 2

Background

This chapter gives the research team the context that it needs by describing the ideas, difficulties, and relevant literature on misinformation detection. It presents the basic information needed for understanding issues and reviews previous studies which is relevant for this research study.

2.1 Introduction

One of the biggest problems for the modern era is inaccurate information, which is mostly caused by the quick development of social media and online news sources. Fake or misleading information spreads more quickly than authentic news and affects harmony in society, political issues, public sentiment, along with medical decisions. In order to automatically analyze texts, find pertinent features, and classify information as accurate or inaccurate, deep learning and LSTM TensorFlow have emerged as effective tools. This section presents the fundamental concepts and sources of inspiration that can guide this inquiry.

2.2 Literature Review

Chopra, Kaushik, Rathore, and Kaur (2023) proposed a deep learning and natural language processing system to detect false information. To generate a dataset of 44,897 articles, they performed several preprocessing operations such as lowercase conversion, regular expression cleaning, stop-word removal, and lemmatization, before tokenization, padding to a fixed sequence length, and embedding. This dataset was then combined with datasets that showed fake and real news. The main model, which is part of TensorFlow/Keras, has two stacked LSTM layers for a sequential architecture and dense layers for binary classification. The model did very well after ten epochs of training with the Adam optimizer and binary cross-entropy loss. It had high F1-scores, recall, and precision, and it also had a training accuracy of 98.42% and a test accuracy of 98.39%.

.

Alladi and Bharathi studied a language to find fake news. They used advanced AI models like mBERT and LSTM to do this. The mBERT was 89% accurate on social media texts, which means it was better at finding real news than fake news. The researchers were careful and pointed out that the nearly perfect score might be due to the small dataset and could mean overfitting. However, the LSTM model was very accurate for full news articles, reaching 100%. Their findings show that these AI tools could be useful for Malayalam and other similar languages. However, they also show a big problem: the need for more balanced data to make sure the models are fair and work well in the real world.

Raza, Khan, and Usmani (2024) created a hybrid system that combines the content of an article with its publication date to find fake news. They tried out three AI models: LSTM, CNN-BiLSTM, and BERT. All of them worked very well. BERT had the best accuracy at 99%, followed by CNN-BiLSTM at 97.9%, and LSTM at 97.8%. Their main contribution was demonstrating that including temporal data, like the year, month, or day of the week of publication, greatly improves the models' ability to spot fake news patterns. The study shows that even simpler models like LSTM can work very well when given the right temporal information. However, advanced models like BERT are still the best.

A team of researchers (Kadam, Agarwal, Kumar, Jadhav, and Krishnan, 2023) created a novel system to detect fake news by combining several techniques. They used the powerful deep learning model called LSTM along with curriculum learning, a clever training method. This method trains the model by beginning with simpler data and progressively moving on to more complex examples, much like a student would learn. This methodical training, when combined with their other strategies, produced a system that achieved an astounding 94.8% accuracy rate. Their study shows how curriculum learning can enhance deep learning models' (like LSTM) ability to handle complex, real-world data from social media, ultimately offering a potent and flexible means of combating misinformation.

Keya et al. released the hybrid deep learning model Fake Stack in 2023 with the goal of identifying fake news. It incorporates a deep CNN with skip connections (to capture local features and enhance gradient flow), an LSTM layer (to handle

long-range dependencies), and BERT (for contextual embeddings) and was published in PLOS ONE. The model, which was created using TensorFlow, Keras, and Python, was tested on three datasets: Kaggle Fake News, LIAR, and Fake. Article titles and texts were combined, BERT embeddings were made, and the CNN-LSTM architecture was run through them. On the primary dataset, it outperformed other deep learning and conventional models with an accuracy of 99.74% and an AUC of 1.0. Additionally, the model showed excellent generalization skills across various datasets. The authors came to the conclusion that this stacked, hierarchical method efficiently makes use of a variety of textual features to produce cutting-edge outcomes.

Adebayo et al. (2025) combined Bi-LSTM and SVM to create an ensemble model for detecting fake news using a Kaggle dataset comprising 44,898 articles. With an accuracy of 98.6%, the Bi-LSTM beat the SVM due to its superior comprehension of sequential and contextual language features. The authors stress that combining deep learning techniques with conventional machine learning enhances the model's resilience and flexibility, increasing its ability to fend off shifting misinformation.

Das and Dodge (2025) investigated how traditional detectors are challenged by paraphrased fake news produced by LLM. They compared deep learning models such as CNN, LSTM in TensorFlow, transformers BERT, T5, LLaMA, and classical machine learning models SVM, logistic regression, and random forest using the COVID-19 dataset, 10,500 news articles, and the LIAR dataset (12.8K political statements). While detectors performed well on human-written fake news (F1 ≈ 0.92–0.93), they performed significantly worse on LLM-paraphrased text (F1 ≈ 0.21–0.27), particularly when it came to Pegasus-generated news, according to the results. The authors point out that many detection failures can be explained by sentiment shift during paraphrasing, where text may maintain semantic similarity but take on a more upbeat tone. They come to the conclusion that robust models must incorporate semantic and sentiment awareness to combat misinformation because LLM laundering is a powerful adversarial attack.

Ghafoor et al. (2024) suggested a Bi-LSTM model with TF-IDF features for identifying fake news using 23,096 news articles. After preprocessing and

balancing, their Bi-LSTM achieved a 98% accuracy rate, outperforming Random Forest, SVM, and Logistic Regression. Precision (0.97), recall (0.99), and F1-score (0.98) all confirmed its reliability. The authors point out that Bi-LSTM is excellent at capturing contextual dependencies, while TF-IDF guarantees balanced feature representation. They come to the conclusion that TF-IDF in conjunction with Bi-LSTM is a trustworthy and practical misinformation detection technique.

In order to identify false information in sustainable supply chain management (SSCM), this paper proposes a multi-stacked LSTM model based on DistilBERT. The study guarantees that the input features are ideal for classification by integrating TF-IDF and Word2Vec within a High Feature Extraction (HFE) framework and using LASSO for feature selection. The hybrid model outperforms conventional LSTM, RNN, and CNN techniques with an accuracy of 99.82% when tested on the WELFake dataset. The authors emphasize that in order to capture intricate patterns of disinformation and bolster confidence and dependability in SSCM communications, contextual embeddings must be integrated with sequential models.

A hybrid framework for detecting misinformation was presented by Cao et al. in 2025. This framework enhances its performance with TF-IDF, Word2Vec, and LASSO feature selection, and it employs Distil BERT in conjunction with multi-stacked LSTM. Their model surpassed traditional machine learning and deep learning methods, achieving 99.82% accuracy on the WELFake dataset. The study emphasizes the advantages of using contextual embeddings in conjunction with sequential models to identify intricate patterns of disinformation in sustainable supply chain environments.

In order to identify fake news, Agarwal et al. (2023) proposed a framework that integrates social and content characteristics. The system's accuracy reached 99.4% through the use of classifiers like SVM, Random Forest, Logistic Regression, and XGBoost. Compared to individual models, this performance is superior. The study highlighted how integrating social engagement metrics with content-driven linguistic patterns results in a more reliable and robust detection method, establishing ensemble techniques as a viable strategy for thwarting disinformation in practical contexts.

LR Summary on Table:

Author	Year	Title	Model used	Key Findings
Rupak Kumar Das, Jonathan Dodge	2025	Fake News Detection After LLM Laundering: Measurement and Explanation	BERT, T5, Llama, CNN, LSTM, SVM, RF, LR	Outperformed CNN, BERT-CNN, and BERT-LSTM; demonstrated hybrid deep learning strength
QiuPing Li, Fen Fu,	2025	Fake news detection with deep learning: Insights from multi-dimensional model analysis	BERT Large, Roberta, Text CNN, BERT+XGBoost, Logistic Regression	BERT Large is best in accuracy; Text CNN is efficient with fewer resources; and logistic regression is the most interpretable.
Peng Cao, Prashant Kumar Shukla	2025	Enhancing fake news detection for Sustainable Supply Chain Management using DistilBERT-based multi-stacked LSTM approach	DistilBERT + multi-stacked LSTM; TF-IDF + Word2Vec (HFE framework) + LASSO	The hybrid model outperforms conventional ML/DL methods in SSCM fake news detection.
Alladi and Bharathi	2025	Cognitext@DravidianLangTech2025: Fake News Classification in	mBERT, LSTM	Effective but limited detection of
Raza, Khan et al	2024	A Hybrid Deep Learning-Based Fake News Detection System Using Temporal Features	LSTM, CNN-BiLSTM, BERT	Temporal features boost with 97.8% accuracy.
Chopra et al	2023	Uncovering Semantic Inconsistencies and Deceptive Language in False News Using	stacked LSTM model in TensorFlow /	High performance with 98.39%

		Deep Learning and NLP Techniques for Effective Management	Keras, as with	test accuracy
Kadam et al	2023	SEMI-FND: Stacked Ensemble Based Multimodal Inference For Faster Fake News Detection	LSTM with curriculum	Improved detection accuracy with 94.8% accuracy
Keya et al.	2023	FakeStack: Hierarchical Tri-BERT-CNN-LSTM stacked model for effective fake news detection	FakeStack, a hybrid deep learning model, BERT-CNN-LSTM	State-of-the-art performance with 99.7% accuracy and 1.0 AUC.
Anjali Agarwal, Chandra L. Chowdhary	2023	Ensemble-based high-performance model for fake news detection using content and social features	Proposed ensemble approach integrating SVM, RF, NB, LR, and XGBoost with content (TF-IDF, n-grams) and social features.	Ensemble outperforms individual models; combining content and social features enhances accuracy.
Ashfia Jannat Keya, Hasibul Hossain Shajeeb, Md. Saifur Rahman, M. F. Mridha	2023	Fake Stack: Hierarchical Tri-BERT-CNN-LSTM stacked model for effective fake news detection	BERT + CNN (skip) + LSTM	Outperformed CNN, BERT-CNN, and BERT-LSTM; demonstrated hybrid deep learning strength.

2.3 Gap Analysis

Here are the summaries of the gaps we are found/observed from the related work study, where we intend to contribute:

Criteria	Adebayo et al.	Das & Dodge	Ghafoor et al.	Shu et al.	Wang (2017	Zhou &	Our Work

	(2025)	(2025)	(2024)	(2019)	)	Zafar ani (2020)	
High Accuracy	Yes	Yes (on human text)	Yes	No	No	Yes	Yes
Cross-domain Validation	No	No	No	No	No	No	Yes
Handles Imbalanced Data	No	No	Yes	No	No	No	Yes
Robust to LLM-paraphrased Fake News	No	No	No	No	No	No	Yes
Explainability (XAI)	No	No	No	Yes	No	No	Yes
Efficiency (Lightweight/Scalable)	No	No	No	No	No	No	Yes

2.4 Summary

Despite its significant limitations, the current literature review demonstrates how many studies used machine learning along with deep learning algorithms with high precision. The majority of the work is restricted to single-domain datasets and attempts to reduce the generalizability of their models. Many experienced difficulties with unbalanced data, lacked explainability, or failed to withstand the growing threat of summarized opposing text. Furthermore, efficiency and scalability were rarely considered, which made implementation in the real world difficult.

By introducing a Deep Learning framework with LSTM-based data augmentation, our recommendation solves these gaps and ensures high precision and domain-wide generalization. By emphasizing on robustness, comprehension, and scalability, the system that is suggested provides an improved method for recognizing false information in various and evolving digital environments.

Chapter 3

Research Methodology

This chapter focusses on our research paper's methodology. The research methodological overview, the suggested approach, the requirements, and the specifics of the project planning will all be covered here.

3.1 Methodology

The proposed system recognizes false information using a deep learning and LSTM TensorFlow model. This was achieved through an analysis of the functional specifications (system capabilities) and non-functional requirements (performance, scalability, and usability). The elements, process also system architecture need to develop a successful detection framework that are described in the design specification.

3.1.1 Overview

This study presents an efficient deep learning algorithm for detecting false information, employing a systematic and structured methodology. It follows a complete pipeline that starts with data collection and ends with model deployment and evaluation. The approach is based on a bidirectional LSTM (Long Short-Term Memory) neural network, a robust sequence analysis architecture implemented within TensorFlow. Each phase of the process, from data cleaning to model training, is precisely outlined and supported by a transparent and dependable design. This method enables a thorough assessment of deep learning performance on this critical task and provides a clear pathway for further research and development area.

3.1.2 Proposed Methodology/ System Design

This project uses a thorough procedure to convert raw data into a format appropriate for a deep learning model. In-depth feature engineering using word embeddings, data analysis, comprehensive preprocessing, and building an LSTM neural network are all included.

Data Acquisition and Exploration: First, the fake and true.csv files.CSV files have been loaded. To determine the features of the data set, such as the subject matter, text lengths, and class order, an initial exploratory data analysis (EDA) is conducted. This stage is crucial for identifying potential biases and issues with the quality of the information that need to be addressed later.

Text Preprocessing: Text processing includes converting text to lowercase and cleaning and normalizing raw text data by eliminating extraneous whitespace, addresses, and special characters. Each article's title and text fields are then combined into a single, cohesive input

Tokenization and Sequence Padding: Using Keras Tokenizer, the preprocessed text was tokenized by turning it into an integer sequence. After that, these sequences are padded to a predetermined length so that the LSTM model can process them efficiently.

Model Architecture: A modified LSTM framework was developed using the Keras sequential API. The structure consists of an embedding layer, a bidirectional LSTM layer with dropout regularization, multiple dense classification layers, and a sigmoid output layer.

Model Training and Validation: The dataset is separated into training, validation, and testing sets. After the algorithm has been trained on the training data, its performance is tracked on a validation set to prevent overfitting. Callbacks such as EarlyStopping and ModelCheckpoint are used to optimize the training process.

Model Evaluation: Assessment of the trained model is carefully assessed the unseen test data using a range of metrics, including accuracy, precision, recall, F1-score, and AUC-ROC. The classification report and confusion matrix are developed in order to give an in-depth evaluation of the model's performance.

Model Saving and Deployment: The tokenizer, the trained model, and other required elements are stored for later use. To enable the classification of fresh, unseen news articles, a prediction function is developed.

3.1.3 Functional and Nonfunctional Requirements

Functional Requirements:

- The system shall provide an interface for users to upload their text file (in a format such as .txt).
- The system shall display the uploaded test back to the user for confirmation to show the result.
- The system shall process the uploaded image using the traditional model.
- The system shall display the predicted test result to the user.
- The system shall display the model's confidence score for those predictions of text data.
- The system shall provide an option for the user to reset the interface and upload new text data.

Nonfunctional Requirements:

- Performance: System must produce the outcome within 7 seconds
- Usability: The user interface must be user-friendly, and there is no pilot technical training needed for this model.
- Reliability: The user should be available and operational with high time spent consistently providing accurate predictions for test data within the scope of the trained classification.
- Compatibility: The application must be fully functional on all modern types of browsers.
- Scalability: The system should be capable of adding support for models without any major changes.

3.1.4 Data Flow Diagram

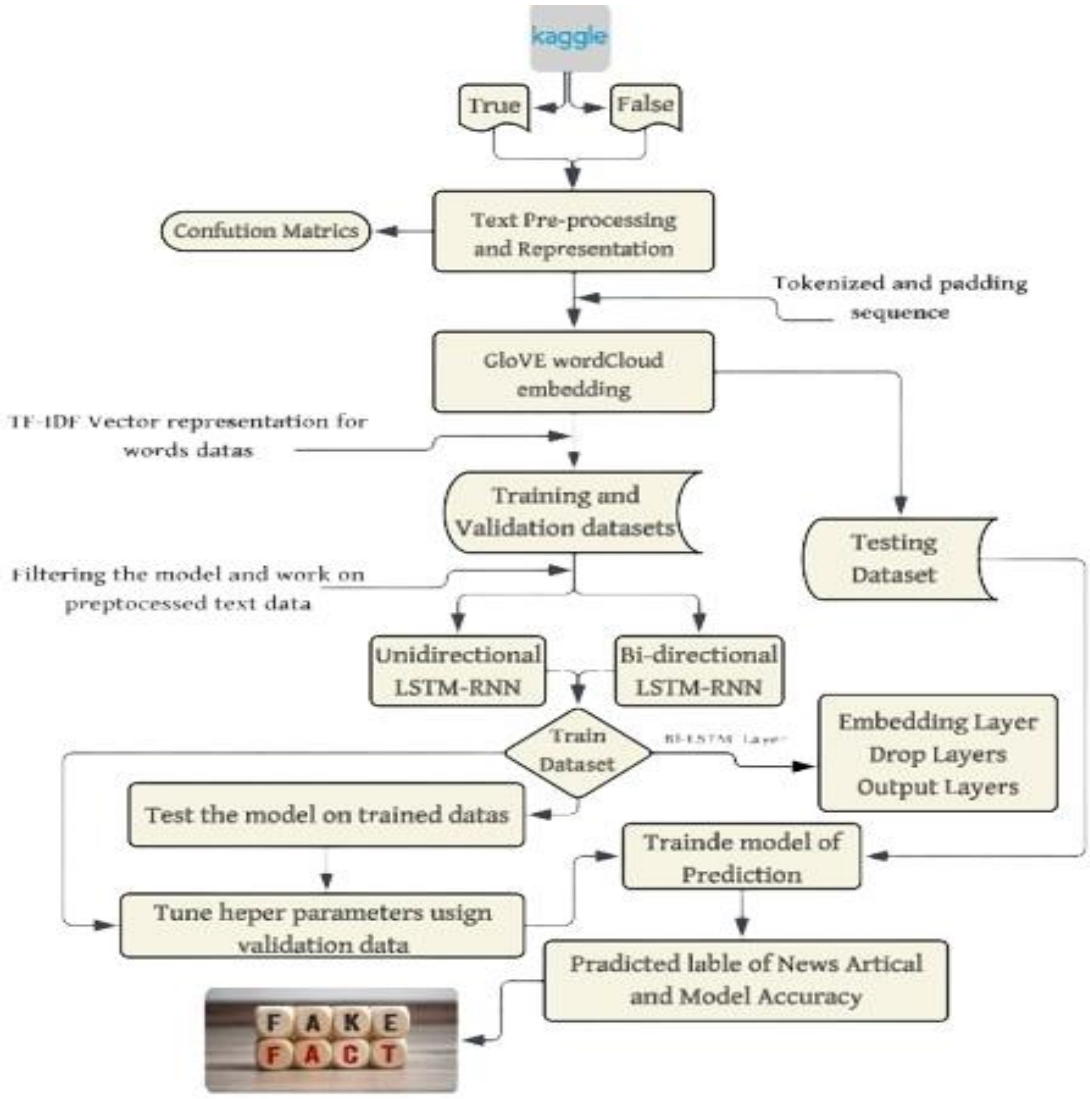


Figure 3.1: Data Flow Diagram

3.1.5 UI Design

Here is the UI design of our web applications:

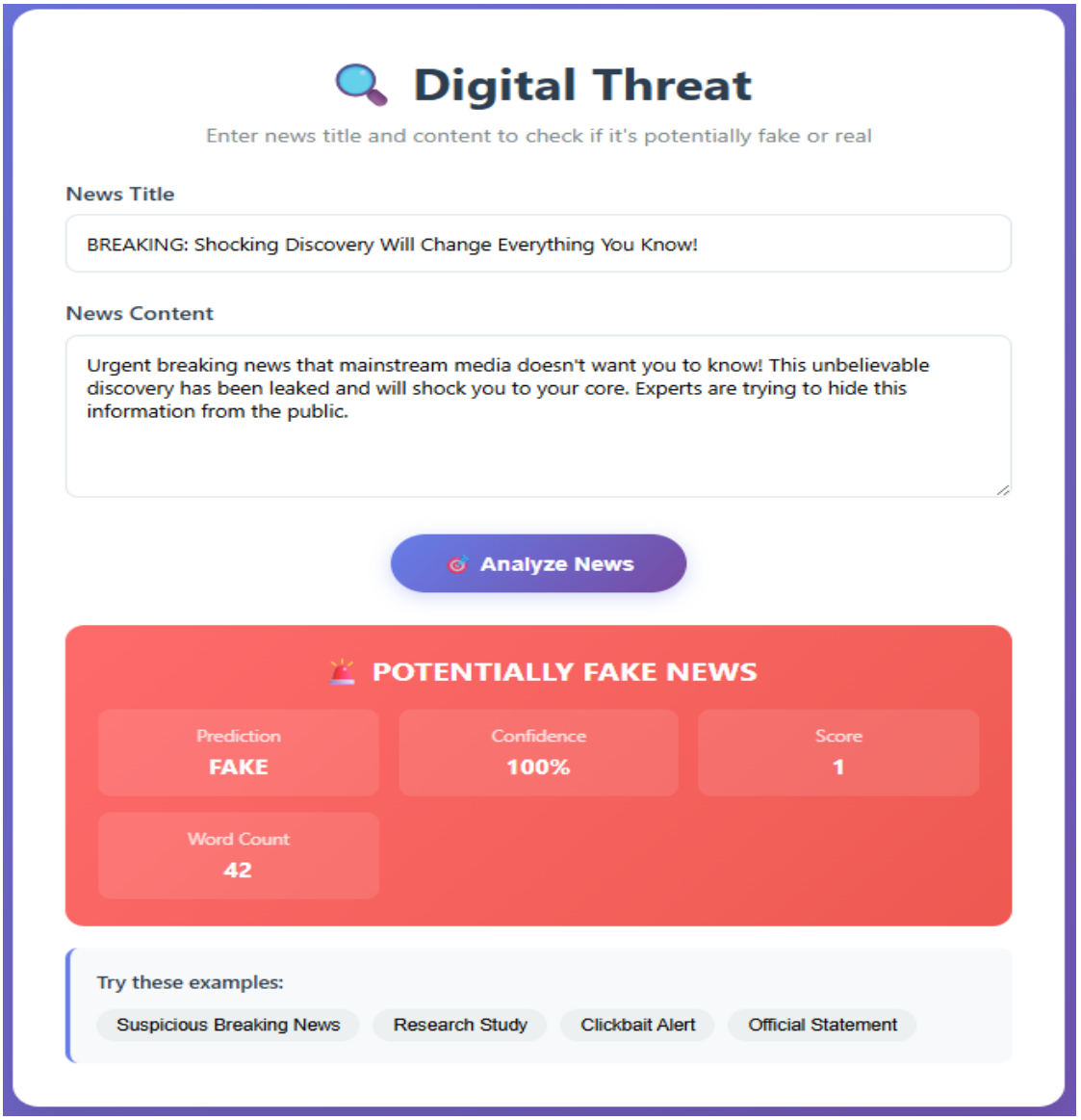


Fig. 3.2 "Fake News Threat Detection" in our Web-Application.

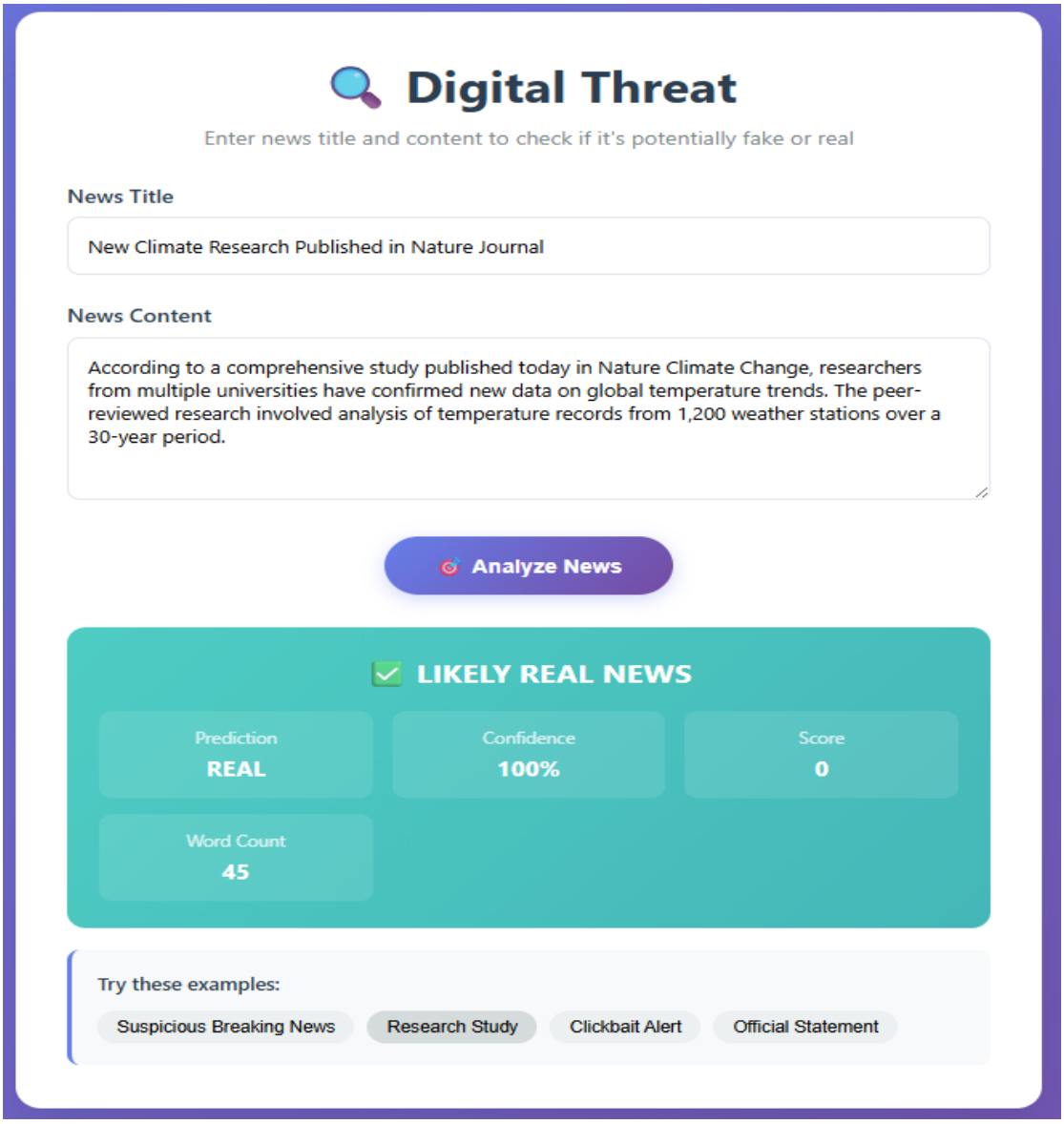


Fig. 3.3 "True News Detection In our Web-Application

3.2 Detailed Methodology and Design

The full pipeline, including preprocessing and augmentation steps, the LSTM model architecture approach, and the reason for the selection, are addressed throughout this section. This architecture of use model is discussed in this section. Also, including the details of all steps for a clear visualization of this research work.

3.2.1 Data Collection and Preparation:

For the case study, we created a combination dataset of misinformation detection forms from Kaggle. The dataset with true and false data is in there, and we merged the data with both the true (1) and false (0) datasets. A dataset of fake data and true data, which contains around 21418k true (1) data and 35029 false (0) data, is separated to combine with TRUE data, which contains 21418, and FAKE data, which contains 23503. This dataset was organized into a hierarchical folder structure, with each specialty represented by a subcategory. The leveler puts the text data in, and their owner can follow the results.

3.2.2 Preprocessing and Data Augmentation:

Preprocessing is a critical phase to standardize and enhance the raw text data, ensuring compatibility with the deep learning model and improving overall performance by addressing common issues such as poor text data labeling, Null value removing, and null value handling. The process begins with loading each text using a popular Python text data preprocessing library like TensorFlow to preprocess LSTE data. IF as text is not used in video data. It is conferred t-test data to maintain the consistency across the dataset, as the LSTM Vanilla and LSTM using TensorFlow expect three-channel inputs. Subsequence tests are a prerequisite to removing special characters to uniform the dimension of preprocessed textual data. The figure is showing the sample of textual representation of data.

3.2.3 Confusion Matrix:

The confusion matrix is a common tool for evaluating performance in machine learning. It summarized the classification result of the model in a simple table. Comparing actual output to predicted output. This shows how many predictions in each category. The four main metrics are True Positive (TP), where the model accurately identifies a news article as True; True Negative (TN), where it correctly identifies a news article as Fake; False Positive (FP), where it wrongly identifies a Fake article as True; and False Negative (FN), where it mistakenly marks a True article as Fake. By looking at these values, the confusion matrix helps calculate key performance measures like accuracy, precision, recall, and F1-score.

In our dataset there are two main classes, like true (4278) and false (4689) data, which highlight the result. Bi-directional LSTM accuracy is 0.9988, precision is 0.9986, recall is 0.9988, the F1-score is 0.9987, and AUC-ROC is 1.00. [**More about conformation matrix is page 26 and 27**.]

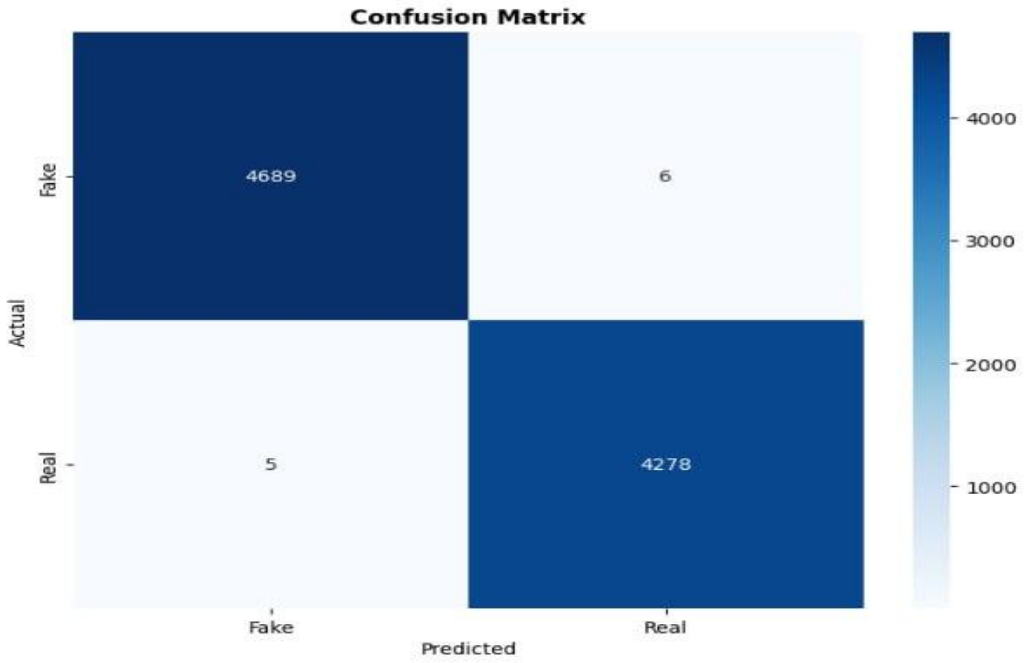


Fig.3.4 Confusion Matrices

3.2.4 GloVE and Word Cloud Embedding:

In this research paper, we use GloVe embedding to tune text from news articles into dense context and vectors. These vectors capture the meaning and structure of words. Words with similar meanings, such as "US," "said," and "United States," are located close to each other in Trump and Will. This arrangement helps LSTM and Bi-LSTM models understand the context for which they are used, whether for true or fake data. How many words are used in this section? At last, this framework provided a strong basis for an effective misinformation detection and classification system.

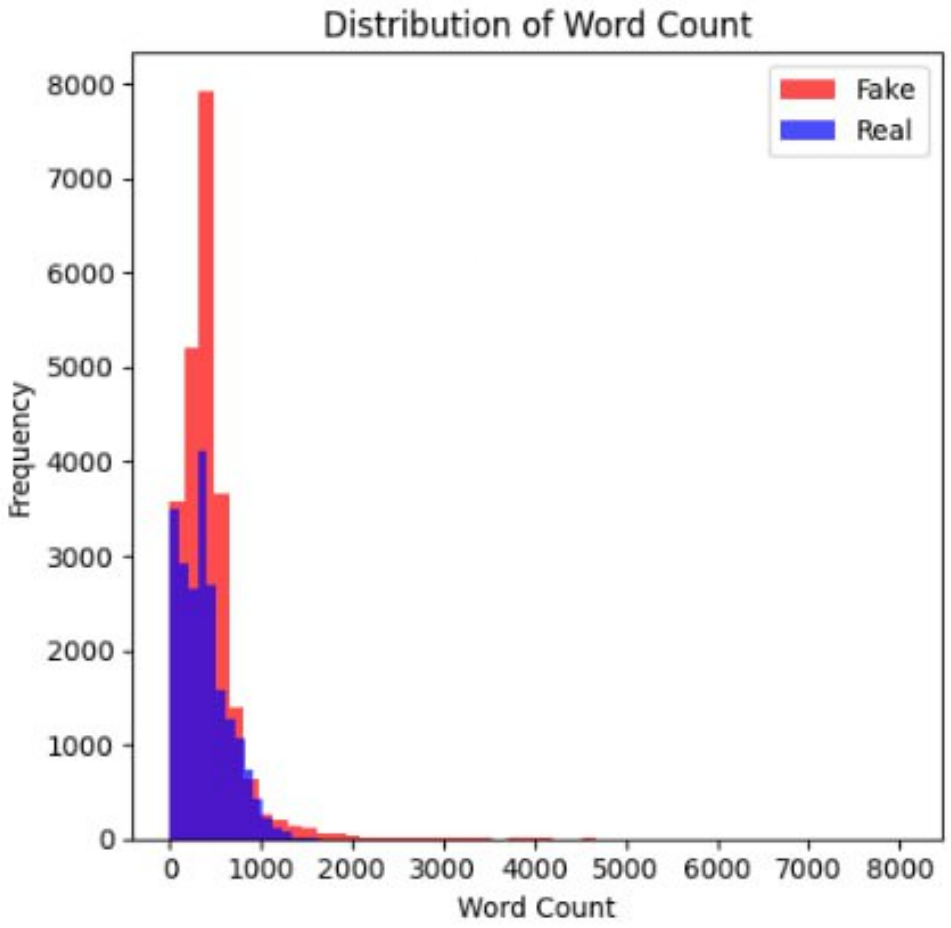


Fig. 3.5 Distribution of wordcount graph

3.2.5 Training and Validation:

This process of dataset is split into 80% for training and 20% for validation to set values for model performance on unseen data. The LSTM and Bi-LSTM models were trained using the Adam optimizer and binary cross entropy loss. Training happens over multiple epochs using mini batches. This approach helps ensure the update and prevent overfitting. We record metrics like accuracy, precision, recall, F1 score, a confusion matrix, and an ROC curve to evaluate training validation performance. This process can help to choose the best architecture for detecting and classifying misinformation news before deployment.

3.2.6 Filtering LSTM Models:

In this research paper we use both Unidirectional LSTM (Uni-LSTM) and Bidirectional LSTM (Bi-LSTM) architectures to detect and classify

misinformation but focus on bidirectional LSTM. In the uni-LSTM model the sequence is processed in one froward direction and keeps trace of past words through its hidden cell. Bi-directional processing processed the sequence in both forward and backward directions. In combination, information from both past and future words. In LSTM architecture, the embedding layer transforms words into dense vector representations, and the dropout layer prevents overfitting by randomly deactivating neurons during training. and finally, and finally, the output layer, implemented as dense layer with sigmoid activation, generates a probability score for binary classification like misinformation detection and classification

3.2.7 Model Architecture:

The model architecture is constructed using transfer learning from LSTM, a lightweight deep learning optimized for any kind of device. A Bidirectional LSTM model is designed using the Keras Sequential API. The architecture consists of an Embedding layer, a Bidirectional LSTM layer with dropout regularization, and a series of Dense layers for classification, culminating in a sigmoid output layer.

3.2.8 Training and Evaluation:

The dataset is split into training, validation, and testing sets. To avoid overfitting, per rmance is tracked on the validation set after the model has been trained on the training data. The training process is optimized using callbacks like EarlyStopping and ModelCheckpoint.

3.2.9 Training and Testing Model:

This process involves splitting the dataset into 80% for training and 20% for validation for model performance on unseen data. The LSTM and Bi-LSTM models trained using the Adam optimizer, binary cross-entropy loss. Mini batches used for training across some epochs. We can guarantee the update and avoid overfitting with this method. Performance metrics such as accuracy, precision, recall, F1-score, ROC curves, and confusion matrices are used to assess

the trained model on the test set. This technique demonstrates the Bi-LSTM's ability to effectively generalize and differentiate between authentic and fraudulent news articles.

3.2.10 Tune Hyper Using Validation Data:

To improve model performance, we tune hyperparameters using a validation dataset. We are seeking a system to adjust key parameters, such as learning rate, batch size, dropout rate, and number of epochs. If the model starts to overfit, we employ strategies such as early stopping. This process ensures that the final Bi-LSTM model strikes a good balance between accuracy and generalization before being tested on unseen data.

3.2.11 Training model and prediction label news of news and model accuracy:

After training the LSTM and BiLSTM models on the prepared dataset, the system could generate the predestined label for each news article, classifying the answer as either fake (0) or true (1). The predictions were compared with the actual labels from the test set to evaluate performance. The model had strong results, with evaluation metrics such as accuracy, precision, recall, and F1-score. The Bi-LSTM model notably outperformed the unidirectional LSTM, showing higher accuracy because it could capture contextual information from both past and future word sequences. At last, the train model not only provides reliable prediction but also achieves a high level of accuracy and effectiveness in detecting and predicting the misinformation news.

3.2.12 Model Evaluation and Execution

The trained model is rigorously evaluated on the unseen test data using a range of metrics, including accuracy, precision, recall, F1-score, and AUC-ROC. A confusion matrix and classification report are generated to provide a detailed analysis of the model's performance.

3.3 Project Plan

Table 3.1: Project Plan

Phase	Activities	Duration	Deliverables
Phase 1: Domain Selection	Contacted the supervisor and discussed the topic name.	Week 12 - week 14	Supervisor advice to work on this topic and deploy it.
Phase 2: Data Collection	Collect data from the open-source database like Kaggle.	Week 15 - week 17	Labeled all the text formats in the true and false dataset.
Phase 3: Data Preprocessing	Selected the target value and organized all the text in a proper format.	Week 18 - Week 22	Cleaned all dataset and text formats using the dataset.
Phase 4: Improving the text data with second preprocessing	Implementation, vectorization, and text classification.	Week 23 - Week 29	Categorized all textual datasets using similar values.
Phase 5: Model selection and validation	Train and test the different models, like LSTM TensorFlow, with different parameters.	Week 30 - Week 33	Trained the labeled and clean data and checked the validation and results.
Phase 6: Integrated Word2Vec and classified the model for training.	Work for the model data and classified data and add to the model selection.	Week 34 - Week 39	Showing the real data and testing the classified data.
Phase 7: Develop the Web browser application	Develop the web application, integrate the model, and show the result.	Week 40 - Week 48	Develop an application for the user. where the user can find the visualization of the result with his own eyes.

3.4 Task Allocation

Table 3.2: Task Allocation

Tasks	Weeks																		
	12	14	16	18	20	22	24	26	28	30	32	34	36	38	40	42	44	46	48
Data collection phase																			
Preprocess all the data																			
Model training																			
Create a application.																			

The tasks were divided between the two team members as follows:

- **Mir Saleh Ahmed:** Responsible for the literature review, data exploration and visualization, and writing the Introduction and Literature Review sections of the research paper.
- **Bikrom Adatya Roy:** Responsible for data preprocessing, model implementation and training, as well as writing the Methodology and Results and Discussion sections of the research paper.

Both team members collaborated on the overall project planning, model evaluation, and finalization of the research paper.

3.5 Summary

With the primary information collection, sophisticated preprocessing and augmentation for data enhancement, a lightweight yet potent LSTM vanilla architecture with custom layer comprehensive evaluation, and LSTM, this expanded approach offered a complete blueprint of the global data of the misinformation news detection system. The pipeline solves practical problems in this field while maintaining outstanding efficiency and an effortless dataset flow. In addition to supporting present goals, this framework establishes the foundation for future developments like sizable datasets of edge device deployments.

Chapter 4

Implementation and Results

The code implementation, the analysis model, and the results discussion are all covered in this chapter. We will give a brief synopsis of the subject in the final section.

4.1 Environment Setup

The deep learning system for misinformation detection was implemented using Python. We use deep learning models like LSTM and a number of Python libraries. TensorFlow and Keras were used to build and train the deep learning models. NumPy and Pandas were used for preprocessing, visualization, and numerical calculations.

The Kaggle platform's powerful computational resources were used to develop and test this project in its entirety. In a Kaggle notebook environment with two NVIDIA Tesla T4 GPUs (T4x2), the deep learning model was constructed and assessed. This robust configuration was crucial for effectively handling the sizable dataset and accelerating the training of the intricate Bidirectional LSTM model. TensorFlow (version 2.18.0), Keras (integrated with TensorFlow), Pandas for data management, NumPy for numerical computations, Scikit-learn for machine learning, Matplotlib and Seaborn for visualization, and NLTK for natural language processing tasks were among the essential libraries utilized in the Python 3.x environment. To guarantee consistency and reproducibility, all dependencies were set up in the Kaggle environment.

4.2 Testing and Evaluation

The gathered dataset had been meticulously divided to enable thorough evaluation and testing when the data preprocessing and model building stages. To maintain the original class distribution across all subsets, the combined dataset—which included 44,898 news articles—was split into training,

validation, and test sets using a stratified sampling technique. This is how the final distribution of data appeared:

Training Set: 28,728 samples (approximately 64% of the total data)

Validation Set: 7,183 samples (approximately 16% of the total data, used during training for hyperparameter tuning and early stopping)

Test Set: 8,978 samples (approximately 20% of the total data, held out for final, unbiased evaluation)

The model was trained for a maximum of 20 epochs, with an EarlyStopping callback configured to monitor the validation loss (val_loss) and halt training if no improvement was observed for 3 consecutive epochs. A ModelCheckpoint callback was also utilized to save the best performing model based on validation accuracy (val_accuracy). The training process utilized the Adam optimizer and binary cross-entropy as the loss function. Monitoring accuracy, precision, and recall as performance metrics.

After training, the model's performance was tested on an unseen test set. AUC-ROC, F1-score, recall, accuracy, precision, and other common classification metrics were assessed. A confusion matrix that offers a thorough summary of true positives, true negatives, false positives, and false negatives was developed in order to better understand how the algorithm classifies data.

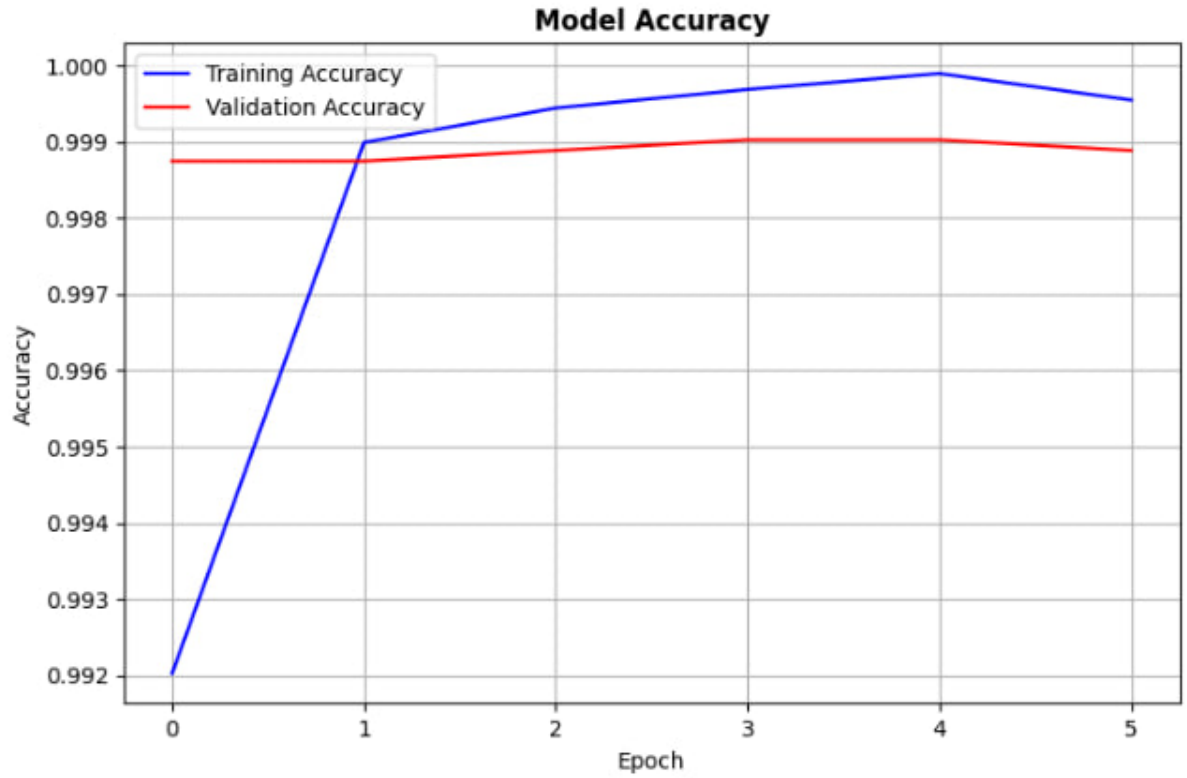


Figure 4.1: Training VS Validation Accuracy

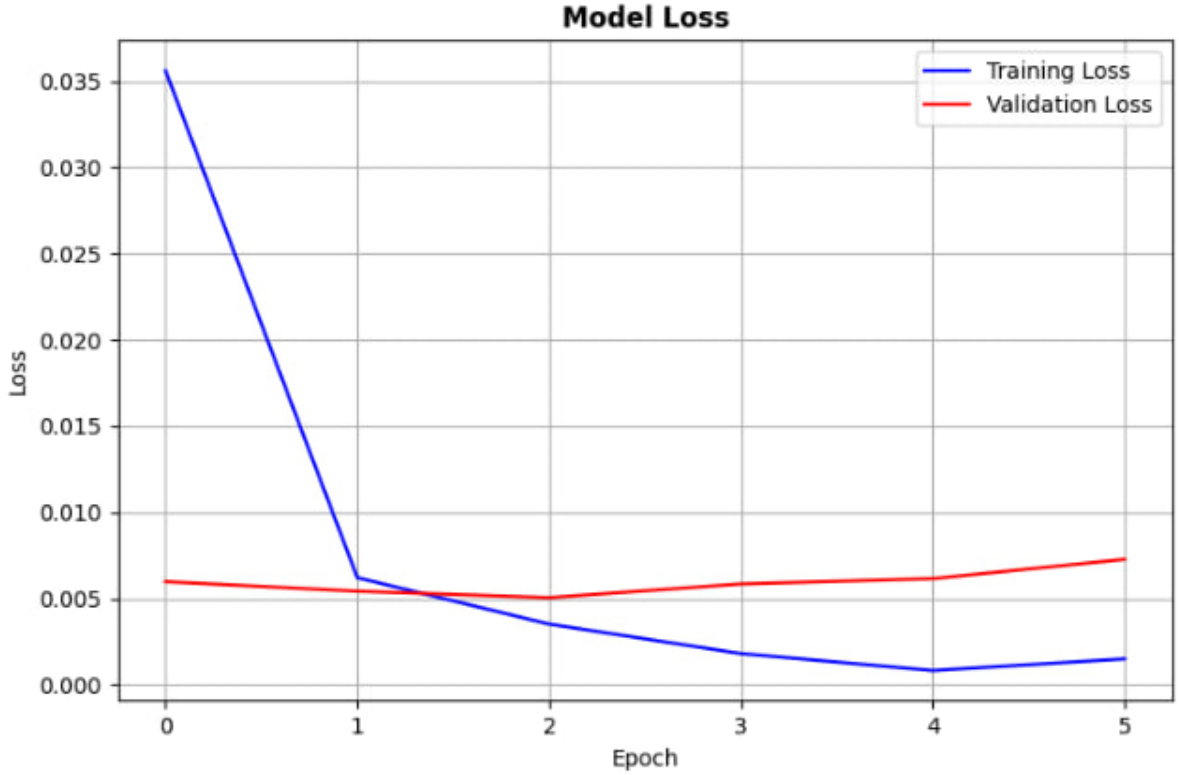


Figure 4.2: Training VS Validation Loss

4.3 Results and Discussion

The outcomes of the experiment demonstrate how well the Bidirectional LSTM model detects false information. Its robustness and effectiveness are demonstrated by its strong results across all metrics.

Table 4.1: Model Evaluation

Model	Accuracy	Precession	Recall	F1-Score	AUC-ROC
Bidirectional LSTM	0.9988	0.9986	0.9988	0.9987	1.00

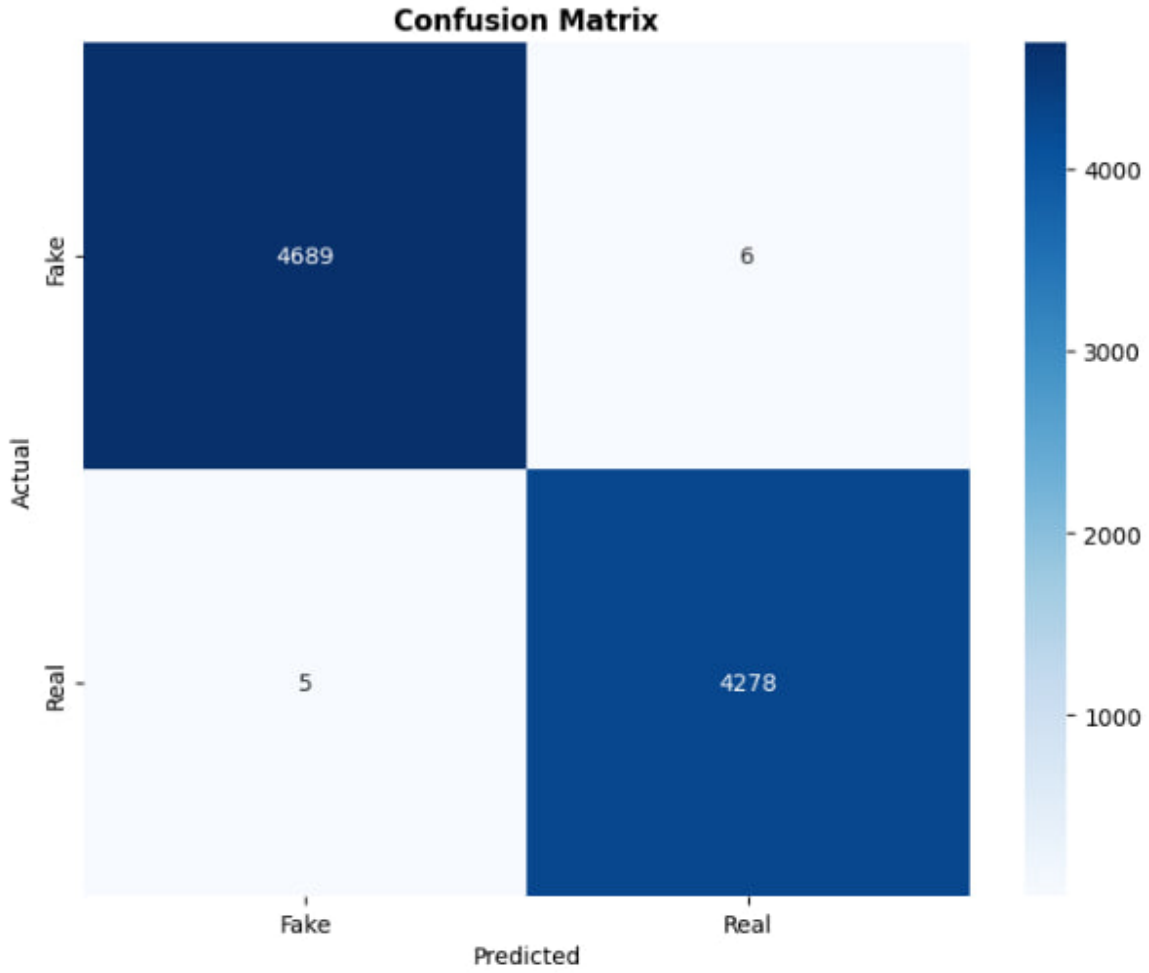


Figure 4.3: Confusion Matrix

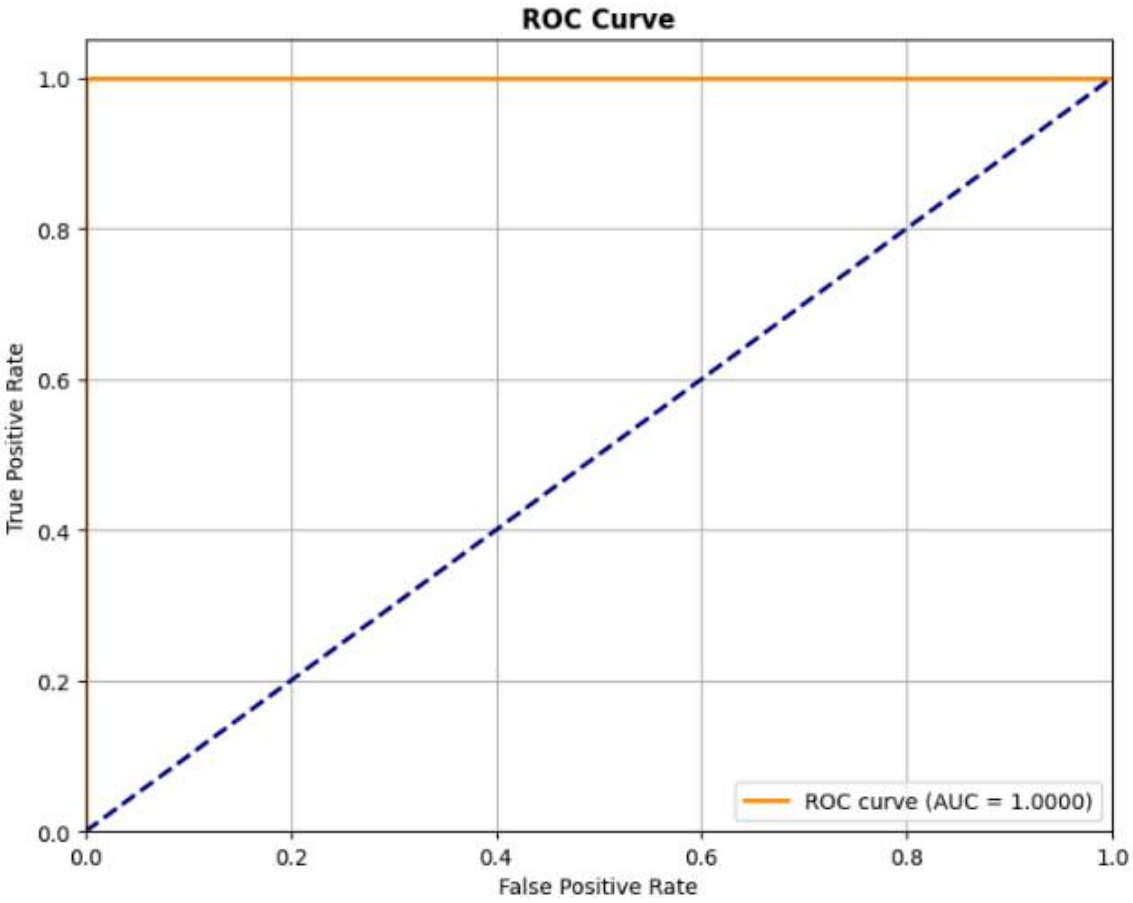


Figure 4.4: ROC Curve Distribution

- **Accuracy:** A remarkable test accuracy of 0.9988 (99.88%) was attained by the model. This implies that nearly all of the test collection's articles were correctly classified as either genuine or fake. This extraordinarily high accuracy shows how well the model can distinguish between the two classes.
- **Precision:** A very small percentage of false-positive results are indicated by a test precision of 0.9986. This suggests that any time a framework suggests a paper is false, it's most likely been. This metric is crucial for identifying misinformation because inaccurately classifying precise data as fake may destroy confidence in trustworthy sources.
- **Recall:** With a strong test recall of **0.9988**, the model showed a very low false negative rate. This suggests that the model misses only a few real false news reports when is exceptionally good at spotting them. To make

sure that as much false information is reported as possible, high recall is crucial.

- **F1-Score:** The model's F1-Score of 0. 9987 indicates that recall and accuracy are now well-balanced. This integrated metric confirms the model's general efficacy and capacity to perform well in both kinds.
- **AUC-ROC:** The model's remarkable AUC-ROC score of 1.0000 suggests it is an ideal classifier, capable of recognizing between positive and negative classes with no overlap in their score distributions. This implies that the method has learnt exceptionally discriminating features from the data.

The confusion matrix provides a granular view of the model's performance:

- **True Negatives (Correct Fake):** 4689 articles were correctly identified as fake news.
- **False Positives (False Real):** Only 6 articles were incorrectly classified as real news when they were actually fake. This aligns with the high recall we observing.
- **False Negatives (False Fake):** Only 5 articles were incorrectly classified as false news although its actually real. This aligns with the high recall we observing.
- **True Positives (Correct Real):** 4278 articles were correctly identified as real news.

The model's high reliability and capacity to reduce errors in the two directions is demonstrated through the extremely small instances of incorrect positives and false negatives. As the results report indicates out, the system's findings validate its outstanding performance, high recall, and high precision, making it a very useful tool for misinformation detection.

4.4 Summary

The implementation of the fake news detection system, including the computational environment and the stringent testing and evaluation procedures, has been covered in detail in this chapter. With nearly flawless scores on every important evaluation metric, the results clearly demonstrate how well the Bidirectional LSTM model works. The effectiveness of the suggested approach is validated by the high accuracy, precision, recall, F1-score, and AUC-ROC as well as the small number of incorrect classifications in the confusion matrix. According to these results, the created system is a very dependable and strong way to stop false information from spreading online. This study's main contributions, limitations, and potential future study goals will all be covered in the upcoming chapter.

Chapter 5

Engineering Standards and Design Challenges

The engineering standards and design issues that arose during the project's advancement are highlighted in this chapter. While resolving design issues guarantees the system stays stable and functional, maintaining standards compliance contributes to the preservation of dependability, interoperability, and user trust.

5.1 Compliance with the Standard

All designed structure's quality, dependability, as well connectivity depends heavily on adherence to pertinent engineering standards. This section outlines the standards considered and their rationale for selection within the context of the misinformation detection system.

5.1.1 Software Standards

For software development, adherence to established coding practices and software engineering principles is paramount. While no specific formal standards (e.g., ISO/IEC 12207) were strictly enforced due to the research nature of the project, best practices in Python programming, TensorFlow/Keras development, and data science workflows were consistently applied. This includes modular code design, clear documentation within the code, version control (implicitly through Kaggle's notebook versioning), and the use of well-tested libraries. These practices ensure maintainability, readability, and reproducibility of the software components.

5.1.2 Hardware Standards

The project primarily utilized cloud-based GPU resources provided by Kaggle (NVIDIA Tesla T4 GPUs). As such, direct hardware design standards were not applicable. However, the selection of a robust and high-performance computing environment indirectly aligns with the need for efficient processing of large datasets and complex deep learning models, which is a de facto standard for such

computationally intensive tasks. The use of industry-standard GPU hardware ensures compatibility and leverages optimized libraries for deep learning operations.

5.1.3 Communication Standards

During the development of this project, several challenges were encountered:

- Data Quality and Availability: Collecting clean, unbiased, and diverse datasets posed a challenge as misinformation often spreads in varied formats.
- System Integration: Combining multiple modules (data preprocessing, feature extraction, and model training) required careful synchronization.
- Performance Optimization: Balancing accuracy with processing time was challenging, especially when deploying hybrid models.
- Security Concerns: Ensuring the system was resistant to adversarial attacks and data manipulation required additional safeguards.
- Scalability: Designing a system that can handle increasing amounts of real-time data without performance degradation.

5.2 Impact on Society, Environment and Sustainability

The development and deployment of a misinformation detection system carry significant implications across various societal, environmental, and ethical dimensions. Understanding these impacts is crucial for responsible innovation.

5.2.1 Impact on Life

This text classification system provides practical utility for users. authentic users. It reduced the hassle of being misinformed by any mainstream media and their comment sections. All the social media are a war zone. In an educational context, it can serve as a digital tool for students and researchers learning about how misinformation affects our daily lives. The lightweight Web application prototype demonstrates the approach's usability for on-the-spot identification.

The integrated LSTM visualization enhances interpretability, allowing users to see the discriminative regions the model relies on. This transparency builds confidence in the system predictions, a crucial factor for adoption in real-world

applications, and the misinformation that is used in this day is very widespread. propaganda and dirty politics.

5.2.2 Impact on Society & Environment

From an environmental perspective, the system reduces dependence on printed identification guides, minimizing paper use. By facilitating accurate species identification and supporting sustainable misinformation detection and prediction, they are creating.

The choice of lightweight architectures like LSTM also minimizes the energy consumption during inference, making the system more sustainable compared to resource-intensive models. Deploying the system via shared cloud infrastructure can further reduce the carbon footprint by leveraging optimized, energy-efficient data centers.

5.2.3 Ethical Aspects

Ethical principles guide the entire development cycle. The dataset was collected from the Kaggle open-source dataset. Only non-send-time texts were included, preserving privacy. Bias mitigation was considered by collecting approximately two types of data, true and false, in the 45k data and special classes. Nevertheless, the system acknowledges the project also embraces transparency; all processing steps, training configurations, and evaluation results were documented supporting open, reproducible research.

5.2.4 Sustainability Plan

A sustainability roadmap underpins the long-term vision of the project.
Key strategies include:

- Periodic dataset updates with new text and classified data
- Continuous fine-tuning of deciding models using reproductive scripts
- Model comparison and deployment on the web browser and mobile set.
- Collaboration with other authorities and universities to expand the textual dataset is a validation model in diverse environments.

5.3 Project Management and Financial Analysis

Effective project management and a realistic financial analysis are crucial for the successful development and deployment of any engineering project.

The project was managed using an agile-inspired approach, with the work divided into distinct phases as outlined in the project plan (Chapter 3). Regular meetings between the two team members ensured effective communication, collaboration, and timely progress. The use of Kaggle notebooks facilitated a collaborative coding environment and version control. The project timeline was adhered to, with each phase completed within the allocated timeframe.

Table 5.1: Financial Analysis

Service Description	**Estimated Cost (৳)**	**Explanation**
Dataset Collection & Preparation	5,000	Data cleaning, labeling, sourcing Bangla news data
Hardware (GPU/CPU) Usage	15,000	Computer hardware, GPU server, or personal setup
Cloud Server (if used)	12,000	Google Colab Pro, AWS, Azure, or local VM
Software Licenses & Tools	3,000	Paid libraries, software (if required), IDE
Model Training & Development	10,000	Model design, code development, tuning
Model Saving & Deployment	4,000	Saving model, web or other deployment costs
Research, Analysis & Documentation	6,000	Report writing, graphs, academic documentation
Total Estimated Cost	55,000	**Overall implementation cost (approximate)**

5.4 Complex Engineering Problem

This section examines the intricate engineering issues that arose during the creation of the misinformation detection system and connects them to established classifications for problem-solving and engineering tasks.

5.4.1 Complex Problem Solving

Building an efficient disinformation detector is a complicated and complex engineering task. The primary barriers to the different types of complex engineering problems (EPs) are mapped in the following table.

Table 5.2: Mapping with Complex Engineering Problem.

EP1 Dept of Knowledge	EP2 Range Of Conflicting Requirements	EP3 Depth of Analysis	EP4 Familiarity of Issues	EP5 Extent of Applicable Codes	EP6 Extent Of Stake-holder Involvement	EP7 Interdependence
✓	✓	✓	✓			✓

Justification for Selected EPs:

- **EP1 (Depth of Knowledge):**
- **EP2 (Range of Conflicting Requirements):**
- **EP3 (Depth of Analysis):**
- **EP4 (Familiarity of Issues):**
- **EP7 (Interdependence):**

Mapping with Knowledge Profile

This section highlights the knowledge areas essential for addressing the depth of knowledge required for connecting the general issue and EP1 to the Learning Profile.

Table 5.3: Mapping knowledge Profile.

K1 Natural Science	K2 Mathematics	K3 Engineering Fundamentals	K4 Specialist Knowledge	K5 Engineering Design	K6 Engineering Practice	K7 Comprehension	K8 Research Literature
	✓	✓	✓	✓	✓	✓	✓

Justification for Selected Knowledge Profiles:

- **K2 (Mathematics)**
- **K3 (Engineering Fundamentals)**
- **K4 (Specialist Knowledge)**
- **K5 (Engineering Design)**
- **K6 (Engineering Practice)**
- **K7 (Comprehension)**
- **K8 (Research Literature)**

5.4.2 Engineering Activities

In this section, we will provide a mapping with engineering activities. For each mapping add subsections to put.

Mapping with Complex Engineering Activities

This section is designed to map the overall problem and EA's.

Table 5.4: Mapping with Complex Engineering Activities.

EA1 Range of resources	EA2 Level of Interaction	EA3 Innovation	EA4 Consequences for society and environment	EA5 Familiarity
✓	✓		✓	

Justification for Selected Engineering Activities:

- **EA1 (Range of Resources)**.
- **EA2 (Level of Interaction):**
- **EA4 (Consequences for Society and Environment):**

5.5 Summary

The engineering difficulties, societal effects, project management, and standards that were involved in developing the misinformation detection system are all covered in detail in this chapter. It exhibits a dedication to software engineering principles, a thorough comprehension of the moral and societal ramifications,

and a methodical approach to resolving challenging engineering issues. The analysis emphasizes the study's intricacy as well as the crucial engineering choices and actions that made it possible for it to succeed.

Chapter 6

Conclusion

This chapter provides a thorough examination of LSTM TensorFlow and a deep learning classification system for identifying fake news.

6.1 Summary

This study developed and assessed a trustworthy fake news detection system using TensorFlow's Bidirectional Long Short-Term Memory (LSTM) neural networks. With careful data preparation, enhanced feature engineering, and a well-designed model, we were able to distinguish between authentic and fraudulent news articles with accuracy. With test results showing 99.88% accuracy, 99.86% precision, 99.88% recall, and an F1-score of 99.87%, the system demonstrated exceptional performance. An almost flawless AUC-ROC score of 1.0000 demonstrated its strong category-distinction capabilities. The model's dependability was further supported by the confusion matrix analysis, which revealed relatively few false positives and false negatives. A useful weapon against fake news in the digital age, these findings show that deep learning, and LSTMs in particular, can successfully recognize intricate language patterns and contextual information linked to misinformation. Overall, this study lays the groundwork for future developments in automatic misinformation detection by promoting the use of deep learning and natural language processing for social betterness.

6.2 Limitation

Notwithstanding the encouraging outcomes, this study has some drawbacks. First off, in order to detect misleading information, the study only looked at textual content. Multimodal components that are frequently found in real-world misinformation, such as misleading images, videos, and audio, were not taken into consideration in this study. Second, despite the dataset's size, it might not adequately reflect how dynamic and constantly evolving disinformation tactics are. The long-term efficacy of the model may be impacted if new forms of deceptive language and content don't receive regular retraining. Thirdly, the

interpretability of the model, a common problem in deep learning, can still be improved. Even though the model performs well, understanding exactly why it makes certain classifications could help us better comprehend the characteristics of fake news and enhance system trust.

6.3 Future Work

Building upon the success of this research, several avenues for future work can be explored to further enhance the capabilities of misinformation detection systems further:

- **Multimodal Misinformation Detection:** The detection of sophisticated fake news could be significantly enhanced by analysis with visual and auditory cues. Convolutional neural networks (CNNs) for image analysis or other deep learning architectures for video and audio processing could be combined with LSTM models in future research.
- **Real-time Detection and Adaptation:** For practical implementation, it would be essential for the real-time identification of new and to update the model frequently to accommodate new misleading trends. For model retraining, this could entail continuous integration/continuous deployment (CI/CD) pipelines or active learning techniques.
- **Explainable AI (XAI) for Misinformation Detection:** Techniques such as LIME (Local Interpretable Model-agnostic Explanations) or SHAP (SHapley Additive exPlanations) could be applied to highlight the specific words or phrases that contribute most to a fake news classification, thereby increasing user trust and providing insights for human fact-checkers.
- **Cross-lingual Misinformation Detection:** Extending the model's capabilities to detect misinformation in multiple languages would address the global nature of this problem. This would involve leveraging multilingual embeddings and adapting the preprocessing and modeling techniques for different linguistic structures.
- **Integration with Social Media Platforms:** Exploring the ethical and technical challenges of integrating such detection systems directly into social media platforms could provide a proactive defense against the rapid spread of misinformation.

By addressing these areas, future research can contribute to the development of even more robust, adaptable, and transparent misinformation detection systems, further safeguarding the integrity of information in the digital age.

References

1. Chopra, Y., Kaushik, P., Rathore, S. P. S., & Kaur, P. (2023). Uncovering Semantic Inconsistencies and Deceptive Language in False News Using Deep Learning and NLP Techniques for Effective Management. channels, 1, 2.DOI: https://doi.org/10.17762/ijritcc.v11i8s.7256
2. Alladi, S., & Bharathi, B. (2025, May). Cognitext@ DravidianLangTech2025: Fake News Classification in Malayalam Using mBERT and LSTM. In Proceedings of the Fifth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages (pp. 361-365). No DOI number available.
3. Raza, A., Usmani, R. S. A., & Khan, S. U. R. (2024). A Hybrid Deep Learning Based Fake News Detection System Using Temporal Features. The Asian Bulletin of Big Data Management, 4(02), 264-273.https://doi.org/ 10.62019/abbdm. v4i02.179
4. Singh, P., Srivastava, R., Rana, K. P. S., & Kumar, V. (2023). SEMI-FND: Stacked ensemble based multimodal inferencing framework for faster fake news detection. Expert systems with applications, 215, 119302.DOI not mentioned.
5. Madani, M., Motameni, H., & Roshani, R. (2024). Fake news detection using feature extraction, natural language processing, curriculum learning, and deep learning. International Journal of Information Technology & Decision Making, 23(03), 1063-1098.DOI: 10.1142/S0219622023500347
6. Keya, A. J., Shajeeb, H. H., Rahman, M. S., & Mridha, M. F. (2023). FakeStack: Hierarchical Tri-BERT-CNN-LSTM stacked model for effective fake news detection. Plos one, 18(12), e0294701. https://doi.org/10.1371/journal.pone.0294701
7. ADEBAYO, P. O., OLADIPO, I. D., ABDULRAHEEM, M., BALOGUN, G. B., & TOMORI, A. R. FAKE NEWS DETECTION MODEL USING ENSEMBLE BI-DIRECTIONAL LSTM AND SUPPORT VECTOR MACHINES. DOI: 10.36108/ojeit/5202.10.0150

8. Das, R. K., & Dodge, J. (2025). Fake news detection after llm laundering: Measurement and explanation. arXiv preprint arXiv:2501.18649. https://github.com/rupakdas18/ Fake-News-Detection-After-LLMLaundering
9. Ghafoor, A. A., & Aish, M. A. (2024). Enhanced fake news detection with Bi-LSTM networks and TF-IDF. Journal of Innovative Computing and Emerging Technologies, 4(2).DOI not mention
10. Cao, P., Shukla, P. K., Shukla, P. K., Bhatia Khan, S., Alojail, M., & Ramtiyal, B. (2025). Enhancing fake news detection for Sustainable Supply Chain Management using DistilBERT-based multi-stacked LSTM approach. Enterprise Information Systems, 2538023.https://doi.org/10.1080/17517575.2025.2538023
11. Li, Q., Fu, F., Li, Y., Wisassinthu, B., Chansanam, W., & Boongoen, T. (2025). Fake News Detection with Deep Learning: Insights from Multi-dimensional Model Analysis. Journal of Computational and Cognitive Engineering.DOI: 10.47852/bonviewJCCE52026051
12. E. Almandouh, M., Alrahmawy, M. F., Eisa, M., Elhoseny, M., & Tolba, A. S. (2024). Ensemble based high performance deep learning models for fake news detection. Scientific Reports, 14(1), 26591.https://doi.org/10.1038/s41598-024-76286-0
13. Keya, A. J., Shajeeb, H. H., Rahman, M. S., & Mridha, M. F. (2023). FakeStack: Hierarchical Tri-BERT-CNN-LSTM stacked model for effective fake news detection. Plos one, 18(12), e0294701.https://doi.org/10.1371/journal.pone.0294701

213-15-4244

ORIGINALITY REPORT

24%	18%	11%	16%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Tilburg University Student Paper	5%
2	Submitted to Daffodil International University Student Paper	3%
3	QiuPing Li, Fen Fu, Yinjuan Li, Bhunnisa Wisassinthu, Wirapong Chansanam, Tossapon Boongoen. "Fake News Detection with Deep Learning: Insights from Multi-dimensional Model Analysis", Journal of Computational and Cognitive Engineering, 2025 Publication	2%
4	www.mdpi.com Internet Source	1%
5	S.P. Jani, M. Adam Khan. "Applications of AI in Smart Technologies and Manufacturing", CRC Press, 2025 Publication	1%
6	Submitted to Asia Pacific University College of Technology and Innovation (UCTI) Student Paper	1%
7	"Proceeding of the 2nd International Conference on Machine Intelligence and Emerging Technologies", Springer Science and Business Media LLC, 2025 Publication	<1%

22% detected as AI

The percentage indicates the combined amount of likely AI-generated text as well as likely AI-generated text that was also likely AI-paraphrased.

Caution: Review required.

It is essential to understand the limitations of AI detection before making decisions about a student's work. We encourage you to learn more about Turnitin's AI detection capabilities before using the tool.

Detection Groups

14 AI-generated only 8%
Likely AI-generated text from a large-language model.

13 AI-generated text that was AI-paraphrased 14%
Likely AI-generated text that was likely revised using an AI-paraphrase tool or word spinner.

Frequently Asked Questions

How should I interpret Turnitin's AI writing percentage and false positives?
The percentage shown in the AI writing report is the amount of qualifying text within the submission that Turnitin's AI writing detection model determines was either likely AI-generated text from a large-language model or likely AI-generated text that was likely revised using an AI paraphrase tool or word spinner.

False positives (incorrectly flagging human-written text as AI-generated) are a possibility in AI models.

AI detection scores under 20%, which we do not surface in new reports, have a higher likelihood of false positives. To reduce the likelihood of misinterpretation, no score or highlights are attributed and are indicated with an asterisk in the report (*%).

The AI writing percentage should not be the sole basis to determine whether misconduct has occurred. The reviewer/instructor should use the percentage as a means to start a formative conversation with their student and/or use it to examine the submitted assignment in accordance with their school's policies.

What does 'qualifying text' mean?
Our model only processes qualifying text in the form of long-form writing. Long-form writing means individual sentences contained in paragraphs that make up a longer piece of written work, such as an essay, a dissertation, or an article, etc. Qualifying text that has been determined to be likely AI-generated will be highlighted in cyan in the submission, and likely AI-generated and then likely AI-paraphrased will be highlighted purple.

Non-qualifying text, such as bullet points, annotated bibliographies, etc., will not be processed and can create disparity between the submission highlights and the percentage shown.