Bachelor's Thesis in Computer Science
May 2025


# Comparative Analysis of LSTM, GRU, and BERT Models for Fake News Detection

**Geetha Krishna Venkatesh Maroju**
**Sai Nandu Posina**

Faculty of Computer Science, Blekinge Institute of Technology, 371 79 Karlskrona, Sweden

This thesis is submitted to the Faculty of Computer Science at Blekinge Institute of Technology in partial fulfillment of the requirements for the degree of Bachelor Thesis in Computer Science. The thesis is equivalent to Weeks weeks of full-time studies.

The authors declare that they are the sole authors of this thesis and that they have not used any sources other than those listed in the bibliography and identified as references. They further declare that they have not submitted this thesis at any other institution to obtain a degree.

**Contact Information:**
Author(s):
Geetha Krishna Venkatesh Maroju
E-mail: gema24@student.bth.se

Sai Nandu Posina
E-mail: sapo24@student.bth.se

University advisor:
Dr. Angelova Milena
Department of Computer Science

Faculty of Computer Science
Blekinge Institute of Technology
SE–371 79 Karlskrona, Sweden

Internet : www.bth.se
Phone : +46 455 38 50 00
Fax : +46 455 38 50 57

# Abstract

**Background:** The dissemination of misinformation through online platforms is a serious concern in today's information-based world. As it has become easier to publish and share information online, fake news has emerged as a critical threat, influencing public perception and distorting facts. It is therefore important to correctly identify fake news in order to maintain the integrity of information and ensure public awareness.

**Objectives:** The objective of this study is to perform a comparative analysis of the Deep Learning (DL) models namely Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), and Bidirectional Encoder Representations from Transformers (BERT) for fake news detection. By evaluating the performance of these models using metrics such as accuracy, precision, recall, and F1-score, the study aims to identify the most effective method for detecting misinformation in text data.

**Methods:** The research adopts an experimental approach, by training and testing various DL models on a labeled fake news dataset sourced from Kaggle. Preprocessing steps such as tokenization and sequence padding are applied to prepare the text data for model input. The LSTM and GRU models are implemented using recurrent neural network layers, while BERT is employed using transfer learning techniques. Each model is evaluated on the same dataset to ensure a fair and consistent comparison.

**Results:** The evaluation indicates that transformer-based models perform significantly better than recurrent neural networks (RNNs) in fake news detection. The BERT model achieved the highest accuracy, reaching 99% among the evaluated models. The LSTM and GRU models achieved approximately 98% and 93% accuracy, respectively. The experimental results highlight the effectiveness of contextual word embeddings and multi-head attention mechanisms in capturing complex textual patterns.

**Conclusions:** This work demonstrates the effectiveness of the BERT model compared to conventional RNNs for fake news detection. By leveraging contextual understanding of textual data, BERT proves to be a robust tool for detecting misinformation with high accuracy. These results highlight the importance of transformer-based methods in constructing more stable and trustworthy fake news detection systems.

**Keywords:** Fake news detection; Deep learning; LSTM; GRU; BERT; Natural Language Processing; Transformer models.

# Acknowledgments

We would like to express our sincere gratitude to our supervisor, Dr. Angelova Milena, for her valuable guidance, continuous support, and motivation throughout the course of our thesis. Her expertise, insightful feedback, and unwavering support have been instrumental in shaping the direction and quality of our research.

We are also deeply thankful to our families and friends for their unfailing support, patience, and encouragement throughout this academic journey. Their support has been a constant source of strength and inspiration.

This thesis is a mirror of the collective support, wisdom, and encouragement we have received from everyone involved.

**Authors:**

Geetha Krishna Venkatesh Maroju

Sai Nandu Posina

# Contents

# List of Figures

# List of Tables

# List of Acronyms

| | |
|---|---|
| **AI** | Artificial Intelligence |
| **DL** | Deep Learning |
| **ML** | Machine Learning |
| **NLP** | Natural Language Processing |
| **LSTM** | Long Short-Term Memory |
| **GRU** | Gated Recurrent Unit |
| **BERT** | Bidirectional Encoder Representations from Transformers |
| **RNNs** | Recurrent Neural Networks |
| **ANNs** | Artificial Neural Networks |
| **FNN** | Feedforward Neural Network |
| **TF-IDF** | Term Frequency-Inverse Document Frequency |
| **SVM** | Support Vector Machine |
| **XAI** | Explainable Artificial Intelligence |
| **LIME** | Local Interpretable Model-Agnostic Explanations |
| **SHAP** | SHapley Additive exPlanations |
| **GPUs** | Graphics Processing Units |
| **TPUs** | Tensor Processing Units |
| **BOWs** | Bag of Words |

# Chapter 1

# Introduction

In the digital era, information dissemination has become nearly instantaneous with the help of online platforms, particularly social media. While these platforms have democratized content sharing and revolutionized public communication, they have also unintentionally become fertile ground for the proliferation of false, misleading, or intentionally deceptive information commonly known as fake news [6]. Fake news has severe consequences, from misinforming the public and politically manipulating democratic processes to endangering public health and safety, as seen during the COVID-19 pandemic [45]. Therefore, the detection and prevention of fake news have become an urgent research challenge.

Manual fact checking is neither scalable nor effective with the massive volume of data being generated daily. Therefore, the development of automated fake news detection systems using Natural Language Processing (NLP) has received significant attention [36]. These systems aim to identify patterns in language, structure, and semantics to distinguish credible news from fake news. Recent advances in Deep Learning (DL) have also supported these operations by enabling models to automatically learn complex representations from raw text data without relying on manually engineered features.

Despite significant advancements in fake news detection, a major challenge remains that is identifying which DL models are most effective for this task in terms of accuracy, efficiency, and interpretability. While numerous studies have implemented individual models such as Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), and Bidirectional Encoder Representations from Transformers (BERT) few have conducted a comprehensive, side-by-side comparison of these architectures using a consistent dataset and evaluation strategy. Additionally, prior work often overlooks the interpretability of these models, which is crucial for real-world deployment. This thesis aims to fill this research gap by systematically comparing three widely used DL models LSTM, GRU, and BERT on the task of fake news detection. Using a balanced dataset sourced from Kaggle [21], this study evaluates each model not only based on predictive performance but also in terms of computational efficiency and explainability using Explainable Artificial Intelligence (XAI) techniques like Local Interpretable Model-Agnostic Explanations (LIME) and SHapley Additive exPlanations (SHAP).

LSTM and GRU are variants of Recurrent Neural Networks (RNNs) designed to capture temporal dependencies within sequential data and are commonly applied in NLP tasks [20] [11]. BERT, in contrast, is a transformer-based model that uses self-attention mechanisms and has achieved state-of-the-art performance in many

NLP benchmarks. However, these models vary significantly in terms of architecture complexity, computational requirements, and ability to capture context. Therefore, comparing them in the context of fake news detection is essential to determine which approach offers the best tradeoff between accuracy, efficiency and explainability.

Moreover, as DL models are often seen as "black-boxes," it is vital to explore their interpretability to build trust in real-world applications. This thesis incorporates XAI tools LIME and SHAP to interpret the models predictions. These tools help visualize which parts of the input text contributed most to a model's decision, thereby improving trust and usability in real-world applications.

LIME is a post-hoc interpretability technique that explains individual predictions of any black-box classifier by locally approximating it with an interpretable model [39]. LIME works by perturbing the input data and observing how the predictions change, thus creating a local surrogate model around a specific prediction that is inherently interpretable. On text data, LIME tends to pick words or phrases that contribute most to the model's decision, essentially providing us with an intuition about what textual features are most critical in classifying an article as fake or real. It is particularly valuable in fake news detection in order to enable researchers and end users to be aware of precise linguistic patterns or content features that lead to classification decisions.

SHAP is another widely impactful model interpretability framework, drawing on cooperative game theory. SHAP assigns an importance value to each feature in a prediction by estimating the average marginal contribution of every feature across all combinations of features. While LIME, SHAP also produces consistent and locally correct attributions with stronger theoretical guarantees. For detecting fake news, SHAP values are able to point out the contribution of single words or linguistic elements towards the prediction, which could be lost due to feature interactions in less complex methods of explanation [24]. With the advent of SHAP, this research aims to have a better insight into how different DL models assign importance to textual components while distinguishing true from false news content.

In summary, this thesis addresses a critical research gap by performing a comparative analysis of LSTM, GRU, and BERT models on the Kaggle fake news dataset. It evaluates not only their predictive accuracy but also their explainability and computational efficiency, with the goal of identifying the most effective and interpretable model for automated fake news detection.

## 1.1 Ethical, Societal and Sustainability aspects

### Ethical Concerns

This research utilizes a publicly available dataset from Kaggle [21], which has news headlines and article bodies. Although the dataset is open source, we also recognize the possibility that it includes sensitive or personally identifiable information. Therefore, we have taken a strong ethical stance on data handling. All information was used strictly for academic and research purposes, and steps are taken to anonymize any potentially sensitive content during preprocessing. Furthermore, as our models involve Machine Learning (ML), we have been mindful of potential algorithmic

bias. Evaluation metrics are carefully reviewed to ensure fairness, and interpretability tools like LIME and SHAP are employed to analyze model behavior and verify that predictions are based on valid textual patterns rather than biased or misleading features.

**Societal Aspects**

As digital media and online platforms have experienced an exponential growth, the threat of fake news has become a serious social concern. This research addresses this issue by developing models that can accurately detect fake news articles. The study explores three different DL models LSTM, GRU, and BERT with the aim of identifying the most reliable and interpretable solution. By incorporating explainability through LIME and SHAP, the project ensures that these models do not operate as black-boxes but instead provide human understandable justifications for their predictions. All of this goes towards increasing media literacy, ensuring public discourse, and strengthening trust in valid sources of information.

**Sustainability Aspects**

The computational costs of training large scale models such as BERT present a non negligible environmental impact. High performance Graphics Processing Units (GPUs) and Tensor Processing Unit (TPUs) consume significant amounts of energy, much of which may come from non renewable sources, contributing to carbon emissions. In response, our research aims to balance performance with sustainability. We compare resource intensive models like BERT with more lightweight alternatives such as LSTM and GRU, evaluating not only accuracy but also training time and energy consumption. By promoting efficient AI models, this work contributes to the broader goal of responsible and sustainable AI development.

## 1.2 Aim and objectives

### 1.2.1 Aim

The aim of this thesis is to conduct a comparative analysis of LSTM, GRU, and BERT models in fake news detection, addressing the critical need for effective automated misinformation detection systems.

### 1.2.2 Objectives

- To conduct a thorough review of existing literature on fake news detection and to design a robust data preprocessing pipeline tailored for NLP tasks, ensuring the dataset is clean, consistent, and suitable for DL models.
- To implement and optimize three neural network architectures LSTM, GRU, and BERT for the task of fake news classification using a labeled dataset sourced from Kaggle.
- To evaluate the performance of the LSTM, GRU, and BERT models by analyzing key classification metrics such as accuracy, precision, recall, and f1-score,

and to further interpret the models predictions using XAI techniques including LIME and SHAP in order to compare their effectiveness and interpretability in the context of fake news detection.

## 1.3 Problem Statement

The spread of fake news on social media poses a serious risk to public trust and democratic institutions. Traditional ML methods such as Logistic Regression and Support Vector Machine (SVM) are based on handcrafted features like Term Frequency-Inverse Document Frequency (TF-IDF) or Bag of Words (BoW), which often fail to capture contextual semantics and generalize poorly to complex or unseen data [4,33]. Recent studies show that DL models, particularly those using contextual embeddings like BERT, yield higher accuracy and understanding of linguistic nuances [5, 38]. However, the majority of these studies focus on individual models and do not take into account interpretability and efficiency. This thesis addresses these gaps by comparing LSTM, GRU, and BERT to identify the most accurate, interpretable model for fake news detection.

## 1.4 Research Questions

1. How do LSTM, GRU, and BERT models compare in terms of accuracy, precision, recall, and f1-score for fake news detection?

    - **Methodology Type:** Experimental and Quantitative.
    - **Method:** This question adopts a controlled experimental approach. Each model (LSTM, GRU, BERT) is trained using a standardized pipeline on a preprocessed fake news dataset. By applying word embeddings for LSTM and GRU, while BERT will use its built in contextual embeddings. Ensuring fair evaluation by using the same training/testing splits and hyperparameter tuning strategies for all models.
    - **Measurement:**The models performances will be evaluated and compared using quantitative metrics such as accuracy, precision, recall, F1-score, and confusion matrices. Performances will be evaluated. Also Scikit-learn will be used for metrics, and TensorFlow/PyTorch for model implementation.

2. What are the practical implementation considerations and resource requirements for deploying these models?

    - **Methodology Type:** Experimental and Quantitative.
    - **Method:** This question also follows an experimental methodology. The goal of this question is to analyze the computational efficiency of each model by monitoring resource utilization during training and inference. Also to measure memory consumption, training time per epoch, inference speed, and GPU utilization.

- **Measurement:** Quantitative data was gathered by manually measuring training and inference times while running all models on Kaggle with GPU P100. Instead of using advanced profiling tools, training and inference times were measured manually, as we could see the actual computing power and resources demanded by each model. This approach allowed us to identify the tradeoffs between model performance and computational efficiency in real-world, accessible environments.

## 1.5 Scope

This thesis focuses on evaluating and comparing three DL models LSTM, GRU, and BERT for detecting fake news. Using a labeled dataset from Kaggle, the study involves several key stages: data preprocessing, tokenization, model training with TensorFlow, and performance evaluation using metrics such as accuracy, precision, recall, f1-score, and confusion matrix. The research also includes model interpretability using LIME and SHAP to understand the reasoning behind predictions. The goal is to identify the most accurate and interpretable model for building reliable fake news detection systems.

## 1.6 Outline

- **Chapter 1: Introduction** Introduces the problem of fake news detection, ethical considerations, objectives of study, problem statement, research questions and scope of the thesis.

- **Chapter 2: Background** Presents key concepts in NLP and deep learning, including LSTM, GRU, BERT, and interpretability techniques.

- **Chapter 3: Related Work** Reviews prior research on fake news detection using traditional and deep learning methods, identifying gaps this thesis aims to address.

- **Chapter 4: Methodology** Describes the dataset, preprocessing steps, model architectures, Hyperparameters setup, and explanation techniques (LIME, SHAP).

- **Chapter 5: Results and Analysis** Reports and compares the performance of LSTM, GRU, and BERT models using evaluation metrics and error analysis.

- **Chapter 6: Discussion** Interprets model performance, discusses trade-offs between accuracy and efficiency, and analyzes interpretability outcomes.

- **Chapter 7: Conclusion and Future Work** Summarizes key findings, reflects on limitations, and proposes directions for extending the research.

## 1.7 System Flow (Simple Schema of Input and Output)

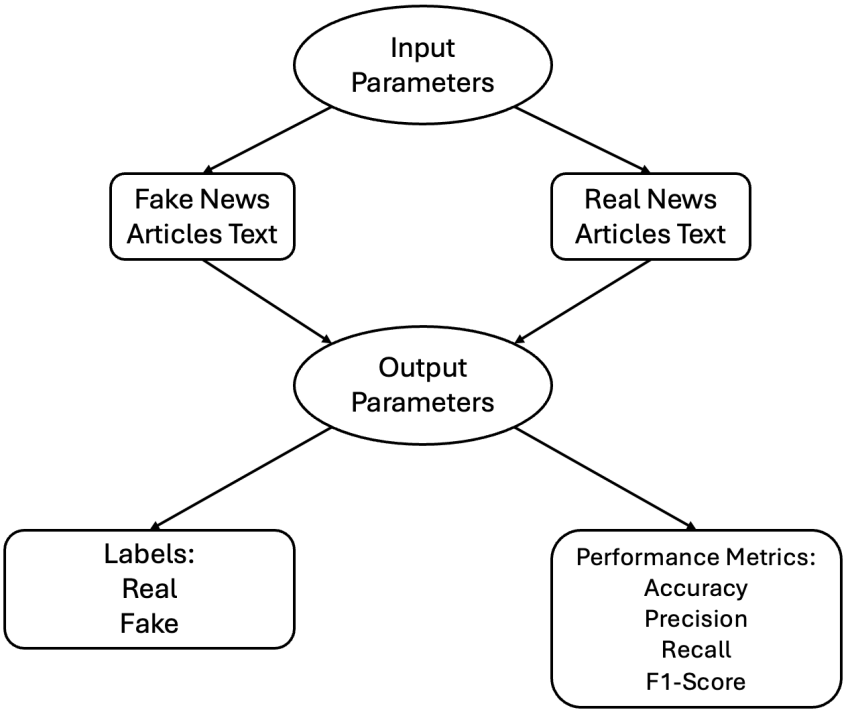


Figure 1.1: Input and Output Parameters

Flowchart Description in Layman's Terms:

**Input Parameters:**

- **Fake News Articles Text:** News articles known to be false, collected from the Kaggle fake news dataset.
- **Real News Articles Text:** News articles known to be genuine, collected from the same dataset.

These two categories of text data serve as the primary input to the deep learning models. The models learn patterns and linguistic features that differentiate real news from fake news.

**Output Parameters:**

- **Labels (Real / Fake):** After processing the input through the model, the system outputs a classification label for each news article either Real or Fake.
- **Performance Metrics:** To evaluate the effectiveness of the models, key performance metrics are calculated.

Figure 1.1 shows this input-output flow, summarizing the core structure of the fake news detection system developed in this thesis.

# Chapter 2

# Background

## 2.1 NLP

NLP is a subfield of Artificial Intelligence (AI) that focuses on enabling machines to understand, interpret, and generate human language. With the rapid spread of online information, NLP has become essential for automatically analyzing vast amounts of textual data, especially in detecting misleading or deceptive content such as fake news [27].

In fake news detection, NLP techniques help identify linguistic patterns, contextual cues, and semantic relationships that may signal the presence of false information. These tasks include tokenization, lemmatization, named entity recognition, and sentence classification. Preprocessing steps such as stopword removal and lowercasing are also critical in enhancing the quality of language models [26].

Modern fake news detection systems leverage DL models built specifically for handling sequential and contextual text data. Among these, LSTM and GRU networks have proven highly effective for modeling long-range dependencies in sequences, such as sentence structures and document streams [20] [11].

BERT recently emerged as a powerful pretrained transformer-based model that is capable of capturing complex language representations by considering both the left and right context of a word simultaneously [14]. BERT's contextualized embeddings have set new benchmarks in multiple NLP tasks, including text classification and stance detection making it especially suitable for the problem of fake news detection.

This thesis investigates and compares the application of LSTM, GRU, and BERT models for automated fake news detection, leveraging their advanced NLP capabilities to capture semantic context and improve classification accuracy.

### 2.1.1 NLP Techniques for Fake News Detection

In the context of fake news detection, NLP techniques play a critical role in transforming raw text data into structured formats that can be effectively analyzed by DL models. This process typically involves several stages like data cleaning, tokenization, embedding, and sequence modeling.

**Tokenization and Text Preprocessing:** Before applying DL models, raw text must be preprocessed to reduce noise and standardize the input. Common steps include converting all text to lowercase, removing punctuation and special characters, and filtering out stopwords. Tokenization, the process of splitting text into individual

words or tokens, is crucial for creating meaningful input sequences for models like LSTM and GRU [2].

**Word Embedding:** To represent text in a format understandable by neural networks, words are converted into dense vectors using embedding layers. These embeddings capture semantic relationships between words. In LSTM and GRU models, an embedding layer is typically trained alongside the model to learn task specific representations. In contrast, BERT uses contextual embeddings, allowing it to capture richer semantic and syntactic information from the input text [26].

**Sequence Modeling with LSTM and GRU:** LSTM and GRU networks are specialized forms of RNN designed to model temporal and sequential data. They are capable of capturing long term dependencies, which is particularly beneficial when detecting subtle linguistic patterns or changes in narrative tone across a piece of text [20] [11].

- **LSTM** incorporates memory cells and gating mechanisms to selectively retain or discard information, making it suitable for handling long news articles or documents with complex sentence structures.

- **GRU**, while similar in purpose, simplifies the architecture by combining the forget and input gates into a single update gate, leading to faster training with comparable performance.

**Contextual Understanding with BERT:** BERT has revolutionized NLP by introducing a strategy that learns bidirectional context from unlabeled text using masked language modeling. Unlike RNN based models that process input sequentially, BERT considers the entire sentence simultaneously, enabling it to learn the full context of a word based on its surroundings [14].

For fake news detection, the ability of BERT to understand the context and subtlety of language allows for more precise categorization, particularly in situations where subtle indicators distinguish real from fake news.

## 2.2 Deep Learning

DL is a subset of ML that focuses on using Artificial Neural Networks (ANNs) with large numbers of layers to learn complicated patterns in data. They are inspired by the structure and functioning of the human brain and have been at the core of the majority of cutting-edge advancements in areas like computer vision, NLP, and reinforcement learning [19]. DL algorithms automatically learn hierarchical feature representations from raw data, which is in contrast to traditional ML methods that require human designed feature engineering.

The primary architecture of DL is the neural network, which is composed of multiple layers of nodes or neurons. Such networks can learn low level and high level abstractions of data, and that is why DL has performed so well in executing complex tasks such as image recognition, language translation, and speech recognition [9]. The Feedforward Neural Network (FNN) is probably the most commonly used type

of deep neural networks, where data flows from the input layer to the output layer in one direction [8].

RNN's are another popular DL model, especially for sequential data like time series or text. They have loops that allow information to persist across time steps, making them suitable for tasks that require memory of previous inputs, such as language modeling and machine translation. A specialized type of RNN, known as LSTM, solves the vanishing gradient problem commonly faced by standard RNNs and is extensively employed in NLP applications [20].

Another groundbreaking architecture is the Transformer model BERT, which introduced the self-attention mechanism. Transformers are particularly highly appropriate for dealing with large scale sequential data, such as machine translation and text classification. Unlike RNNs like LSTM and GRU, where information is processed sequentially, Transformers process entire sequences in parallel, thus making them highly parallelizable and efficient [11].

The success of DL has been fueled by advances in computational power, particularly through GPUs, and the availability of large datasets. This enables DL models to be trained on large quantities of data, and this, in turn, has enhanced performance in numerous applications [22]. In the realm of NLP, models like BERT and its variants have demonstrated superior performance in tasks such as question answering, text classification, and sentiment analysis [14].

## 2.3 Evaluation Metrics

**Accuracy:** Accuracy refers to the overall correctness of the model's predictions. It is calculated as the ratio of correctly predicted instances to the total number of instances in the dataset [44].

**Precision:** Precision focuses on the positive predictions made by the model. It is calculated as the ratio of true positive predictions to the total number of positive predictions made by the model. Precision is particularly useful when the cost of false positives is high [29].

**Recall:** Recall measures the proportion of correctly predicted positive instances out of all actual positive instances. It helps evaluate how well the model detects positive instances. Recall is especially important in tasks where missing a positive case can have significant consequences [13].

**F1-Score:** The f1-Score is the harmonic mean of precision and recall, providing a single metric that balances both concerns. It is often used when there is an imbalance between the classes, offering a balance between precision and recall. A higher f1-Score indicates a better overall model performance in handling both false positives and false negatives [1].

**Macro Average:** This metric calculates the arithmetic mean of precision, recall, or f1-score across all classes, treating each class equally regardless of its frequency. It's particularly useful when assessing model performance uniformly across classes, especially in imbalanced datasets [23].

**Weighted Average:** This approach computes the average of precision, recall, or f1-score by weighting each class's metric according to its support (i.e., the number of true instances). It provides a more balanced evaluation by accounting for class imbalance, giving more influence to classes with more instances [23].

**Confusion Matrix:** The evaluation framework employs confusion matrices to assess model performance, providing a systematic tabulation of True Positives (TP, correctly identified fake news), True Negatives (TN, correctly identified genuine news), False Positives (FP, genuine news incorrectly labeled as fake), and False Negatives (FN, fake news mistakenly classified as genuine) [3].

# Chapter 3

# Related Work

Several studies have explored the use of ML and DL models for fake news detection. These attempts reflect a growing consensus that automated systems are essential for combating misinformation. However, most prior work focuses either on implementing a single model or evaluating performance in isolation, often neglecting comparative and interpretability aspects. This section reviews key contributions and highlights the limitations that justify the need for our study.

Rai and Kumar [32] proposed a hybrid model based on a pretrained BERT model and LSTM layer for fake news detection. Their model was evaluated on the FakeNewsNet dataset, which contains two prominent subsets PolitiFact and GossipCop. The hybrid model BERT + LSTM achieved 88.75% and 84.10% on PolitiFact and GossipCop, respectively, outperforming traditional baselines such as TCNN-URG (71.20%), CSI (82.70%), HAN (74.20%), and even standalone BERT (86.25% and 83.80%). The results suggest the potential benefit of integrating contextual embeddings with sequential learning mechanisms. However, the work failed to compare to simpler architectures like GRU, and it also failed to address model interpretability. The computational cost and efficiency of the model were not considered at all. Our work, however, comparatively analyzes LSTM, GRU, and BERT on one and the same dataset, and also considers XAI tools like LIME and SHAP in order to provide more insightful observations regarding model behavior.

A study in [7] proposes a fake news detection model using Bi-directional LSTM (Bi-LSTM), highlighting its ability to capture context in both directions. The model was evaluated against CNN, vanilla RNN, and unidirectional LSTM using two public datasets (DS1 and DS2). On DS1, Bi-LSTM achieved a test accuracy of 91.08%, slightly lower than unidirectional LSTM (91.48%) but higher than CNN (90.77%). On DS2, Bi-LSTM attained a test accuracy of 98.75%, outperforming vanilla RNN (96.38%) and closely following unidirectional LSTM (98.63%) and CNN (98.33%). The results confirm Bi-LSTM's strong performance in fake news detection, showcasing its ability to effectively model sequential dependencies and extract nuanced linguistic patterns that contribute to better classification accuracy. However, the study lacked comparative evaluation with transformer-based models such as BERT, did not explore model interpretability, and provided limited discussion on computational cost. Our thesis addresses these gaps by comparing both RNN based models and transformer models.

Mahara and Gangele [25], tackles the issue of misleading information by proposing an RNN based fake news detection model utilizing LSTM and Bi-LSTM architectures. The authors employed the NLTK toolkit for text preprocessing and incorpo-

rated GloVe word embeddings to capture semantic relationships in the data. The model integrates dropout layers and dense layers to improve efficiency, and it is optimized using the adam optimizer and binary cross-entropy loss. The Bi-LSTM model achieved a high accuracy of 94%, outperforming other configurations. Notably, the study found that increasing dropout beyond a certain threshold (0.3) leads to decreased performance, highlighting the importance of fine-tuning hyperparameters in DL based fake news detection. However, the study was limited in scope, as it focused only on RNN variants without comparing them to more recent transformer-based models. Moreover, the evaluation was conducted on a relatively narrow dataset, leaving questions about generalizability and comparative efficiency.

The research in [34] addresses the detection of fake news in under resourced languages, focusing specifically on Urdu language. The proposed model utilizes a Bidirectional GRU (Bi-GRU) architecture to extract latent features from news articles. These features are further enhanced through the concatenation of average and max pooling layers, with final classification handled by a softmax layer. Evaluated as part of the UrduFake shared task at FIRE-2020, the model achieved an average f1-score of 80.78% and an overall accuracy of 81.75%, securing the fourth position in the competition. This study highlights the effectiveness of Bi-GRU in handling fake news detection tasks, especially in linguistically low resource environments.

The study in [15] investigates fake news detection using a purely DL based approach, focusing on content level classification. The authors compare various RNN models, including vanilla RNN, LSTM, and GRU, on the LAIR dataset. Among the models, GRU achieved the best performance with a score of 0.217, slightly outperforming LSTM (0.2166) and vanilla RNN (0.215). Based on these results, the authors suggested the potential for a hybrid GRU-CNN model to further improve accuracy. While the study demonstrates that GRU can effectively model temporal dependencies in text, it remains limited to RNN based architectures and does not benchmark against more recent models like BERT.

Ramzan, Ali, Khan in [33] compares traditional ML models (Logistic Regression and Random Forest) and DL (LSTM) with the transformer-based model BERT for fake news detection. Logistic Regression, Random Forest, and LSTM achieved high accuracies of 98.7%, 98.8%, and 95% respectively on the training dataset. However, these models struggled with generalization when evaluated on unseen data due to limited contextual understanding. In contrast, BERT, though showing a slightly lower accuracy of 84%, proved more effective in capturing contextual nuances, making it a promising model for detecting fake news in real-world, diverse scenarios. While the study acknowledged BERT's potential, it lacked a detailed evaluation of model efficiency, scalability, and did not include GRU models.

A study in [5] addresses the increasing challenge of fake news spread on social media platforms, highlighting the inefficiency of traditional models in detecting such content. To tackle this, the researchers explored the capabilities of BERT, a state-of-the-art pretrained language model based on transfer learning. Their analysis compared baseline ML and DL models with BERT based architectures and concluded that BERT significantly outperforms earlier approaches in terms of accuracy, precision, recall, and reduction of false positives and negatives. The model was fine tuned by unfreezing its final layer and retraining it on a large dataset of 23,502 fake

and 21,417 real news articles collected from Kaggle. This customization allowed the model to better capture the linguistic and contextual nuances associated with fake news. By leveraging the power of transfer learning, the approach not only reduced training time but also improved generalization. The final model achieved an impressive 99.96% accuracy and 99.96% f1-score, showing its strong potential for real-world fake news detection. The study supports the growing preference for using pretrained transformer models in fake news research due to their contextual understanding and robust performance. While the study strongly validates BERT's contextual modeling capabilities and generalization power, it did not compare performance against RNN based models like LSTM or GRU. Furthermore, aspects like resource efficiency and model selection tradeoffs were not explored.

In this study [38], Subhash and Gupta focuses on the pressing issue of fake news dissemination, especially due to the rising influence of social media on public discourse in critical domains such as politics and the economy. To tackle this challenge, the researchers explored and compared various DL models paired with popular word embedding techniques. Specifically, they utilized GloVe and Word2Vec embeddings to convert textual data into numerical vectors and combined these with seven deep learning architectures: RNN, LSTM, Bi-LSTM, GRU, Bi-GRU, CNN-LSTM, and CNN-Bi-LSTM. In addition, FastText and BERT models were also evaluated to benchmark performance. The goal was to identify the most effective model in terms of classification accuracy and contextual understanding. Among all tested approaches, the BERT based model stood out significantly, achieving 99.20% accuracy on the test data and outperforming all other baseline models. The study was used accuracy, precision, recall, and f1-score as evaluation metrics to ensure robust assessment. This work confirms the strength of contextual embeddings and transformer architectures in capturing nuanced language patterns related to misinformation. The BERT model's state-of-the-art performance reaffirms its suitability for high accuracy fake news detection systems, making it a promising solution for real-world applications.

This paper [4] conducts a benchmark study evaluating various ML and DL models for fake news detection using text based features. It compares classical ML methods like Logistic Regression, SVM, Decision Tree, and ensemble techniques, alongside advanced DL models such as CNN, Bi-LSTM, Bi-GRU, and hybrid models like CNN-BiLSTM. Transformer based models, specifically BERT base and RoBERTa base, are also assessed. The study uses four real-world datasets LIAR, PolitiFact, GossipCop, and COVID-19 and contrasts context independent embeddings (like GloVe) with contextual ones (like BERT). Results show that BERT based models outperform others, demonstrating the importance of contextual understanding in effective fake news detection. However, the study does not explore the scalability or efficiency of these models in real-time applications, nor does it compare more lightweight alternatives such as LSTM.

In this study [12], the authors Choudhary and Arora address the growing concern of fake news, particularly its impact on political divisions and public trust. The research focuses on detecting fake news through text analytics, evaluating sequential memory based models like LSTM, Bi-LSTM, and Attention based Bi-LSTM, along with the transformer based BERT model. Using four diverse datasets from political, entertainment, satire, conspiracy, and global pandemic news, they find that

the BERT model significantly outperforms other models. Additionally, the Attention based Bi-LSTM achieved state-of-the-art results with high training accuracy. The study concludes that transformer-based models, especially BERT, are highly effective in detecting fake news, with attention mechanisms further enhancing model performance. Although the study highlights the effectiveness of transformer models, particularly BERT, it does not discuss the trade-offs between model complexity and real-world application requirements, such as computational efficiency and scalability.

| **Study** | **Model(s) Used** | **Dataset** | **Accuracy / F1-Score** | **Limitations** |
|---|---|---|---|---|
| Rai and Kumar (2022) [32] | BERT + LSTM, Baselines: TCNN-URG, CSI, HAN, BERT | FakeNewsNet (PolitiFact, GossipCop) | 88.75% (PolitiFact), 84.10% (GossipCop) | No GRU comparison; No interpretability; Ignored efficiency |
| Bahad et al. (2019) [7] | Bi-LSTM, CNN, RNN, LSTM | DS1, DS2 (public) | 91.08% (DS1), 98.75% (DS2) | No transformer comparison; No interpretability; Limited cost analysis |
| Mahara and Gangele (2022) [25] | LSTM, Bi-LSTM + GloVe | Custom (unspecified) | 94% | No transformer models; Narrow dataset; No scalability evaluation |
| Reddy et al. (2020) [34] | Bi-GRU + Avg/-Max Pooling + Softmax | UrduFake (FIRE-2020) | 81.75% Accuracy, 80.78% F1-Score | Focused on low-resource language (Urdu); No transformer baseline |
| Girgis and Amer, (2018) [15] | RNN, GRU, LSTM | LAIR | GRU (0.217), LSTM (0.2166) | RNN only; No transformer comparison; No interpretability |
| Ramzan and Ali. (2024) [33] | Logistic Regression, Random Forest, LSTM, BERT | Not specified | LR: 98.7%, RF: 98.8%, LSTM: 95%, BERT: 84% | No GRU; Generalization only partially discussed; No efficiency tradeoff |
| Aljawarneh and Swedat, (2024) [5] | BERT vs traditional ML/DL models | Kaggle (23,502 fake / 21,417 real) | 99.96% Accuracy and F1 | No RNN baseline; No efficiency or tradeoff analysis |
| Subhash and Gupta (2023) [38] | RNN, LSTM, Bi-LSTM, GRU, Bi-GRU, CNN-LSTM, BERT | Public dataset | BERT: 99.20% | No cost or real-time feasibility; Limited analysis on interpretability |
| Alghamdi (2022) [4] | ML (SVM, RF), DL (Bi-LSTM, Bi-GRU, CNN), BERT, RoBERTa | LIAR, PolitiFact, GossipCop, COVID-19 | BERT performed best | No lightweight model comparison; No real-time application focus |
| Choudhary and Arora (2024) [12] | LSTM, Bi-LSTM, Attention Bi-LSTM, BERT | Political, Satire, Conspiracy, COVID-19 | BERT outperformed others | No scalability or computational tradeoff discussed |
| **This Thesis** | LSTM, GRU, BERT with LIME and SHAP | Single Kaggle Dataset | LSTM: 98%, GRU: 93%, BERT: 99% | Direct comparison, interpretability via XAI, and resource tradeoffs evaluated |

Table 3.1: Summary of Related Work on Fake News Detection

However, none of the reviewed studies conducted a direct, side-by-side performance comparison of LSTM, GRU, and BERT models on a common single fake news dataset while also addressing interpretability and computational cost. Many works focus on a single model or compare models independently, often neglecting XAI tools like LIME and SHAP. Moreover, while BERT has shown state-of-the-art results, few studies examine tradeoffs between accuracy and resource demands. We selected LSTM, GRU, and BERT for this comparison as they represent diverse and widely adopted deep learning architectures in NLP ranging from traditional recurrent models to modern transformer-based systems. This thesis addresses this gap by comparing LSTM, GRU, and BERT on the same dataset, incorporating model interpretability and efficiency in order to present real world applications for fake news detection systems.

# Chapter 4

# Method

This thesis adopts a DL experimentation approach to explore and evaluate the effectiveness of various neural network architectures in detecting fake news. The dataset used in this study was collected from Kaggle, consists of labeled real and fake news articles [21], providing a comprehensive foundation for model training and evaluation.

The methodology begins with data preprocessing and intensive collection, such as text tokenization, and padding sequences to ensure uniform input lengths. Following preprocessing, the dataset is divided into three subsets after preprocessing: 60% for training, 20% for validation, and 20% for testing. This split ensures the models are trained effectively and tests them on unseen data.

Three DL models are developed and compared which are LSTM, GRU, and a Transformer based model named BERT. The LSTM and GRU models are built with embedding layers, recurrent layers, dropout for regularization, and a dense layer as the final layer with sigmoid activation for binary classification. The BERT model is implemented using a custom-trained tokenizer and a transformer-based sequence classification model. Text data is tokenized and converted into TensorFlow datasets with padding and truncation. The model is trained using early stopping and validated on separate validation data.

All models are evaluated using performance metrics such as accuracy, precision, recall, f1-score, and confusion matrix analysis. Moreover, XAI techniques like LIME and SHAP are applied to these models to understand the model's decision making process.

This methodological framework delivers a rigorous comparison of standard RNN based models with transformer-based models, not only their predictive precision but also models' interpretability in identifying fake news.

## 4.1 Working Environment

The code for this research was developed and executed in a Python programming environment version 3.13, utilizing a range of libraries and frameworks designed for DL, NLP, and explainability. Python's simplicity and powerful ecosystem make it an ideal choice for implementing complex ML models and data analysis tasks.

**Pandas:** Pandas is a versatile library for data manipulation and analysis, offering efficient data structures like DataFrames and a wide range of functions for data cleaning, transformation, and aggregation. It was employed to preprocess and manage the dataset efficiently [28].

**NumPy:** NumPy provides essential support for numerical computations, offering high performance arrays and mathematical functions. It was utilized for array operations and efficient handling of numerical data throughout the model development process [18].

**Matplotlib and Seaborn:** Matplotlib and Seaborn were used to create visualizations such as accuracy and loss curves, confusion matrices, and performance plots. These libraries helped in effectively communicating trends, model performance, and data distributions [10].

**TensorFlow:** TensorFlow is an open source DL framework that was used to build and train the LSTM, GRU, and BERT based models. TensorFlow provides robust tools for model development, optimization, and evaluation, making it well suited for handling large scale ML tasks [30].

**Keras:** Keras, a high level API running on top of TensorFlow, was employed for rapid prototyping and easy model building of the LSTM and GRU networks, providing simplified functionalities for defining, training, and evaluating DL models [16].

**Hugging Face Transformers:** The Hugging Face Transformers library was used to build a BERT model from scratch using TFBertForSequenceClassification and to create a custom tokenizer with BertTokenizerFast. It provides powerful tools for implementing state of the art transformer models, enabling efficient text classification for the fake news detection task [42].

**Scikit-learn (sklearn):** Scikit-learn was used for data preprocessing tasks such as data splitting (train-validation-test), feature extraction, and calculating evaluation metrics like accuracy, precision, recall, f1-score. It also supported the generation of the confusion matrix for model evaluation [31].

**LIME:** LIME was applied to the all models to provide local explanations of individual predictions, offering insights into the important words influencing the classification results [39].

**SHAP:** SHAP was utilized to provide a global understanding of the models by quantifying the contribution of each feature (word) to the predictions, enhancing the explainability of the DL model [24].

By leveraging these libraries and frameworks, the research was able to efficiently handle data preprocessing, model building, evaluation, and interpretability analysis, streamlining the overall development and experimentation process. As illustrated in Figure 4.1, this workflow systematically integrates these stages, ensuring a structured and comprehensive approach for comparing the performance of LSTM, GRU, and BERT models in fake news detection.

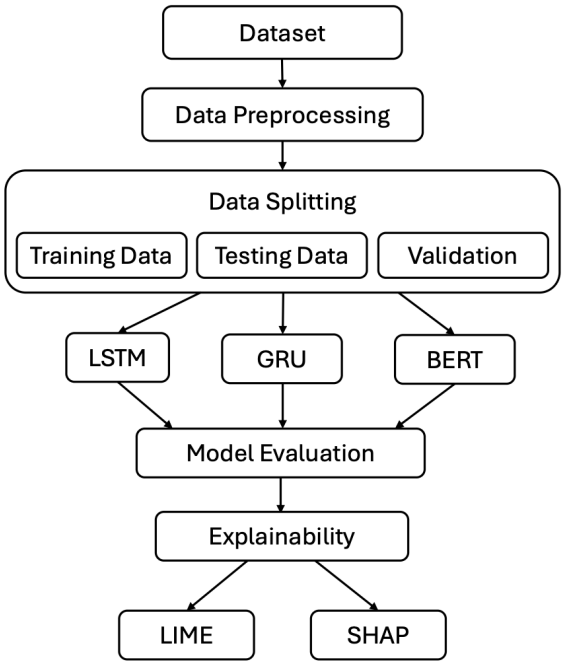


Figure 4.1: Working Process

## 4.2 Dataset

This research utilized a fake news dataset sourced from Kaggle [21]. The dataset is composed of two primary classes, Real and Fake news articles. Each record contains the article's title, text, subject, and date, where the true.csv contains all real news articles and fake.csv contains all fake news articles. For the purpose of this study, the **text** field (i.e., the full article body) was used as the primary input to all models during training, validation, and testing. The other fields were excluded to maintain consistency across samples and focus on the linguistic features within the article body.

Before training the models, the dataset was analyzed to understand its distribution and structure. Fake and real news articles were labeled and combined into a single dataset. The dataset was then split into training, validation and testing subsets to facilitate model training and evaluation.

| Label | Description | Percentage |
|---|---|---|
| True | Real news articles | 51.4% |
| Fake | Fake news articles | 48.6% |

Table 4.1: Distribution of True and Fake News Articles in the Dataset

This nearly balanced distribution supports effective training of ML models without introducing significant bias toward any particular class.

## 4.3 Pre-Processing

Initially, the raw data were prepared by assigning binary labels 0 for real, 1 for fake news articles respectively. Then data from the two CSV files were concatenated into a single DataFrame using Pandas and shuffled to ensure randomization. The merged dataset was then partitioned into training, validation, and test sets using stratified sampling to maintain class balance. Specifically, 60% of the data was allocated for training, while the remaining 40% was split evenly into validation (20%) and test (20%) sets. We used scikit-learn's train_test_split with a fixed random seed to ensure reproducibility.

Text pre-processing differed between the LSTM/GRU models and BERT model. For the LSTM and GRU models, we used Keras's Tokenizer to convert each article text into a sequence of integer word indices. The tokenizer was fit on the training corpus vocabulary only, and the <**OOV**> token was included to handle out of vocabulary words. Each document was then converted into a sequence of integers and padded or truncated to a fixed maximum length (100 tokens) using TensorFlow's **pad_sequences**, ensuring equal length inputs.

For the BERT model, we used a custom tokenizer trained using Hugging Face's **BertWordPieceTokenizer** on the training texts. The tokenizer handled subword tokenization and generated input IDs and attention masks. These were padded or truncated to a fixed maximum length of 128 tokens. The tokenized data was then converted into TensorFlow datasets for efficient training and evaluation.

After tokenization, data were converted into TensorFlow **tf.data.Dataset** objects for efficient training. Batches of data (batch size 64 for LSTM/GRU and 32 for BERT, due to memory constraints) were prepared using the **Dataset.batch()** method. The training pipeline also included shuffling, batching, and prefetching to optimize performance. This setup allowed the models to process batches of (input, label) pairs efficiently during training.

Here you can review the codes which we are executed during the thesis. Here you can check code in Github Repository.

## 4.4 Traditional Baselines

Before exploring advanced DL models, it is important to acknowledge the role of traditional ML baselines in fake news classification. These models are computationally inexpensive, easier to interpret, and often achieve competitive results, especially on smaller datasets. Including a discussion of these models provides context for understanding the motivation to pursue more complex DL architectures.

The commonly used baseline models in fake news detection tasks are:

- **Logistic Regression with TF-IDF features:** Logistic Regression is a linear model widely applied for binary classification problems, including fake news detection. By utilizing TF-IDF features, the model assigns weights to words based on their frequency and informativeness across documents. In the context of fake news, Logistic Regression classifies news articles by identifying key terms

that distinguish real from fake news. Its simplicity, fast training time, and interpretability make it a frequent baseline in text classification studies [17].

- **SVM with TF-IDF features:** SVMs are effective classifiers for high dimensional and sparse data such as text. When combined with TF-IDF features and a linear kernel, SVM can separate fake and real news articles by identifying a hyperplane that maximizes the margin between the two classes. SVMs are robust to overfitting, particularly in text data, and have been shown to perform well in fake news detection tasks where clear boundaries between classes exist [35].
- **Multinomial Naive Bayes:** Naive Bayes classifiers, particularly the multinomial variant, are commonly used for text classification problems, including spam detection and fake news identification. The model assumes that the presence of each word in a document is conditionally independent of other words, given the class label. Despite this simplifying assumption, Multinomial Naive Bayes is often surprisingly effective for fake news detection, especially when working with word frequency features like BOWs. It is also computationally efficient, making it a popular choice in scenarios with limited computational resources [43].

Although these models provide a useful benchmark and have demonstrated reasonable performance in previous studies on fake news detection, this thesis intentionally focuses on advanced DL models. The rationale for this decision is as follows:

1. **Contextual Understanding:** Traditional models largely work on the frequency of words as features and have no mechanism for understanding the contextual or sequential relation between words. DL models, particularly RNNs and transformers, can learn complex semantic and syntactic patterns from raw text data, which is crucial for detecting implicit misinformation.
2. **Prior Work Saturation:** Previous studies have extensively benchmarked traditional models on standard fake news datasets, establishing their performance levels. This thesis seeks to move past such traditional measures and compare relatively the merits of modern day DL frameworks.
3. **Research Objective Alignment:** The main goal of this thesis is to compare the performance of LSTM, GRU, and BERT models. Focusing on these advanced architectures allows for a more in-depth analysis of their capabilities in handling fake news classification tasks.

By narrowing the scope to DL methods, this study aims to assess whether recent advancements in DL, especially transformer-based model like BERT can offer substantial improvements over established traditional methods.

## 4.5 Deep Learning Models

This section presents the DL models implemented for the binary classification of news articles as fake or real. Such models are developed to utilize sequential patterns

and context relationships inherent in textual data, hence being optimally suited for detecting fake news.

### 4.5.1 LSTM Model

The first model implemented is a unidirectional LSTM model designed for binary classification of news articles. The architecture begins with a word embedding layer, which maps each input word index to a dense vector representation. This embedding layer, trained from scratch, produces vectors of dimension 128. To improve generalization, a SpatialDropout1D layer with a dropout rate of 0.2 is applied to the embeddings, which randomly drops entire word embeddings during training to prevent coadaptation of features.

The core of the model is an LSTM layer, which contains 100 hidden units and is responsible for capturing temporal dependencies in the text sequences. To mitigate overfitting, the LSTM layer incorporates both input dropout and recurrent dropout rates of 0.2. The LSTM operates through a set of gating mechanisms input gate, forget gate, and output gate that control the flow of information and regulate updates to the hidden and cell states. These operations can be mathematically expressed using standard LSTM cell equations in [41].

The mathematical operations of the LSTM cell can be expressed as:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{4.1}$$
$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{4.2}$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \tag{4.3}$$
$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \tag{4.4}$$
$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{4.5}$$
$$h_t = o_t \odot \tanh(C_t) \tag{4.6}$$

where $f_t$ is the forget gate that controls how much of the previous cell state to retain, $i_t$ is the input gate that regulates new information flow, $\tilde{C}_t$ is the candidate cell state, $C_t$ is the updated cell state, $o_t$ is the output gate that determines the next hidden state, and $h_t$ is the final output at time step $t$.

The final hidden state output by the LSTM is passed to a dense output layer with a sigmoid activation, producing the probability that an article is fake. The model is optimized using the adam optimizer and binary cross-entropy loss function, which is suitable for binary classification tasks.

**Architecture pipeline:**

The architecture pipeline outlined in Figure 4.2 illustrates the sequential flow of operations which were used in the LSTM model for fake news detection. It starts with raw text input, which is converted into word embeddings, followed by dropout for regularization, processed through an LSTM layer to capture temporal dependencies, and finally classified through a dense output layer. Each of these steps is described in detail in this section 4.5.1.

This design leverages LSTM's ability to capture long term dependencies in sequences, making it appropriate for text classification tasks such as fake news detection.

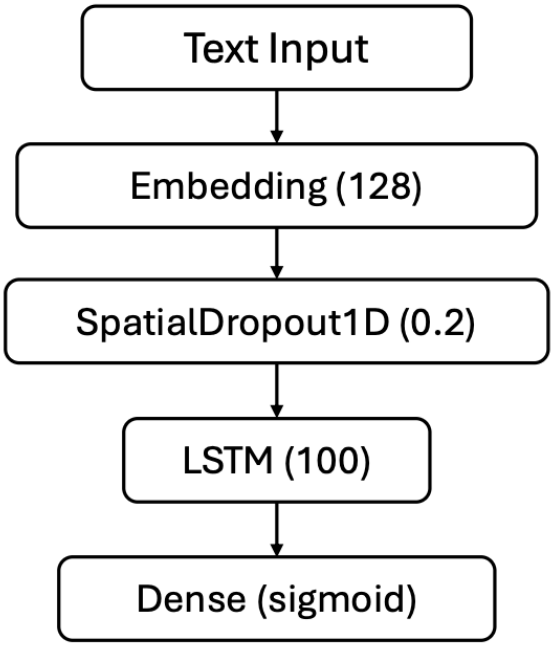


Figure 4.2: Architectural Pipeline of LSTM

### 4.5.2 GRU Model

The second model implemented is a unidirectional GRU network designed for binary classification of news articles. The architecture begins with a word embedding layer, which maps each input word index to a dense vector representation. This embedding layer, trained from scratch, produces vectors of dimension 128. To improve generalization, a SpatialDropout1D layer with a dropout rate of 0.2 is applied to the embeddings, which randomly drops entire word embeddings during training to prevent coadaptation of features.

The core of the model is a GRU layer, which contains 100 hidden units and is responsible for capturing temporal dependencies in the text sequences. To mitigate overfitting, the GRU layer incorporates both input dropout and recurrent dropout rates of 0.2 same as done in LSTM model execution. The GRU operates through a simpler gating mechanism compared to LSTM, utilizing only two gates namely update gate and reset gate. The update gate determines how much of the previous memory should be kept, while the reset gate controls how much of the previous state should be forgotten. This simplified structure often results in faster training while maintaining comparable performance to LSTM. This description of the GRU mechanism is adapted from [40].

The mathematical operations of the GRU cell can be expressed as:

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t] + b_z) \tag{4.7}$$

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t] + b_r) \tag{4.8}$$

$$\tilde{h}_t = \tanh(W \cdot [r_t \odot h_{t-1}, x_t] + b) \tag{4.9}$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t \tag{4.10}$$

where $z_t$ is the update gate, $r_t$ is the reset gate, $\tilde{h}_t$ is the candidate activation, and $h_t$ is the final output at time step $t$.

The GRU outputs the final hidden state, which is passed to a Dense output layer with a sigmoid activation to predict the probability that an article is fake. The model is optimized using the adam optimizer with a binary cross-entropy loss function, suitable for binary classification tasks, similar to the LSTM model.

**Architecture pipeline:**

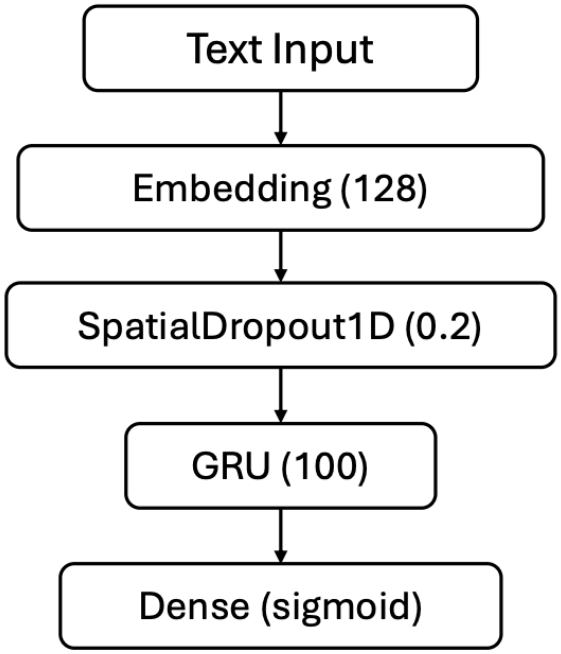


Figure 4.3: Architectural Pipeline of GRU

The architecture pipeline outlined in Figure 4.3 illustrates the sequential flow of operations which were used in the GRU model for fake news detection. It starts with word embeddings, followed by dropout for regularization, processed through a GRU layer to capture sequential patterns, and finally classified through a dense output layer. Each of these steps is described in detail in this section 4.5.2.

This design leverages GRU's efficiency in capturing sequential patterns while addressing the vanishing gradient problem common in standard RNNs, making it well suited for text classification tasks such as fake news detection.

### 4.5.3 BERT Model

The third model uses a custom BERT-base-uncased model for classification. We used this BERT-base-uncased model for binary fake news classification by adding a classification head on top and updating all model parameters on our dataset. A WordPiece tokenizer was trained from scratch on the training data with a vocabulary size of 10,000. The architecture consists of 12 transformer encoder layers with a hidden size of 768, and the classification head is applied to the [CLS] token representation to produce the final prediction for the fake news detection task.

The implementation begins with custom-trained WordPiece BERT model tokenizer created using the training texts, which converts text inputs into token IDs, attention masks, and token type IDs. The tokenizer handles special tokens such as [CLS] (at the beginning of each sequence) and [SEP] (separating text segments), with a maximum sequence length of 128 tokens. Longer sequences are truncated, and shorter ones are padded to maintain uniform input dimensions.

The core of the model is a BERT encoder configured with 12 transformer layers and 12 attention heads, which processes these tokenized inputs through its stacked transformer layers. Each transformer layer employs multi-head self-attention mechanisms that capture contextual relationships between all tokens in the sequence, allowing the model to incorporate bidirectional context for each word representation.

The model is trained end to end on the fake news dataset, meaning all parameters of the BERT model are updated alongside the classification head. This approach allows the model to adapt its powerful language understanding capabilities specifically to the fake news detection task.

**Architecture pipeline:**

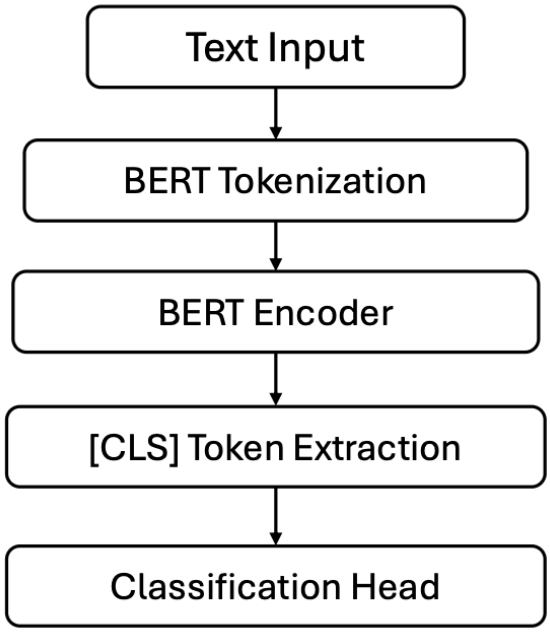


Figure 4.4: Architectural Pipeline of BERT Model

The architecture pipeline outlined in Figure 4.4 illustrates the sequence of operations were used in BERT model for fake news detection. It starts with tokenizing the input text, followed by processing through the BERT encoder to capture contextual relationships, after which the [CLS] token representation is extracted and passed to a classification head to produce the final prediction. Each of these steps is described in detail in this section 4.5.3.

This design leverages BERT's bidirectional context modeling and transfer learning from large-scale text, making it particularly effective for natural language understanding tasks such as fake news detection.

## 4.6 Hyperparameter Selection and Justification

All models in this study were trained to minimize classification loss using the Adam optimizer. For the LSTM and GRU models (binary classification tasks), the binary cross-entropy loss function was used. For the BERT model (also a binary classification task), sparse categorical cross-entropy was employed since the class labels were integer encoded (0 for real, 1 for fake).

We experimented with a set of default hyperparameters from past literatures and performed initial trial-and-error tuning on the validation set. Model performance was tracked continuously during training, and early stopping was implemented to avoid overfitting. Validation loss was monitored at each epoch, and training was halted if it did not decrease for 3 consecutive epochs (patience parameter). Upon triggering early stopping, the model weights at the epoch with the best validation performance were restored.

A summary of the chosen hyperparameters and their justification is shown in Table 4.2.

| **Hyperparameter** | **Model(s)** | **Value(s)** | **Justification** |
|---|---|---|---|
| Batch Size | LSTM, GRU | 64 | Based on common practice and memory availability, provided stable training. |
| | BERT | 32 | Smaller size used due to BERT's higher memory consumption. |
| Epochs | All | 5–15 | Early stopping with patience=3 was applied, range reflects max training allowed. |
| Learning Rate | LSTM, GRU | 1e-3 | Default Adam value, worked well on validation performance. |
| | BERT | 1e-5 | Common practice for BERT, stable convergence observed. |
| Dropout | LSTM, GRU | 0.2–0.3 | Used to mitigate overfitting, range selected based on validation results. |
| | BERT | Internal | BERT includes built-in dropout in its layers, no manual setting required. |
| Sequence Length | All | 100 tokens | Based on dataset statistics and NLP baseline practices. |
| Loss Function | LSTM, GRU | Binary Cross-Entropy (BCE) | Standard for binary classification using probability outputs. |
| | BERT | Sparse Categorical Cross-Entropy (Sparse CCE) | Suitable for integer-encoded labels (0, 1) and transformer output format. |

Table 4.2: Hyperparameter Choices and Their Justification

These hyperparameter settings were determined based on empirical testing, validation accuracy, and conventions established in previous research. Their careful selection helped ensure model stability, generalization, and reproducibility of results.

## 4.7 Explainability Techniques

To interpret the predictions of all three models (LSTM, GRU, BERT), we applied two complementary explainability methods to provide insights into their decision making processes.

LIME builds a simple, interpretable model that reflects the black-box model's prediction locally around an instance. That is, for a given news article, LIME perturbs the text (e.g., by removing or censoring words) and examines each model's predictions on perturbed samples. LIME then fits a weighted linear model to approximate

the model's predictions based on the presence/absence of words. The coefficients of this local surrogate indicate which words most influenced the classification. In this way, LIME identifies the top contributing words for an individual prediction (true or fake). This technique was applied to all three models, allowing us to compare how different architectures (LSTM, GRU, and BERT) focus on different linguistic patterns in the text.

SHAP was also used for all models. SHAP computes Shapley values from cooperative game theory to determine the contribution value of each word towards a specific prediction. SHAP values are the average word contribution to the prediction over all feature subsets. We used the SHAP framework to compute and visualize word significance in sample articles for all three model architectures. SHAP provides a property of global consistency and can be summed over instances to expose aggregated feature importance overall. This allowed us to learn from processing differences of textual information for transformer-based model (BERT) versus recurrent models (LSTM, GRU).

Both LIME and SHAP help to validate that the models are using reasonable evidence (e.g., certain indicative words) to make their decisions, and to expose potential biases or model errors. The comparative analysis of these explainability techniques across different model architectures provides valuable insights into how different neural networks approach the fake news detection task. Additionally, these techniques help identify which models rely on semantically meaningful patterns versus potentially spurious correlations, contributing to our understanding of model robustness and reliability.

# Chapter 5

# Results and Analysis

## 5.1 LSTM

The LSTM model demonstrated excellent performance in the fake news detection task across the training period of 6 epochs. As shown in the training plots in Figure 5.1, the model achieved a steady improvement in performance across training epochs. The accuracy plot reveals that the training accuracy started at approximately 0.951 and exhibited rapid improvement, reaching about 0.987 by the first epoch, then steadily climbing to 0.993 by the second epoch, and maintaining consistent growth to achieve over 0.997 by the sixth epoch. The validation accuracy began at 0.980, quickly improved to approximately 0.981 by the first epoch, peaked at around 0.983 during epochs 2-3, then showed slight fluctuations, declining to 0.977 by epoch 4 before recovering to 0.980 by epoch 6. This pattern indicates that the model achieved optimal validation performance around epochs 2-3 and began showing signs of overfitting as training accuracy continued to increase while validation accuracy declined and fluctuated.

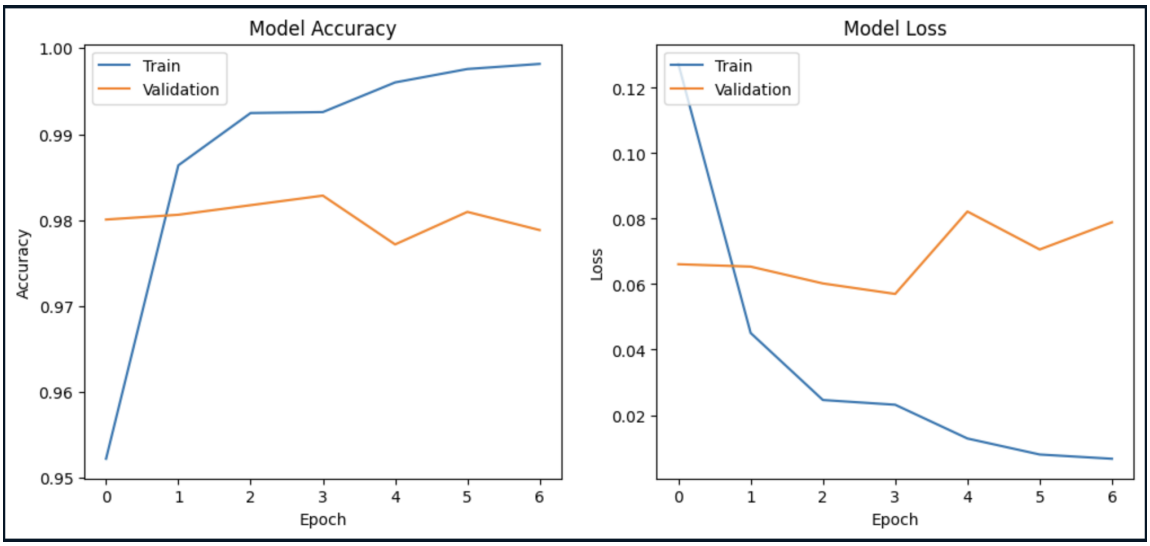


Figure 5.1: LSTM Model Accuracy vs Model Loss

The loss curves further support this observation, with training loss showing a continuous decrease from an initial value of approximately 0.125 to below 0.015 by the sixth epoch. The validation loss initially dropped from 0.066 to around 0.060 by epoch 1, remained relatively stable through epoch 2, then began fluctuating between 0.058

and 0.082 throughout epochs 3-6, ending at approximately 0.078. This divergence between training and validation loss trajectories after epoch 3 provides additional evidence of the model beginning to overfit to the training data.

These observations indicate that the LSTM model begins to overfit the training data after the third epoch. While early stopping was applied during this run, dropout layers with a rate of 0.2 were included between LSTM and dense layers to mitigate overfitting. Future training cycles could benefit from additional regularization techniques or early stopping to further optimize generalization.

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| Real | 0.98 | 0.99 | 0.98 | 4284 |
| Fake | 0.99 | 0.98 | 0.99 | 4696 |
| accuracy | | | 0.98 | 8980 |
| macro avg | 0.98 | 0.98 | 0.98 | 8980 |
| weighted avg | 0.98 | 0.98 | 0.98 | 8980 |

Table 5.1: Classification Report of LSTM

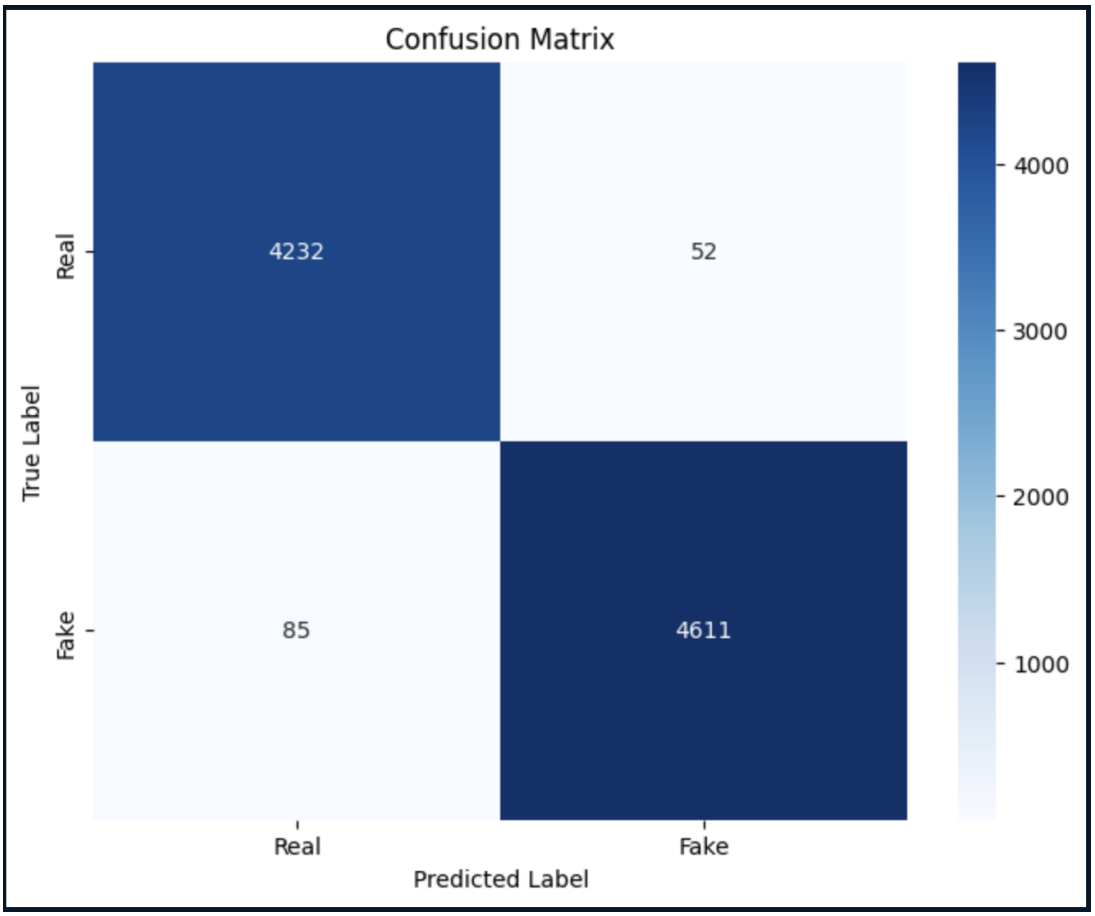


Figure 5.2: Confusion matrix of LSTM

The classification report presented in Table 5.1 shows that the LSTM model achieved an impressive overall accuracy of 0.98 across 8,980 test samples. The model demon-

strated perfectly balanced performance across both classes, with precision scores of 0.98 for "Real" news and 0.99 for "Fake" news, while recall scores were 0.99 for "Real" news and 0.98 for "Fake" news. This resulted in f1-scores of 0.98 for "Real" news and 0.99 for "Fake" news, indicating a marginal improvement in fake news detection capability. The support column indicates that the test dataset contained a slightly higher number of "Fake" news samples (4,696) compared to "Real" news samples (4,284). Despite this minor class imbalance, the model maintained equivalent performance metrics for both categories, demonstrating its robustness. Both macro and weighted averages of precision, recall, and f1-score align at 0.98, further confirming the model's balanced performance across classes.
The confusion matrix illustrated in Figure 5.2 provides a detailed breakdown of the model's predictions. Out of 4,284 "Real" news articles, the model correctly classified 4,232 (TP), while misclassifying 52 as "Fake" (FN). Similarly, from 4,696 "Fake" news articles, the model correctly identified 4,611 (TN), while erroneously labeling 85 as "Real" (FP). This results in a TP rate (sensitivity) of 98.8% and a TN rate (specificity) of 98.2%.

The symmetrically balanced error rates are particularly valuable in the context of fake news detection, where both types of misclassifications carry significant implications. The model's ability to maintain high precision and recall for both classes suggests that it has successfully learned discriminative features that reliably distinguish between real and fake news content without significantly favoring either class.

To better understand model limitations, a sample of misclassified headlines is shown in the Table 5.2 below. These examples illustrate common patterns where the LSTM model struggled:

| **Text Snippet** | **True Label** | **Predicted Label** | **Comment** |
|---|---|---|---|
| The Wall Street Journal reported on Friday tha... | Fake | Real | The reference to an established media source may have misled the model into classifying it as legitimate news |
| HANOI (Reuters) - U.S. President Donald Trump ... | Real | Fake | The formal wire service dateline format may have triggered false fake news detection despite being legitimate reporting |
| Ask your liberal friends to tell you more abou... | Fake | Real | The conversational and politically charged language may have appeared legitimate to the model despite being fabricated content |

Table 5.2: Examples of LSTM Model Misclassifications

These cases highlight that the LSTM model sometimes misclassifies:

- Source credibility bias: Headlines mentioning established sources (Wall Street Journal) may mislead the model toward predicting legitimacy
- Format-based misclassification: Formal wire service formatting (Reuters datelines) triggers false fake news detection
- Content vs style confusion: The model struggles to distinguish between legitimate statistical/political reporting and fabricated content with similar linguistic patterns

Future improvements could include fine-tuning the model with more domain specific data, and implementing attention mechanisms that focus on semantic content rather than stylistic cues to address these misclassification patterns.

## 5.2 GRU

The GRU model demonstrated solid but somewhat lower performance compared to the LSTM model in the fake news detection task. As illustrated in Figure 5.3, the model's training accuracy showed consistent improvement throughout the training process, starting at approximately 0.85 in epoch 0 and steadily increasing to 0.97 by epoch 7. However, the validation accuracy exhibited notably different behavior, beginning at about 0.91, gradually increasing to a peak of 0.927 around epoch 3, then remaining relatively stable with minor fluctuations, ending at about 0.927 by epoch 6. This divergence between training and validation accuracy curves strongly indicates overfitting, with the model increasingly optimizing for the training data at the expense of generalization capabilities.

The loss curves in Figure 5.3 provide further evidence of this pattern. The training loss started extremely high at around 900 and rapidly decreased to near zero levels by epoch 1, then maintaining extremely low values (close to 0) throughout the remaining epochs. In contrast, the validation loss remained relatively stable and low throughout training. The dramatic difference between initial training and validation loss suggests that the model quickly adapted to the training data but struggled to generalize effectively to unseen examples.

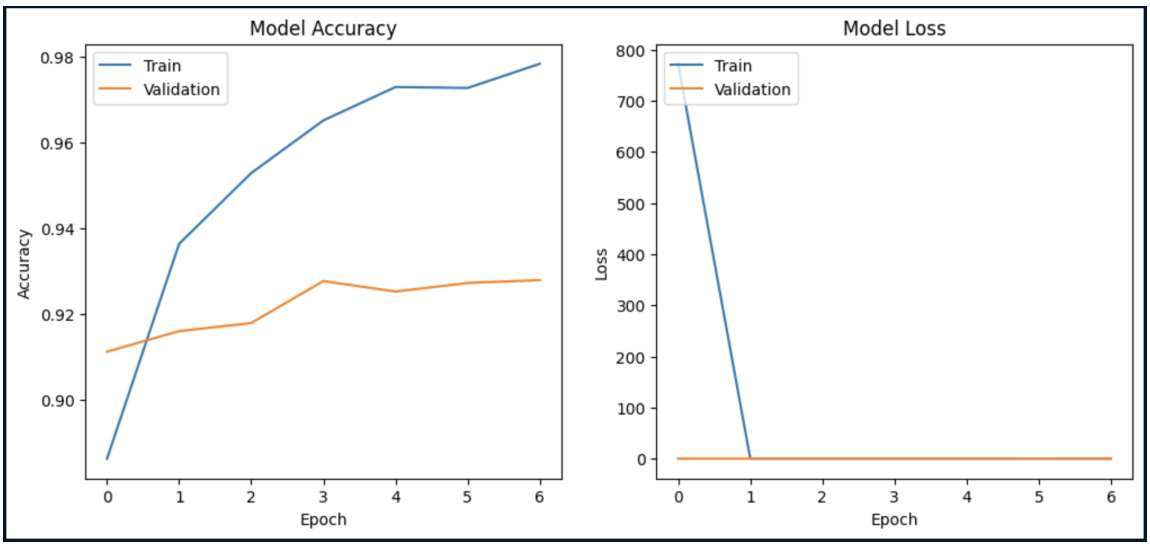


Figure 5.3: GRU Model Accuracy vs Model Loss

The GRU model shows signs of overfitting due to the large gap between training and validation loss, but the stable validation performance indicates the model isn't severely overfitting in the traditional sense where validation performance degrades. The model appears to have reached a plateau in its ability to generalize, rather than actively memorizing training data at the expense of validation performance.

The classification report presented in Table 5.3 shows that the GRU model achieved an overall accuracy of 0.93 across 8,980 test samples. Unlike the LSTM model, the GRU exhibited slight variations in performance between classes. For "Real" news detection, the model achieved a precision of 0.94 and a recall of 0.91, resulting in an f1-score of 0.93. For "Fake" news detection, it demonstrated a slightly higher precision of 0.92 but a marginally higher recall of 0.95, also yielding an f1-score of 0.93. The test dataset maintained the same distribution as in the LSTM evaluation, with 4,284 "Real" news samples and 4,696 "Fake" news samples. Both macro and weighted averages of precision, recall, and f1-score align at 0.93, reflecting the model generally balanced performance despite minor variations between classes.

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| Real | 0.94 | 0.91 | 0.93 | 4284 |
| Fake | 0.92 | 0.95 | 0.93 | 4696 |
| accuracy | | | 0.93 | 8980 |
| macro avg | 0.93 | 0.93 | 0.93 | 8980 |
| weighted avg | 0.93 | 0.93 | 0.93 | 8980 |

Table 5.3: Classification Report of GRU

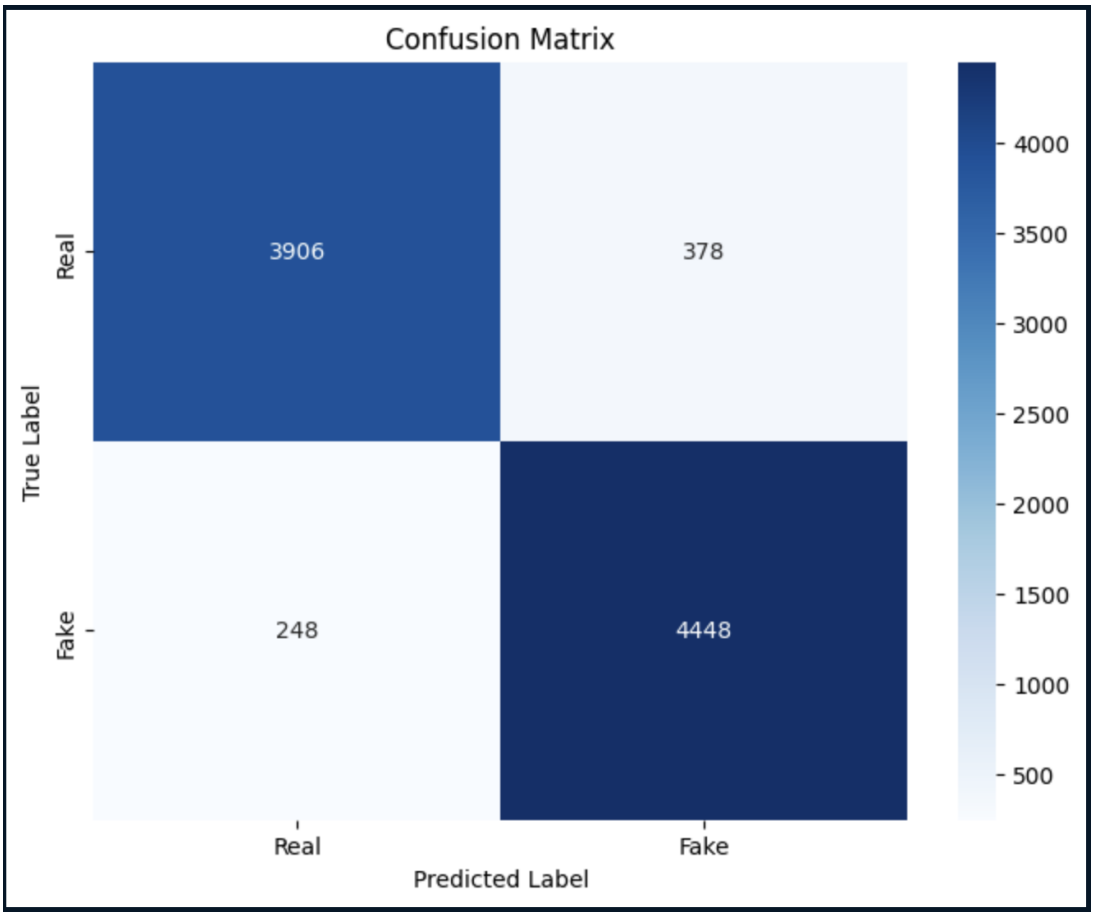


Figure 5.4: Confusion matrix of GRU

The confusion matrix illustrated in Figure 5.4 provides a detailed breakdown of the model's predictions. Out of 4,284 "Real" news articles, the GRU correctly classified 3,906 (TP), while misclassifying 378 as "Fake" (FN). From 4,696 "Fake" news articles, the model correctly identified 4,448 (TN), while erroneously labeling 248 as "Real" (FP). These results translate to a TP rate (sensitivity) of 91.2% and a TN rate (specificity) of 94.7%.

The error distribution reveals that the GRU model made substantially more classification errors than the LSTM model in both directions. The model struggled slightly more with FP (378) than FN (248), meaning it had a somewhat higher tendency to incorrectly classify fake news as real than to misclassify real news as fake. This imbalance in error types could have important implications in practical applications, potentially allowing more fake news to be distributed as real content.

To better understand model limitations, a sample of misclassified headlines is shown in the Table 5.4. These examples illustrate common patterns where the GRU model struggled:

| **Text Snippet** | **True Label** | **Predicted Label** | **Comment** |
|---|---|---|---|
| LOS ANGELES (Reuters) - Democratic lawmakers in California are working on new climate change legislation... | Real | Fake | The formal tone and news-like structure may have led to confusion with fake news |
| CNN cut away from a Senate Judiciary Committee hearing on the Trump administration's handling of Russia... | Fake | Real | The mention of a high-profile political context may have misled the model into classifying it as real |
| Pastor Kenneth Sharpton Glasgow claims he s Re... | Fake | Real | The inclusion of specific names and religious authority figures may have given this fabricated content an appearance of credibility to the model |

Table 5.4: Examples of GRU Model Misclassifications

These cases highlight that the GRU model sometimes misclassifies:

- News like structured content that appears credible but is actually fake.
- Headlines with political context that could confuse the model, especially with real events involving significant figures.
- Emotionally charged headlines with extreme language that misled the model into treating them as real news.

Future improvements could focus on adding more varied data from news sources with political or highly charged emotional language to better address these misclassification patterns.

## 5.3 BERT

The BERT model demonstrated exceptional performance in the fake news detection task, significantly outperforming both the LSTM and GRU architectures. As illustrated in Figure 5.5, the model exhibited remarkable training and validation metrics throughout the training process. The training accuracy began at an already impressive 0.9253 in epoch 0 and steadily improved to achieve perfect accuracy of 0.999 by epoch 9. The validation accuracy started even higher at approximately 0.9978 and maintained near perfect performance throughout training, with minimal fluctuations.

The loss curves in Figure 5.5 provide additional insights into the model's training dynamics. The training loss began at approximately 0.1284 and consistently decreased to approximately 0.0022 by epoch 9, demonstrating steady optimization. The validation loss started extremely low at around 0.0098 and maintain steadiness through the training process, reaching its minimum of about 0.0019 at epoch 4 before slightly increasing to 0.0032 by epoch 5. Despite this minor uptick in validation loss toward the end of training, the model maintained perfect validation accuracy, indicating that any increase in loss was negligible for classification decisions.

Unlike the LSTM and GRU models, BERT shows no significant overfitting. The validation accuracy remained consistently high throughout training with minimal divergence from training performance. This stability may be attributed to the use of pretrained weights, the application of dropout layers, and a relatively small number of training epochs, which helped avoid overfitting.

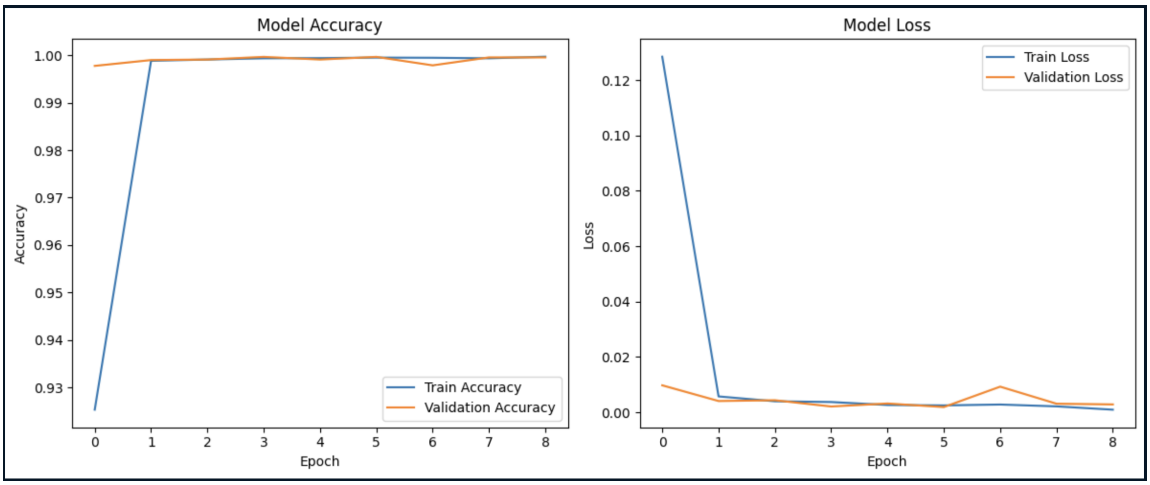


Figure 5.5: BERT Model Accuracy vs Model Loss

The classification report presented in Table 5.5 confirms the model's superior performance, with perfect scores across all metrics. The BERT model achieved an unprecedented 1.00 in precision, recall, and f1-score for both "Real" and "Fake" news categories across the 8,980 test samples. The test dataset contained a balanced distribution of 4,284 "Real" news samples and 4,696 "Fake" news samples. Both macro and weighted averages naturally achieve perfect 1.00 scores across all metrics, reflecting the model's flawless performance.

The confusion matrix illustrated in Figure 5.6 provides the most compelling evidence of BERT's exceptional capabilities. Out of 4,284 "Real" news articles, the model

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| Real | 1.00 | 1.00 | 1.00 | 4284 |
| Fake | 1.00 | 1.00 | 1.00 | 4696 |
| accuracy | | | 0.99 | 8980 |
| macro avg | 1.00 | 1.00 | 1.00 | 8980 |
| weighted avg | 1.00 | 1.00 | 1.00 | 8980 |

Table 5.5: Classification Report of BERT

correctly classified 4,278 (TP), with only 6 instances misclassified as "Fake" (FN). Similarly, from 4,696 "Fake" news articles, the model correctly identified 4,695 (TN), while just single instance as "Real" (FP). This results in a TP rate (sensitivity) of 99.86% and a TN rate (specificity) of 99.98%.

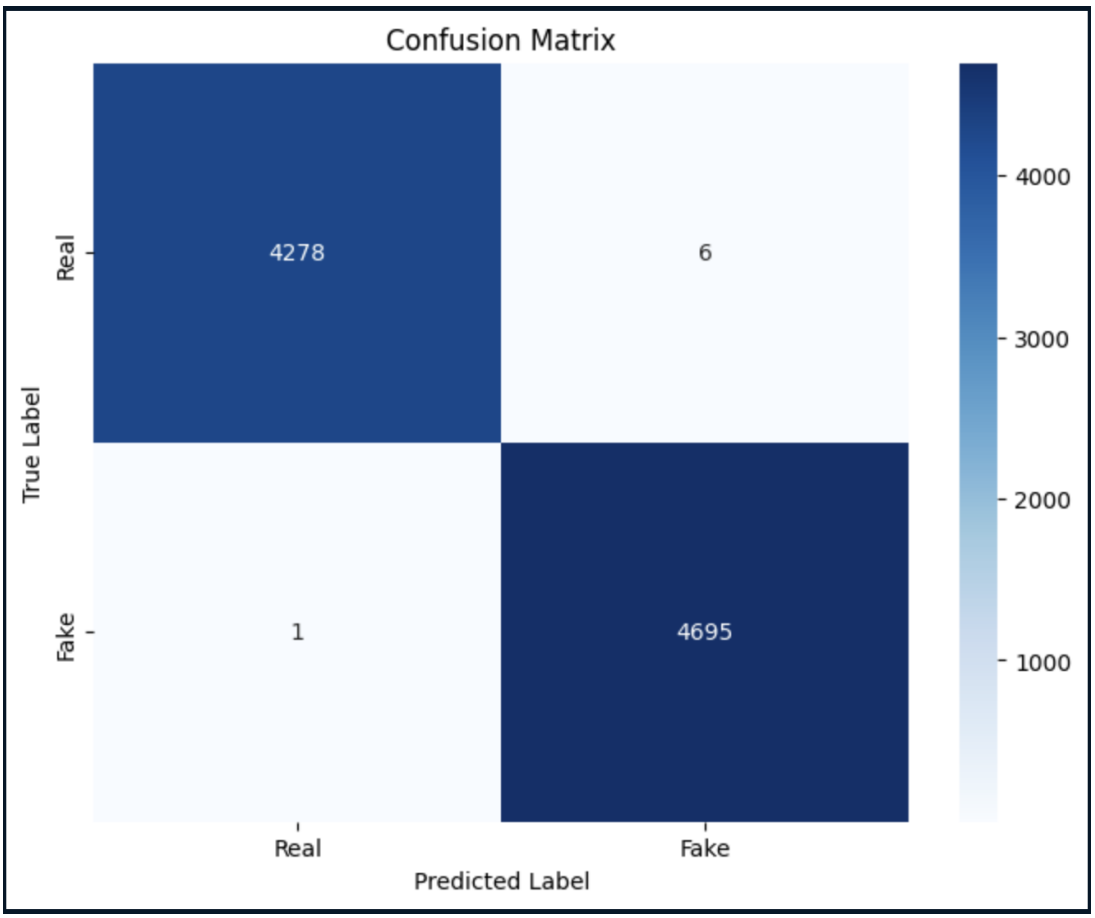


Figure 5.6: Confusion matrix of BERT

The near perfect performance demonstrated by BERT represents a substantial advancement over the previous models examined. While the LSTM model achieved a commendable 98% accuracy and the GRU model reached 93%, BERT's performance approaches theoretical perfection with an accuracy of 99%. This remarkable improvement can be attributed to BERT's sophisticated architecture, which leverages bidirectional context and transformer mechanisms to capture deeper semantic

understanding of text content compared to the sequential processing of RNNs like LSTM and GRU. With only 7 misclassifications out of 8,980 samples, BERT model demonstrates an unprecedented level of reliability for fake news detection, making it an exceptionally promising candidate for real-world applications where both high precision and recall are critical requirements.

To better understand model limitations, a sample of misclassified headlines is shown in the Table 5.6 below. These examples illustrate common patterns where the BERT model struggled:
These cases highlight that BERT model sometimes misclassifies:

| **Text Snippet** | **True Label** | **Predicted Label** | **Comment** |
|---|---|---|---|
| As is normal these days for Trump rallies, prot... | Real | Fake | The casual, opinion-based language and political bias may have led the model to misinterpret legitimate political coverage as fake news |
| A health care executive cheated Medicare out of... | Fake | Real | The factual reporting style and specific legal context may have given this fabricated story an appearance of credibility to the model |
| WELLINGTON (Reuters) - The final opinion poll b... | Real | Fake | The polling data and election-related content from an international source may have been misinterpreted as potentially manipulated information by the model |

Table 5.6: Examples of pretrained BERT Model Misclassifications

### 5.3.1 Analysis of BERT Misclassifications

Even though BERT performed well overall, there are some incorrectly classified articles as shown in the Table 5.6 they show recurring trends that draw attention to its shortcomings. BERT often struggles with fake news articles that are written in a formally structured and polished journalistic style, which it tends to associate with credibility often it cause problems. For example, the model may mistakenly predict headlines as authentic if they retain grammatical coherence and resemble the tone of actual news. Additionally, it seems that the inclusion of well known people or authoritative organizations like institutions or political leaders biases BERT to accept such content as factual even when it is not. This implies that the contextual embeddings in BERT might give named entities more weight. Moreover, ambiguity is added by references to social media sites like Twitter, which frequently lack obvious stance indicators and combine factual and subjective tones, making it challenging for BERT to assess authenticity. Despite its advanced language understanding capabilities, these patterns highlight BERT's susceptibility to contextually subtle or stylistically misleading content.

### 5.3.2 Comparative Misclassification Insight

By comparing these BERT error cases with those from the GRU and LSTM models, a pattern emerges:

- **BERT fails on contextually complex, nuanced texts.**
  For example, articles that appear highly formal or contain references to political figures or social media tend to confuse BERT. This includes cases where fake news mimics journalistic structure or leverages popular names, leading BERT to incorrectly classify them as real.

- **LSTM fails on formally structured and geopolitically dense news reports.**
  Examples include:

  – *"The Wall Street Journal reported on Friday..."* (Fake → Real)
  – *"HANOI (Reuters) - U.S. President Donald Trump..."* (Real → Fake)
  – *"BANGKOK (Reuters) - Thailand officially ended..."* (Real → Fake)

  LSTM appears to struggle with distinguishing between factual global news and fabricated content when the writing is formal, concise, and information-rich, potentially due to limitations in capturing long-range dependencies in such texts.

- **GRU fails on politically charged or opinion-heavy content.**
  Examples include:

  – *"CNN cut away from a Senate Judiciary Committee..."* (Fake → Real)
  – *"Pastor Kenneth Sharpton Glasgow claims he's Re..."* (Fake → Real)
  – *"PARIS (Reuters) - Corsican nationalists won al..."* (Real → Fake)

  GRU tends to misclassify content that blends opinion with fact or that involves emotionally or politically sensitive subjects. Its reliance on sequential context may be insufficient when subtle indicators of credibility are embedded in such narratives.

Future improvements could include training on data that specifically includes social media contexts and refining the model to better differentiate between news involving famous personalities.

## 5.4 Computational Metrics

To provide a clearer understanding of the computational demands of each model, we report estimated training and inference times along with parameter count and GPU memory usage. All training, development, testing, and inference operations were performed on cloud based platform Kaggle using accelerator GPU P100.

| Model | Training Time | Inference Time (per sample) | Epochs | GPU RAM Usage |
|---|---|---|---|---|
| BERT | ∼1:05 hours | ∼524 ms | 9 | ∼13GB |
| GRU | ∼8 minutes | ∼154 ms | 7 | ∼8GB |
| LSTM | ∼13 minutes | ∼255 ms | 7 | ∼8GB |

Table 5.7: Estimated Computational Metrics for Each Model

These results show that transformer-based model like BERT require significantly more training time and GPU memory compared to traditional RNN based models like LSTM and GRU. This is largely due to their higher parameter count and self attention mechanism. However, transformers offer improved accuracy and contextual understanding, often justifying their computational cost in practical applications.

In summary, among the three models evaluated LSTM, GRU, and BERT the BERT model consistently outperformed the others across all key performance metrics. With a test accuracy of 99%, it demonstrated almost flawless performance, showing greater proficiency at picking up contextual information and linguistic nuances. While LSTM and GRU also posted admirable performance with 98% and 93% accuracies respectively, they lagged behind in predictive power as well as robustness to complex language structures. However, BERT remains the most accurate and reliable model in this study, confirming its state-of-the-art status for the fake news detection task.

While BERT demonstrated the highest accuracy, it also incurred the greatest computational cost. Given that training large transformer models has been shown to emit significant $CO_2$ [37], this raises concerns for sustainable deployment. In contrast, lightweight models like DistilBERT which require fewer parameters and less GPU memory offer a more environmentally friendly alternative with only a marginal drop in accuracy. These findings emphasize the importance of balancing model performance with energy efficiency, especially when scaling up to global fake news detection systems.

# Chapter 6

# Discussion

This chapter discusses the outcomes of the study in light of the research objectives and questions. The results from Chapter 5 are interpreted to evaluate the effectiveness and practicality of LSTM, GRU, and BERT models in detecting fake news. Additionally, model interpretability is examined through LIME and SHAP visualizations.

**Research Question 1:** How do LSTM, GRU, and BERT models compare in terms of accuracy, precision, recall, and f1-score for fake news detection?

The experiments confirmed that BERT significantly outperforms LSTM and GRU across all standard classification metrics. Specifically:

- BERT model achieved the highest overall performance with 99% accuracy, precision, recall value, and f1-score, showcasing its strength in understanding context and language nuances.
- LSTM followed with 98% accuracy, performing well due to its ability to capture sequential dependencies using memory cells.
- GRU attained 93% accuracy, slightly lower than LSTM, possibly due to its simpler architecture.

These results validate the effectiveness of transformer-based models like BERT in fake news detection, especially when contextual understanding plays a critical role. In contrast, LSTM and GRU, while effective, are more reliant on surface level word patterns and less capable of capturing deeper semantic meaning.

**Research Question 2:** What are the practical implementation considerations and resource requirements for deploying these models?

While BERT showed superior performance, its computational demands were significantly higher:

- **Training Time:** BERT models requires more time per epoch compared to LSTM and GRU, mainly due to their deeper architecture and attention mechanisms.
- **Memory Consumption:** BERT's memory footprint was larger, especially during tokenization and forward passes, requiring a smaller batch size (32) compared to LSTM and GRU (64).
- **Inference Speed:** LSTM and GRU offered faster inference times, which may be beneficial for real-time systems with limited hardware resources.

- **Resource Utilization:** Monitoring tools showed that BERT model utilized GPU and RAM more intensively, confirming its higher resource requirements.

These findings suggest that while BERT model is ideal for offline or high performance environments, LSTM and GRU may be more practical choices for low resource or embedded systems where speed and memory are constrained.

## 6.1 Model Interpretability with LIME and SHAP

To improve the transparency and trustworthiness of the models, LIME and SHAP were applied to each model to visualize how specific features (words) influenced predictions.

### 6.1.1 LSTM

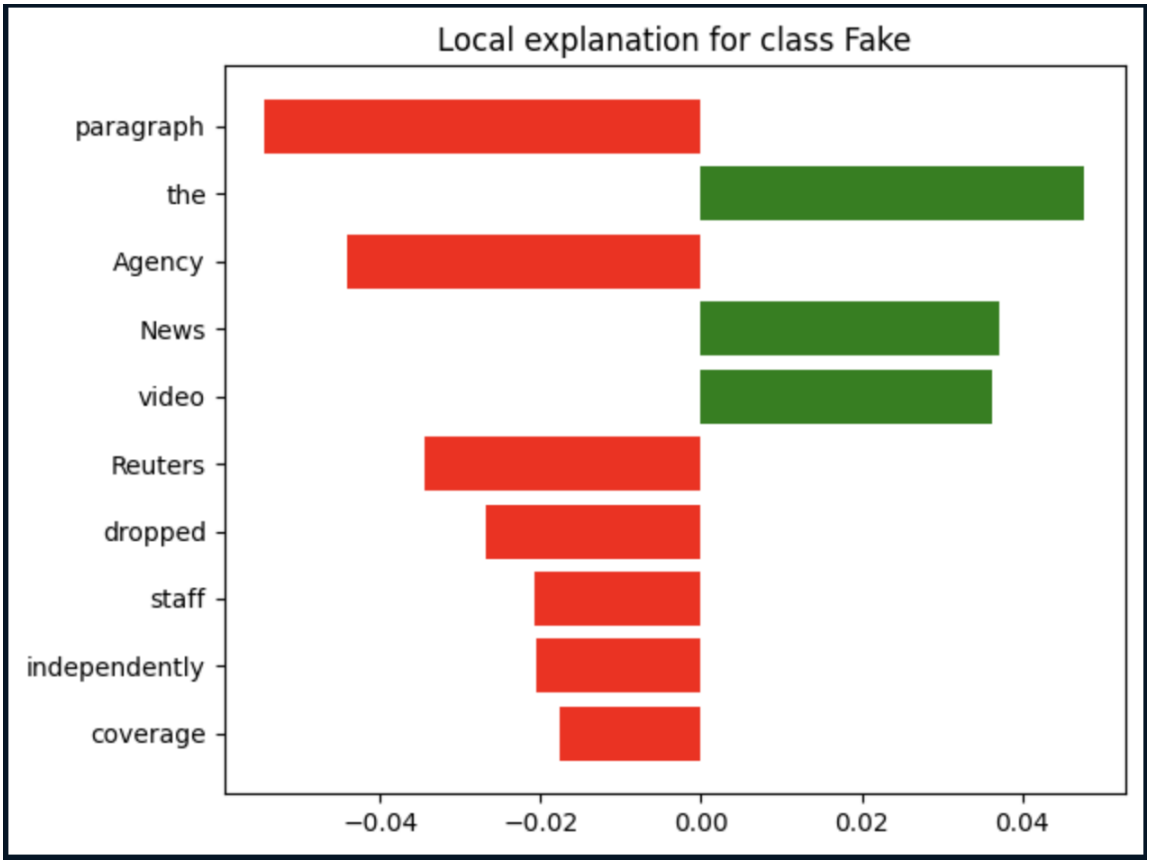


Figure 6.1: LIME interpretability for LSTM

Figure 6.1 presents the LIME explanation for a prediction classified as Fake by the LSTM model. Green bars indicate words that contributed positively toward the fake class, while red bars had a negative impact. Words like “News” and “video” supported the fake classification, whereas words like “paragraph” and “Agency” leaned toward the real class. Some tokens like “the” may indicate tokenization artifacts.

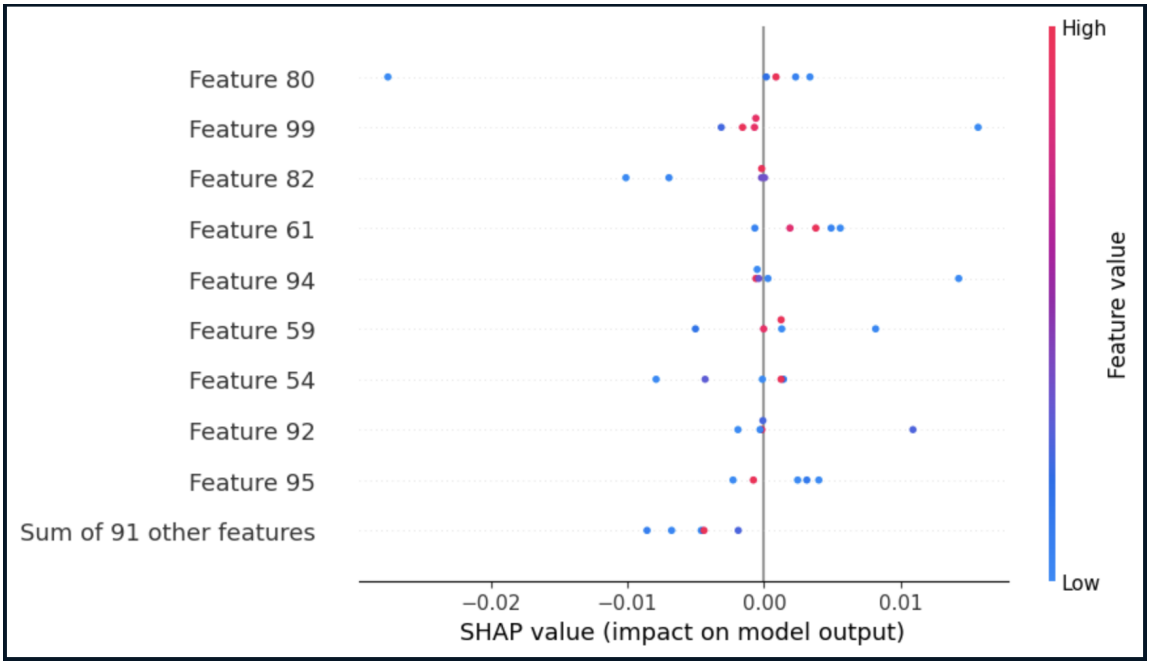


Figure 6.2: SHAP interpretability for LSTM

Figure 6.2 shows the SHAP summary plot, highlighting the overall contribution of features (words) to the model's output. Features with higher SHAP values (e.g., Feature 99) had a greater influence on predictions. The color gradient shows how the feature's value (high or low) impacts the model.

These XAI tools help validate that the LSTM model learns meaningful patterns, while also pointing out areas for preprocessing improvement.

### 6.1.2 GRU

Figure 6.3 presents the LIME explanation for a prediction classified as Fake by the GRU model. Green bars indicate words that contributed positively toward the fake class, while red bars had a negative impact. Words like "the," "neon" and "at" strongly supported the fake classification, with "the" having the most substantial positive impact. The word "Tuesday" had a significant negative impact, pushing toward the real class. Smaller contributions came from words like "at" (positive) and "levels" (negative). This shows how specific language patterns influence the GRU model's classification decisions.

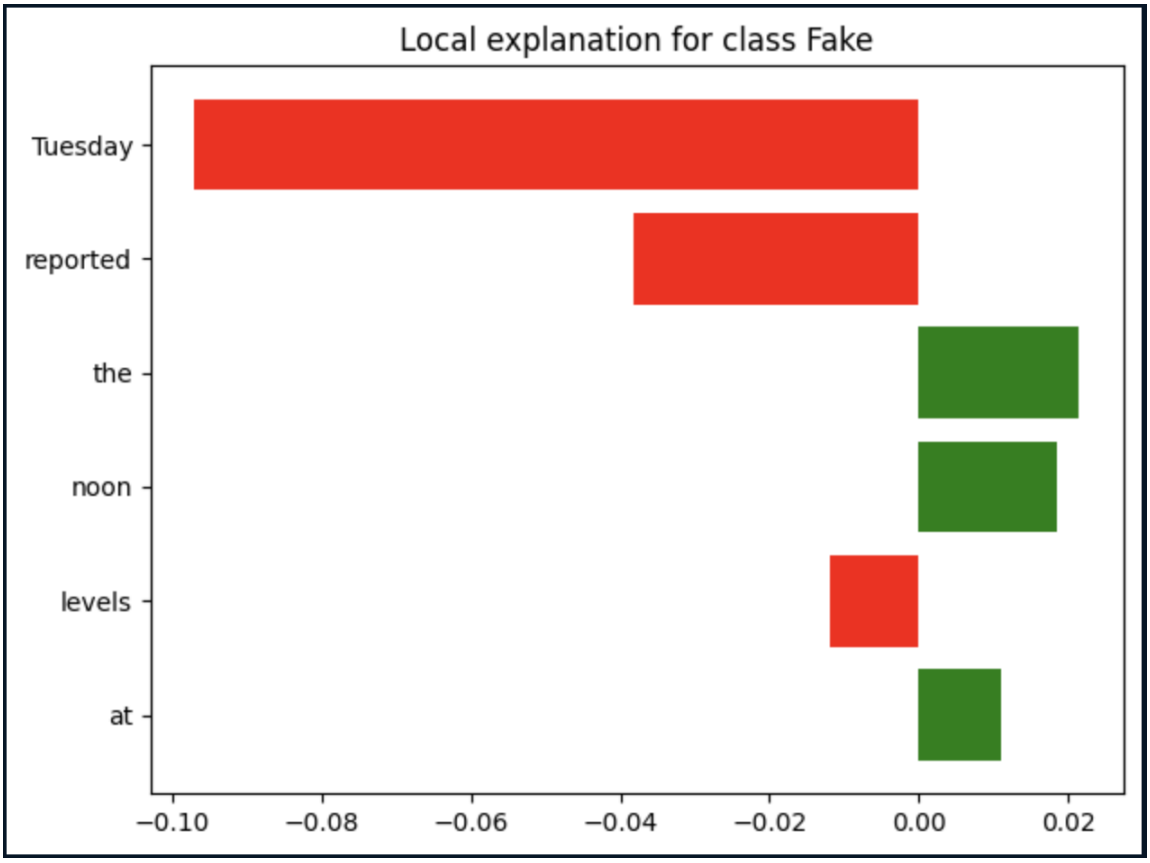


Figure 6.3: LIME interpretability for GRU

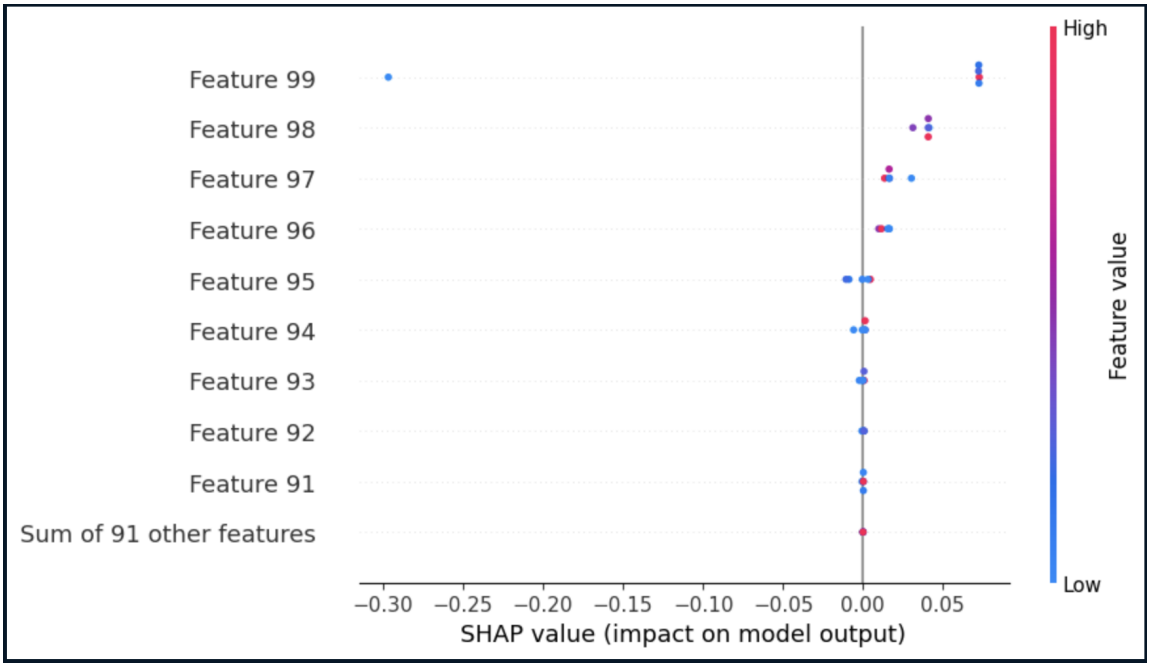


Figure 6.4: SHAP interpretability for GRU

Figure 6.4 shows the SHAP summary plot, highlighting the overall contribution of features (words) to the GRU model's output. Features with higher absolute SHAP

values, such as Feature 99 and Feature 98, exerted the greatest influence on predictions. Feature 99 has the most substantial negative impact on model output when present. The color gradient from blue to red indicates how each feature's value (low to high) affects the classification. Most features cluster near zero SHAP value, while a few key features (particularly 97-99) show significant influence. This demonstrates that the GRU model relies on specific word patterns rather than a broad distribution of features when making predictions.

These explainability tools confirm that the GRU model identifies meaningful linguistic patterns in the text for fake news detection, while also revealing which specific words and features carry the most weight in its decision making process.

### 6.1.3 BERT

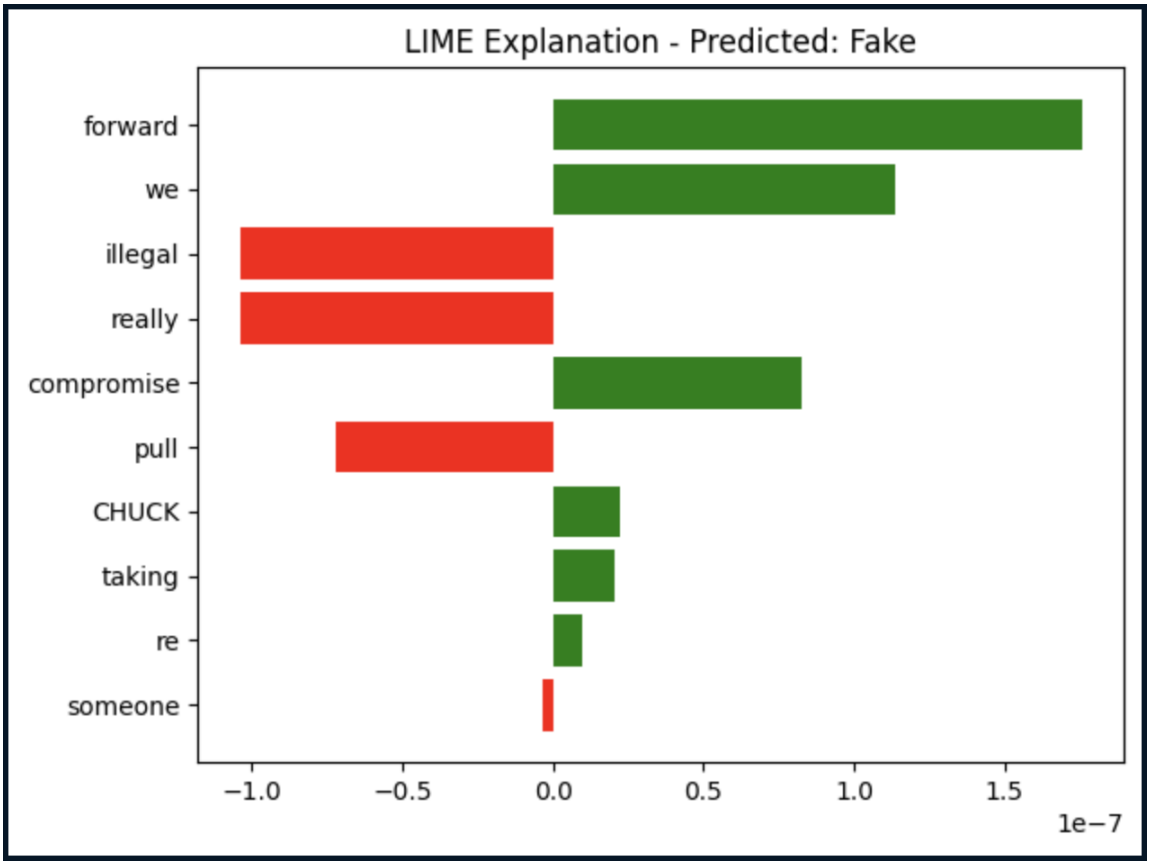


Figure 6.5: LIME interpretability of BERT

Figure 6.5 presents the LIME explanation for a prediction classified as Fake by BERT model. Green bars indicate words that contributed positively toward the fake classification, while red bars had a negative impact. Words like "forward" and "we" had the strongest positive influence on the fake classification, followed closely by "compromise". Conversely, words like "illegal", "really" and "push" pushed toward the real class prediction with negative contributions. This visualization demonstrates how BERT weighs specific contextual language cues when making its classification decision.

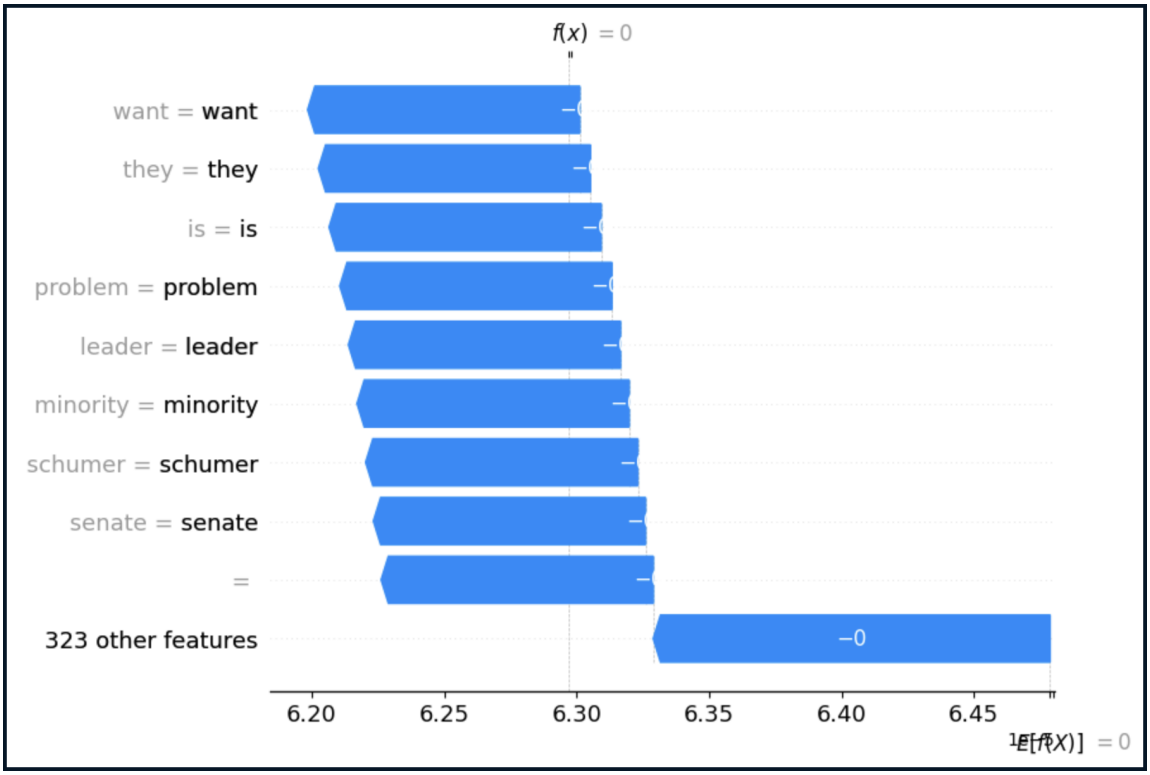


Figure 6.6: SHAP waterfall plots for real news classification by BERT model

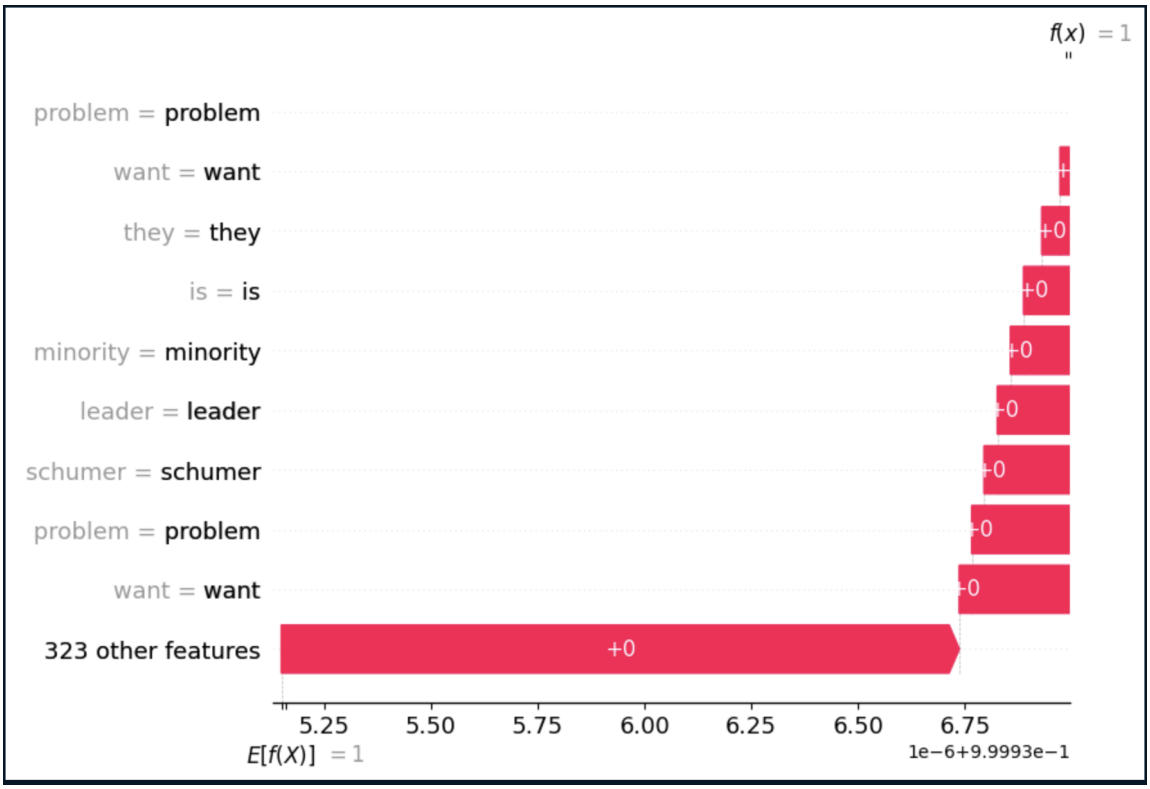


Figure 6.7: SHAP waterfall plots for fake news classification by BERT model.

Based on the SHAP waterfall plots for the Real News class (Figure 6.6), the vi-

sualization demonstrates how individual features contribute to classifying a specific instance as real news. The model starts with a base value around 6.20 and progresses toward the final prediction through cumulative feature contributions. Key words like "want," "they," "is," "problem," "leader," "minority," "schumer," and "senate" all contribute positively to the real news classification, with each feature incrementally pushing the prediction score higher. The waterfall format effectively illustrates the sequential decision-making process, showing how the model builds confidence in its real news prediction through the accumulation of supportive linguistic features, ultimately reaching a final prediction value near 6.45.

The SHAP waterfall plot for the Fake News class (Figure 6.7) reveals a contrasting pattern in feature contributions for fake news classification. Starting from a lower base value around 5.25, the model demonstrates how various features contribute to identifying fabricated content. Words such as "problem," "want," "they," "is," "minority," "leader," and "schumer" show strong positive contributions toward fake news classification, with the substantial "323 other features" block providing the most significant cumulative impact. The visualization shows how the model systematically accumulates evidence for fake news classification, with the final prediction reaching approximately 6.75. This waterfall analysis reveals the model's ability to distinguish fake news through specific linguistic patterns and feature combinations that differ markedly from those used for real news identification.

These XAI visualizations demonstrate that BERT model captures sophisticated linguistic patterns in fake news detection, leveraging its contextual understanding to identify specific words and phrases that characteristically appear in fake versus legitimate news sources.

### 6.1.4 Comparative Analysis of Model Explanations

While each model was interpreted using LIME and SHAP, a cross-model comparison reveals important differences in how LSTM, GRU, and BERT process and prioritize input features. Both LSTM and GRU models placed more emphasis on specific keywords usually nouns or named entities (e.g., "video," "neon," "Tuesday") and tended to attribute weight to either frequent words or those in prominent positions in the sentence. This suggests a dependence on superficial patterns. In contrast, BERT discriminated more subtle and context-dependent tokens like "compromise," "push," or "illegal," showing its capability to interpret word meaning in relation to surrounding text through self-attention mechanisms.

Another key distinction lies in how the models behave across different prediction types. For an LSTM and a GRU, false-positive explanations often contained contribution from non-informative tokens, such as articles ("the") or tokens like "paragraph," betraying the fact that the models had some weakness in really grasping deeper semantics. BERT, on the other hand, has a more structured accumulation of evidence, at least in SHAP waterfall plots, with the decision being built somewhat gradually through contextual word relationships. This can be observed particularly well by comparing how BERT deals with classifying real news to how it deals with classifying fake news, using entirely different combinations of feature contributions

in each case.

Overall, BERT's interpretability outputs suggest a more context-aware and linguistically grounded decision-making process compared to the more pattern-based reasoning of LSTM and GRU. These explanations from explainability tools not only validate the model architectures but also highlight the strength of transformer models in tasks requiring semantic understanding.

## 6.2 Reflection

This thesis was motivated by a gap identified in prior research. While many studies explored individual deep learning models for fake news detection, few conducted a direct, side-by-side comparison of LSTM, GRU, and BERT models using a consistent dataset, evaluation pipeline, and interpretability techniques. Additionally, computational efficiency and model explainability were often underexplored in earlier works.

Through this study, we addressed these gaps in following ways:

- We performed a systematic comparison of three DL architectures LSTM, GRU, and BERT using the same Kaggle fake news dataset. This ensured a fair and consistent evaluation of their relative strengths and weaknesses.
- We integrated XAI tools, like LIME and SHAP, to interpret the predictions of all three models. This provided valuable insights into how each model reasons about the input text and highlighted differences in their interpretability.
- We included an analysis of computational efficiency, comparing training time and resource utilization across models, which is important for real-world deployment but often overlooked in prior studies.

By combining performance evaluation, interpretability analysis, and efficiency within a same environment, this thesis contributes a more holistic understanding of the trade-offs involved in selecting DL models for fake news detection. By doing this, it contributes to closing the gap between academic research and practical, real-world AI applications in this field.

# Chapter 7

# Conclusions and Future Work

## 7.1 Conclusion

This thesis has explored the effectiveness of DL models in the task of fake news detection, with a particular focus on comparing RNNs (LSTM and GRU) against transformer-based models such as BERT The results clearly demonstrate that deep contextual models significantly enhance performance in this domain.

Among all the models evaluated, BERT model achieved the highest accuracy, nearing perfect classification. Its ability to capture bidirectional context and use self-attention mechanisms to successfully identify subtle patterns in the text that often distinguish fake news from real news. This advantage was especially apparent in complex or ambiguous examples where traditional models struggled.

In comparison, LSTM and GRU models also performed well, with accuracies of 98% and 93%, respectively. These percentages verify that sequential models are still useful in text classification tasks. But later on they were overtaken by BERT because it has a better comprehension of long range and contextual relationships dependencies in language.

The study further highlighted the importance of interpretability in AI models. Applying explainability methods such as LIME and SHAP revealed that all models base their predictions on meaningful textual cues including named entities and sentiment-laden words. This transparency helps build user trust and aligns model decisions with human reasoning.

Nonetheless, the research has some limitations worth noting. The limitations are:

- The experiments were conducted on a single English-language dataset, which may limit generalizability to other languages, domains, or noisy real-world data such as social media posts.
- The dataset contains well-structured news articles, whereas real-world fake news often appears in noisy, informal formats (e.g., tweets, memes, user comments). Our models were not tested on such data.
- We did not test the models on external datasets like LIAR or PolitiFact to evaluate their robustness and generalization.
- We did not evaluate how the models perform against manipulated or adversarial fake news inputs, which is increasingly relevant in real-world misinformation scenarios.

- Additionally, model evaluation was limited to offline accuracy metrics without deployment in real-time settings.

Overall, these findings emphasize the critical role of deep contextual modeling in combating misinformation. They provide a foundation for extending fake news detection research toward broader, more practical, and more interpretable AI systems.

## 7.2 Future Work

Building on the findings and limitations observed in this thesis, several specific and practical directions for future research are recommended:

**1. Multilingual and Cross-Lingual Evaluation:** This thesis focused exclusively on an English-language dataset, limiting the assessment of model generalization to other languages. Future work should extend the study to multilingual datasets, such as the multilingual FakeNewsNet or PAN CLEF datasets, to investigate whether models like BERT maintain their high accuracy across languages and cultural contexts. This would also address technical challenges around tokenization, embedding, and language-specific nuances that affect model performance.

**2. Application to Noisy and Informal Text:** Although the dataset used here mostly consists of well-written news articles, fake news frequently spreads on social media sites where the text is brief, informal, and noisy (e.g., tweets, Reddit comments). Spelling mistakes, slang, emojis, and abbreviated language were not present in this study but are crucial in real-world detection systems. Therefore, future research should apply and modify the models to these difficult domains to assess robustness against them.

**3. Robustness Against Adversarial Manipulation:** In order to avoid detection, fake news producers may purposefully alter text. An important future direction is testing model robustness against adversarial attacks such as paraphrasing, synonym substitution, or insertion of irrelevant content. A limitation not covered here is that models could be hardened using methods like defensive distillation or adversarial training, which would guarantee dependable performance in adversarial situations.

**4. Exploration of Lightweight Transformer Variants:** While BERT achieved superior accuracy, it is computationally expensive and resource-intensive, limiting real-time or on-device deployment. Future work should compare the performance and efficiency trade-offs of lighter models, like DistilBERT, ALBERT, and MobileBERT, in fake news detection tasks. This is consistent with the resource limitations identified as a problem in this thesis.

**5. Model Interpretability and User Trust:** Although LIME and SHAP were used here to provide initial interpretability insights, more advanced and user-friendly explanation techniques should be developed. Especially in politically or socially sensitive situations, these could include interactive visualizations or counterfactual reasoning that aid users in understanding model decisions. Addressing the ethical issues brought up in this thesis requires improving interpretability.

By addressing the limitations and findings of this study, future research can develop more robust and practical fake news detection systems for real-world scenarios.

# References


[1] "F1 Score in Machine Learning: Intro & Calculation." [Online]. Available: https://www.v7labs.com/blog/f1-score-guide

[2] "A study on the evaluation of tokenizer performance in natural language processing," iSSN: 0883-9514. [Online]. Available: https://www.tandfonline.com/doi/epdf/10.1080/08839514.2023.2175112?needAccess=true

[3] "Understanding the Confusion Matrix in Machine Learning," section: GBlog. [Online]. Available: https://www.geeksforgeeks.org/confusion-matrix-machine-learning/

[4] J. Alghamdi, Y. Lin, and S. Luo, "A Comparative Study of Machine Learning and Deep Learning Techniques for Fake News Detection," *Information*, vol. 13, no. 12, p. 576, Dec. 2022, number: 12 Publisher: Multidisciplinary Digital Publishing Institute. [Online]. Available: https://www.mdpi.com/2078-2489/13/12/576

[5] S. A. Aljawarneh and S. A. Swedat, "Fake news detection using enhanced bert," *IEEE Transactions on Computational Social Systems*, vol. 11, no. 4, pp. 4843–4850, 2024.

[6] H. Allcott and M. Gentzkow, "Social media and fake news in the 2016 election," *Journal of Economic Perspectives*, vol. 31, no. 2, p. 211–36, May 2017. [Online]. Available: https://www.aeaweb.org/articles?id=10.1257/jep.31.2.211

[7] P. Bahad, P. Saxena, and R. Kamal, "Fake News Detection using Bi-directional LSTM-Recurrent Neural Network," *Procedia Computer Science*, vol. 165, pp. 74–82, Jan. 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050920300806

[8] G. Bebis and M. Georgiopoulos, "Feed-forward neural networks," *IEEE Potentials*, vol. 13, no. 4, pp. 27–31, 1994.

[9] Y. Bengio, I. Goodfellow, A. Courville *et al.*, *Deep learning*. MIT press Cambridge, MA, USA, 2017, vol. 1.

[10] E. Bisong, "Matplotlib and seaborn," in *Building machine learning and deep learning models on google cloud platform: A comprehensive guide for beginners*. Springer, 2019, pp. 151–165.

[11] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," 2014. [Online]. Available: https://arxiv.org/abs/1406.1078

[12] A. Choudhary and A. Arora, "Assessment of bidirectional transformer encoder model and attention based bidirectional LSTM language models for fake news detection," *Journal of Retailing and Consumer Services*, vol. 76, p. 103545, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0969698923002965

[13] J. Davis and M. Goadrich, "The relationship between precision-recall and roc curves," in *Proceedings of the 23rd International Conference on Machine Learning*, ser. ICML '06. New York, NY, USA: Association for Computing Machinery, 2006, p. 233–240. [Online]. Available: https://doi.org/10.1145/1143844.1143874

[14] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, J. Burstein, C. Doran, and T. Solorio, Eds. Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 4171–4186. [Online]. Available: https://aclanthology.org/N19-1423/

[15] S. Girgis, E. Amer, and M. Gadallah, "Deep learning algorithms for detecting fake news in online text," in *2018 13th International Conference on Computer Engineering and Systems (ICCES)*, 2018, pp. 93–97.

[16] A. Gulli and S. Pal, *Deep learning with Keras*. Packt Publishing Ltd, 2017.

[17] V. V. V. R. Gurram and J. M. Nalam, "Automated detection of fake news in natural language processing: A comparative study of tf-idf and lexical-based stance detection with logistic regression," 2024.

[18] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant, "Array programming with NumPy," *Nature*, vol. 585, no. 7825, pp. 357–362, Sep. 2020, publisher: Nature Publishing Group. [Online]. Available: https://www.nature.com/articles/s41586-020-2649-2

[19] J. Heaton, "Ian Goodfellow, Yoshua Bengio, and Aaron Courville: Deep learning," *Genetic Programming and Evolvable Machines*, vol. 19, no. 1, pp. 305–307, Jun. 2018. [Online]. Available: https://doi.org/10.1007/s10710-017-9314-z

[20] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[21] B. Jikadara. (2022) Fake news detection. [Online]. Available: https://www.kaggle.com/datasets/bhavikjikadara/fake-news-detection

[22] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," Jan. 2017, arXiv:1412.6980 [cs]. [Online]. Available: http://arxiv.org/abs/1412.6980

[23] A. Kumar, "Micro-average, Macro-average, Weighting: Precision, Recall, F1-Score," Dec. 2023. [Online]. Available: https://vitalflux.com/micro-average-macro-average-scoring-metrics-multi-class-classification-python/

[24] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," *Advances in neural information processing systems*, vol. 30, 2017.

[25] G. S. Mahara and S. Gangele, "Fake news detection: A rnn-lstm, bi-lstm based deep learning approach," in *2022 IEEE 1st International Conference on Data, Decision and Systems (ICDDS)*, 2022, pp. 01–06.

[26] A. Mandelbaum and A. Shalev, "Word Embeddings and Their Use In Sentence Classification Tasks," Oct. 2016, arXiv:1610.08229 [cs]. [Online]. Available: http://arxiv.org/abs/1610.08229

[27] C. Manning and H. Schutze, *Foundations of Statistical Natural Language Processing*. MIT Press, May 1999, google-Books-ID: YiFDxbEX3SUC.

[28] W. McKinney *et al.*, "pandas: a foundational python library for data analysis and statistics," *Python for high performance and scientific computing*, vol. 14, no. 9, pp. 1–9, 2011.

[29] E. J. Michaud, Z. Liu, and M. Tegmark, "Precision Machine Learning," *Entropy*, vol. 25, no. 1, p. 175, Jan. 2023, number: 1 Publisher: Multidisciplinary Digital Publishing Institute. [Online]. Available: https://www.mdpi.com/1099-4300/25/1/175

[30] B. Pang, E. Nijkamp, and Y. N. Wu, "Deep learning with tensorflow: A review," *Journal of Educational and Behavioral Statistics*, vol. 45, no. 2, pp. 227–248, 2020.

[31] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, "Scikit-learn: Machine learning in python," *the Journal of machine Learning research*, vol. 12, pp. 2825–2830, 2011.

[32] N. Rai, D. Kumar, N. Kaushik, C. Raj, and A. Ali, "Fake News Classification using transformer based enhanced LSTM and BERT," *International Journal of Cognitive Computing in Engineering*, vol. 3, pp. 98–105, Jun. 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2666307422000092

[33] A. Ramzan, R. H. Ali, N. Ali, and A. Khan, "Enhancing fake news detection using bert: A comparative analysis of logistic regression, rfc, lstm and bert," in *2024 International Conference on IT and Industrial Technologies (ICIT)*, 2024, pp. 1–6.

[34] S. M. Reddy, C. Suman, S. Saha, and P. Bhattacharyya, "A gru-based fake news prediction system: Working notes for urdufake-fire 2020." in *FIRE (Working Notes)*, 2020, pp. 464–468.

[35] R. Rizal, A. Faturahman, A. Impron, I. Darmawan, E. Haerani, and A. Rahmatulloh, "Unveiling the truth: Detecting fake news using svm and tf-idf," in

*2025 International Conference on Advancement in Data Science, E-learning and Information System (ICADEIS)*. IEEE, 2025, pp. 1–6.

[36] K. Shu, S. Wang, and H. Liu, "Beyond News Contents: The Role of Social Context for Fake News Detection," in *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, ser. WSDM '19. New York, NY, USA: Association for Computing Machinery, Jan. 2019, pp. 312–320. [Online]. Available: https://dl.acm.org/doi/10.1145/3289600.3290994

[37] E. Strubell, A. Ganesh, and A. McCallum, "Energy and policy considerations for deep learning in nlp," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019, pp. 3645–3650.

[38] P. M. Subhash, D. Gupta, S. Palaniswamy, and M. Venugopalan, "Fake news detection using deep learning and transformer-based model," in *2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, 2023, pp. 1–6.

[39] G. Visani, E. Bagli, F. Chesani, A. Poluzzi, and D. Capuzzo, "Statistical stability indices for LIME: Obtaining reliable explanations for machine learning models," *Journal of the Operational Research Society*, Jan. 2022, publisher: Taylor & Francis. [Online]. Available: https://www.tandfonline.com/doi/abs/10.1080/01605682.2020.1865846

[40] Wikipedia contributors, "Gated recurrent unit — Wikipedia, the free encyclopedia," https://en.wikipedia.org/w/index.php?title=Gated_recurrent_unit&oldid=1266937743, 2025, [Online; accessed 30-April-2025].

[41] ——, "Long short-term memory — Wikipedia, the free encyclopedia," https://en.wikipedia.org/w/index.php?title=Long_short-term_memory&oldid=1280106018, 2025, [Online; accessed 30-April-2025].

[42] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. Le Scao, S. Gugger, M. Drame, Q. Lhoest, and A. Rush, "Transformers: State-of-the-art natural language processing," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, Q. Liu and D. Schlangen, Eds. Online: Association for Computational Linguistics, Oct. 2020, pp. 38–45. [Online]. Available: https://aclanthology.org/2020.emnlp-demos.6/

[43] A. Yerlekar, N. Mungale, and S. Wazalwar, "A multinomial technique for detecting fake news using the naive bayes classifier," in *2021 International Conference on Computational Intelligence and Computing Applications (ICCICA)*. IEEE, 2021, pp. 1–5.

[44] M. Yin, J. Wortman Vaughan, and H. Wallach, "Understanding the effect of accuracy on trust in machine learning models," in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, ser. CHI '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 1–12. [Online]. Available: https://doi.org/10.1145/3290605.3300509

[45] J. Zarocostas, "How to fight an infodemic," *The Lancet*, vol. 395, no. 10225, p. 676, Feb. 2020, publisher: Elsevier. [Online]. Available: https://www.thelancet.com/journals/lancet/article/PIIS0140-6736(20)30461-X/fulltext?onwardjourney=584162_v2

# Appendix A

# Supplemental Information


Faculty of Faculty, Blekinge Institute of Technology, 371 79 Karlskrona, Sweden