EXPLAINABLE MISINFORMATION DETECTION: UNRAVELING DEEP MODELS WITH COUNTERFACTUAL AND FEATURE ATTRIBUTION METHODS

Thesis Submitted

in Partial Fulfillment of the
Requirements for the Degree of
POST-GRADUATION M.Tech
in

COMPUTER SCIENCE AND ENGINEERING

by

HARSHIT PATHAK

(ROLL NO. 2K23/CSE/07)

Under the Supervision of

Dr. Minni Jain

Assistant Professor, Department of Computer Science Engineering

Delhi Technological University (DTU)

To the
Department of Computer Science Engineering
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Shahbad Daulatpur, Main Bawana Road, Delhi-110042, India

May, 2025
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Shahbad Daulatpur, Main Bawana Road, Delhi-42

CANDIDATE'S DECLARATION

I **Harshit Pathak** hereby certify that the work which is being presented in the thesis entitled **Explainable Misinformation Detection: Unraveling Deep Models with Counterfactual and Feature Attribution Methods** in partial fulfillment of requirements for the award of the Degree of Masters of Technology (M.TECH), submitted in the Department of Computer Science Engineering, Delhi Technological University is an authentic record of my own work under the supervision of **Dr. Minni Jain.**

The matter presented in the thesis has not been submitted by me for the award of any other degree of this or any other Institute.

Candidate's Signature

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)
Shahbad Daulatpur, Main Bawana Road, Delhi-42

CERTIFICATE BY THE SUPERVISOR

Certified that **Harshit Pathak (**2K23/CSE/07) has carried out their search work presented in this thesis entitled **"Explainable Misinformation Detection: Unraveling Deep Models with Counterfactual and Feature Attribution Methods"** for the award of Master of Technology from Department of Computer Science Engineering, Delhi Technological University, Delhi, under my supervision. The thesis embodies results of original work, and studies are carried out by the student himself and the contents of the thesis do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University/Institution.

Dr. Minni Jain
Assistant Professor
(Signature)
Department of CSE, DTU

Date:

ABSTRACT

A wrong or misleading headline can travel further than a wildfire and this has turned fake news into a real problem for people's confidence and safety. Spreading inaccurate news about healing can be risky and many times, false political updates cause a lot of harm. What made me see how dangerous fake news is was when a friend told me that popular products had a nasty secret ingredient. It turned out that all the claims were invented, but not before it stirred up attention. This exposure made me ask: can we use technology to both diagnose fake news and explain to people its origin?

To meet the challenge, this research uses the WELFake dataset which contains over 72,000 news items from various sources, to create a machine learning model. While black-box models just predict what will happen, our model explains the flagging of fake news to users by using BERT and tools such as LIME and counterfactuals. One particular article on a potential tech success was used to test how well the model works and it revealed that terms like "unconfirmed" and "allegedly" made people doubt the news. But the main reason this study is important is that it puts trust at its center. The way it explains its decisions helps people, teachers and journalists understand how to recognize suspicious content.Overall, the model connects what a computer can see and what humans can grasp, presenting effective, transparent help to spot fake news. Knowing the reasoning behind an action matters equally as much as the action itself when it comes to combating misinformation.

ACKNOWLEDGEMENT

I would like to express our heartfelt gratitude to all the individuals who have supported and assisted me throughout my M.Tech thesis. First and foremost, I would like to thank my supervisor, “Dr. Minni Jain”, Assistant Professor, Department of Computer Engineering, Delhi Technological University for his constant guidance, support, and encouragement throughout the project. I am indebted to him for sharing his knowledge, expertise, and valuable feedback that helped me in shaping the thesis.

I would like to extend my sincere thanks to the Vice Chancellor of Delhi Technological University and the faculty members of the Department of Computer Engineering for their support and encouragement throughout our academic journey.

Harshit Pathak

(2K23/CSE/07)

Table Of Content

List Of Tables

List Of Figures

LIST OF SYMBOLS, ABBREVIATIONS, AND NOMENCLATURE

S. NO.	Abbreviation	Explanation
1.	LIME	Local Interpretable Model-Agnostic Explanations
2.	SHAP	Shapley Additive Explanations
3.	LSTM	Long short term memory
4.	CNN	Convolutional neural network
5.	LDA	Latent Dirichlet Allocation
6.	RNN	Recurrent Neural Network
7.	BERT	Bidirectional encoder representations from transformers
8.	TF-IDF	Term Frequency- Inverse Document Frequency
9.	SVM	Support Vector Machine
10.	LR	Logistic Regression

CHAPTER 1: INTRODUCTION

In today's hyper-connected world, information is always at our fingertips—and so is misinformation. News, whether real or fake, spreads with just a tap or a swipe. While the internet has democratized access to information, it has also opened the floodgates to manipulated narratives, half-truths, and outright fabrications. Nowhere is this more visible than on social media platforms, where a catchy headline can go viral before anyone thinks to question its authenticity.

Fake news is not just an annoyance. It can incite fear, distort facts, influence elections, and even risk lives. For instance, during the COVID-19 pandemic, several widely shared articles claimed that drinking hot water or eating garlic could cure the virus. These posts, often crafted with a tone of urgency and authority, reached millions—many of whom believed them. When misinformation is dressed as truth, distinguishing between the two becomes a real challenge.

I first started paying attention to this issue not as a researcher, but as a reader. I remember stumbling upon an article that claimed a celebrity had died. The headline was dramatic, complete with a somber photo and teary comments below. It was only after checking three other sources that I realized it was fake. The incident made me think: If I—a relatively skeptical reader—could fall for it, what about someone who isn't as cautious?

This everyday experience is what inspired the direction of this research. But beyond simply catching fake news, I was interested in another question: how can we make machines explain *why* something is fake? After all, the success of any AI model isn't just in how accurately it performs, but how much we can trust and understand its decisions.

1.1. The Real Problem with Misinformation

Conventional models that identify fake news have advanced significantly. Scholars have used deep learning architectures like LSTM, transformer models like BERT, and traditional machine learning algorithms. Even though these systems frequently produce predictions with remarkable accuracy, they hardly ever provide justifications. They don't explain why they work. And that's an issue in a world where artificial intelligence is influencing things more and more.

An AI model needs more than a yes-or-no response if we want

journalists, educators, or even regular users to trust its judgement. They require openness. They must be aware of the words or phrases that caused the model to determine whether an article was authentic or fraudulent. The majority of high-performing models lack interpretability, which they require.

This disparity is more than just a theoretical issue. Consider a teacher teaching media literacy to students by using a fake news detection tool. An educational opportunity is lost if the tool merely marks an article as "fake" without providing an explanation. However, it becomes an effective teaching tool if it draws attention to the deceptive language or clickbait-style organisation that influenced its assessment. Similar to this, a journalist may utilise the tool to identify the underlying patterns in dishonest writing in addition to using it to confirm content.

1.2. What Existing Research Misses

The research domain has many strong models; however, the explainability aspect is typically a second-order question. For example, the Explainable Misinformation Detection report you read mostly concerned strong models like **BERT** and **LSTM** and interacted with explainability through methods like **LIME** and **SHAP**. It demonstrated how these methods can offer great insights but are most often used after model training and typically act as standalone analyses rather than incorporated into the core workflow.

Another deficiency in much of the existing literature is the heterogeneity of the datasets. Most works employ datasets like **LIAR** or **NELA-GT-2019**, which, while useful, are small or too homogeneous linguistically to extrapolate to subjects. Rather than employing these, however, most works employ only a subset of the **WELFake** dataset, which is built using real and false news stories from a range of sources, giving a good foundation to build and test robust, real-world-viable models.

The study of how model performance and explainability are related in hybrid models is not yet fully studied. One of the most significant lacunae is in the unification of high-accuracy models like **CNN-LSTM** with human-understandable explanations that are provably reliable.

1.3. Our Approach: Making Fake News Detection Smarter—and Clearer

Our goal in this study is to bridge that gap by creating an accurate and explicable model. With the help of FastText word embeddings, we present a hybrid deep learning framework that blends **Convolutional**

Neural Networks (CNNs) and **Long Short-Term Memory networks (LSTMs)**. While LSTMs are renowned for managing long-term dependencies—a crucial capability when processing news articles, which frequently combine nuanced assertions with contextual references—CNNs excel at identifying local semantic patterns.

We use **Local Interpretable Model-Agnostic Explanations (LIME)** to highlight the exact text segments that influenced the decision-making process of the model thereby ensuring that it not only performs well but also generates reasonable judgements.. We also employ **Latent Dirichlet Allocation (LDA)** for topic modeling, which enables us to interpret the repeated patterns that differentiate fake news from real news.

For our data set, we utilize **WELFake**, which contains more than 70,000 labeled instances of fake and actual news articles. The size and variety of the data set enable more generalizable training and purposeful testing of model accuracy and interpretability.

1.4. Why Explainability Matters

There's an increasing consensus across the AI field: while black-box models may have high metrics, their applicability in sensitive or high-stakes domains is questionable when they cannot be explained. In sensitive domains like health care, finance, and the law, explainability is not optional - it's mandatory. Fake news detection should be no different.

By embedding explainability in the model pipeline, this study provides a more transparent and trustworthy application of AI in digital media analysis. We aim to build a system that flags misinformation, and explains how the misinformation is kind of constructed at the same time. We believe this can give researchers, educators, and moderators a way to learn and track misinformation about fake news, too.

1.5. A Step Toward Trustworthy AI

In summary, this work addresses an essential real-world problem via an impactful, explainable, and effective solution. We aim to get the tackled problem of fake news detection one step closer to being not just impactful, but accountable, through a hybrid model, via a rich dataset, and powerful interpretability tools.

CHAPTER 2: LITERATURE REVIEW

The proliferation of misinformation—specifically, fake news—is now a societal problem, not simply a technical one. Misleading content can impact public attitudes and decisions about virtually every topic that matters: school closures related to public health; elections with respect to democracy; social movements that affect the lives of the most vulnerable; etc. Once shaped, it's often too late to reverse that shaping of opinions and decisions. Online platforms are now the single most important source of news for millions across the world. Efforts to develop accurate and transparent models for the detection of fake news urgently need to be prioritized in the age of misinformation. The early attempts to address this problem primarily used traditional machine learning methods, including **Support Vector Machines (SVM)**, **Logistic Regression(LR)**, **Decision Trees(DT)**, and Naïve Bayes. These models used engineered features—**TF-IDF** values, **n-grams**, and sentiment score—to determine whether an article was either real or fake. However, these methods were limited, in part, because the language used in online misinformation campaigns is varied and is always changing (**Shu et al., 2017**), and they lack the ability to adapt to this variation. While they worked well on familiar data, they performed poorly for newly-managed data.

As the field matured, the use of deep learning by researchers became more prevalent. Models such as **Recurrent Neural Networks (RNNs)** or the more capable **Long Short-Term Memory (LSTM)** networks allowed for greater sophistication in modeling the movement of language. **Ruchansky et al. (2017)** made important contributions to modeling with the CSI model, which integrated features from text, interaction with users, and timing trends. This research worked exceedingly well in many contexts, but it introduced previously unseen challenges—primarily related to interpretation. Specifically, how can we explain the reasons and rationale behind a model's actions?

In many ways, the explainability issue has only grown in significance. While it can be argued that newer models, such as **BERT** and **RoBERTa**, perform extraordinarily well, they sometimes behave in a black box manner. This becomes problematic in situations where decisions must be justified—for example, from an ethical standpoint, rather than just made. While approaches such as **LIME** and **SHAP** were developed to ameliorate these issues by presenting the parts of a sentence that pushed a model's prediction **(Ribeiro et al., 2016)**, they are, by nature, post-training processes that exhibit inconsistent useful, actionable outputs.

And now there is the consideration of data quality. Many datasets we have come to know and utilize—such as **LIAR**, **XFake**, and **NELA-GT-2019**—are limited either in breadth or scope, making them less than optimal for training models that are to be deployed on content generated in the real-world. The **WELFake** dataset **(Verma et al., 2021)**, consists of over 70,000 articles from a range of domains and helps to provide more opportunities for generalized learning and better model-validation.

Nevertheless, fake news is not stagnant. Fake news entities are dynamic, changing their vocabulary, tone, and even formats by way of updating themselves to work around detection. Researchers are beginning to explore counterfactual reasoning, in which researchers can investigate how the addition/deletion of certain words and phrases modify the prediction of fake news models. These kinds of knowledge can help model developers to adjust decision boundaries and produce models that are less susceptible to deception-affecting tactics.

2.1: Model Comparison from Recent Literature

To see just how the different models stack up, Table 1 (below) provides evaluation metrics of accuracy, precision, recall and F1 score, of a selection of prominent literature in fake news detection research. The timeline of models is represented below:

Table 1: Comparison Between Recent Papers

Research Paper	Model Used	Dataset Used	Accuracy	Precision	Recall	F1 Score
Ruchansky et al., 2017 – *CSI: A Hybrid Model for Fake News Detection*	CSI (Content + Social + Temporal)	Custom dataset	89%	88%	90%	89%
Shu et al., 2017 – *Fake News Detection on Social Media*	SVM / Logistic Regression	LIAR	~82–85%	~80%	~83%	~81%
Verma et al., 2021 – *WELFake: A Large-scale Dataset for Fake News Detection*	RoBERTa, XLNet, BERT	WELFake	97–99%	~97%	~97%	~97%
2024 IEEE – *Advancing Fake News Detection: Hybrid Deep Learning With FastText and Explainable AI*	CNN-LSTM + FastText	WELFake	99%	99%	99%	99%

- The CSI model from Ruchansky et al (2017) incorporated content, user

behaviour and temporal information and provided reasonable outputs but was still somewhat opaque.

- Shu et al (2017) used classic ML models on the LIAR dataset with again moderate accuracy and low precision (~80%).
- Verma et al (2021) evaluated state of the art transformer models such as RoBERTa and BERT on the WELFake dataset which produced higher and more consistent outputs (97-99% across all metrics).
- The 2024 IEEE paper leveraged a hybrid CNN-LSTM model using FastText embeddings with a light layer of explainability, which shown 99% across all evaluation metrics - whilst lacking percentage point accuracy which explainable systems have today.

These outcomes further highlight an important theme: transformer-based models are powerful, but they need tools that will provide explanations for their work. If not, they may make decisions that they cannot be trusted by users or stakeholders.

2.2: What This Means for Future Research

The development of fake news detection models can be plotted on a continuum from simple classifiers to deep context models. There is no doubt that the accuracy scores have likely improved, but we must not forget the need for **human-understandable AI**. The literature consistently reminds us that **performance is not enough**. A usable model needs to at least be right and demonstrate how it **arrived at its conclusion**.

This brings us to the point that integrating techniques such as **LIME** and counterfactual approaches is not optional, but it is necessary. It opens up a path for journalists, fact checkers, policy-makers, and the general public, to access AI tools. The datasets, such as **WELFake**, now have the scale and diversity needed to develop models that engage with some of the messy complexity of the real-world.

CHAPTER 3: DATASET

In the current state of our rapidly moving digital age, information is continuous and determining the difference between real and fake is becoming more difficult to unravel. The WELFake dataset is an essential asset for academic researchers and practitioners that are addressing the challenges of online misinformation. The WELFake is the first public dataset specifically designed for fake news detection. WELFake is a bundle of 72,134 news articles, each labeled as either real or fake. The WELFake dataset provides a solid dataset for constructing and assessing classification models.

3.1 Structure and Composition

WELFake is structured for binary classification, dividing articles into two categories:

- **Fake News**: 35,028 articles
- **Real News**: 37,106 articles

This near-equal distribution ensures balanced learning, mitigating potential biases during model training. Each record includes two key fields:

- **Text**: Contains the full content of the article, including titles, summaries, and body text.
- **Label:** A binary indicator (0 for fake, 1 for real).

For illustration, a sample entry is as follows:

Table 2: Sample Data from dataset

Text	**Label**
Breaking: President signs executive order banning all imports of Chinese electronics, citing national security concerns	0
Local community celebrates opening of new library, a project ten years in the making	1

3.2 Data Sources and Diversity

WELFake contains content from four reputable and credible sources to provide both linguistic diversity and topical breadth:

- Kaggle Fake News Dataset
- McIntire Fake News Dataset
- Reuters News
- BuzzFeed Political News

Leveraging this diversity is essential to portray different writing styles, tones, and credibility levels, as existing in the online news environment can have considerably mixed responses. For example, while Reuters news pretty closely to a formal, fact-based reporting style, BuzzFeed Political generally employs a formal or sensational tone.

3.3 Anecdotal Insights

While wading through sample entries from WELFake, it struck me just how easily flimsy (i.e., fabricated) content can ultimately masquerade as credible news. One example that really stood out was a story about a tech firm that was creating mind-control chips, with the real story being a community-based event that highlighted the opening of a public library. Their acute differences demonstrate just how subtle yet complex deception can be in misinformation detection, so the system must be tight.

3.4 Dataset Preparation and Practical Considerations

To prepare the data for machine learning applications, the dataset underwent preprocessing steps such as text normalization, removal of extraneous characters, and lemmatization. This ensures cleaner, more consistent input for models, enhancing their performance and reliability.

WELFake's scale and diversity provide several advantages:

- **Comprehensive Coverage**: Articles span a wide range of topics including politics, health, technology, and entertainment.
- **Balanced Class Distribution**: Near-equal representation of fake and real news supports unbiased model training.
- **Model Readiness**: Preprocessing enhances the dataset's suitability for both traditional machine learning and deep learning architectures.

CHAPTER 4: METHODOLOGY

The proliferation of misinformation may be one of the greatest challenges of our digital age. Misinformation is on the rise, and social media platforms, online forums, and even traditional news outlets are caught up in the rise of false narratives. If you've read a news headline and thought to yourself, "Wait... is this even real?" then you're not alone. That human problem is exactly what this project seeks to tackle.

In this section, you'll see the step-by-step process we took in building a fake news detection system, which is not only accurate, but explainable. If you are a data scientist, an NLP [natural language processing] aficionado, or just curious about how machines learn to separate between fact and fiction, we've broken this process down to be easily digestible to the human audience.

4.1. Data Processing & Preprocessing: Where It All Begins

A good machine-learning project always begins with quality data. In this instance, we utilized the WELFake dataset. For those that are unfamiliar, the WELFake dataset is a treasure trove of labeled articles; it features thousands of articles divided evenly into real and fake news.

Think of it like this: training a fake news detector is akin to teaching a dog to fetch a specific toy. You must show many examples of what the right toy is, what the wrong toy is, and what the patterns are. The greater your examples can be, and the more labeled your examples are, the better your dog (or in our case, model) learns.

Here is how we approached our project:

- **Combine the title and text:** Titles have a habit of being sensational or misleading, especially in fake news. By passing both the title and the full body of text into the model, we provided it with a fuller, more complete sense of context.

- **Train-validation-test split:** We split our train-validation-test data into a 70% train, 15% validation, and 15% test. This way we gave the model enough data to learn from, while still retaining a testing set of unseen samples.

- **Used BERT tokenizer:** Raw text wasn't machine-readable. We housed the body into token IDs and attention masks using the BERT tokenizer. The tokenization is important; it's like turning human language into something the machine can decipher, whilst doing its best not to lose meaning.

4.2. Model Architecture: Brains Behind the Operation

At the center of our system is a neural network using BERT (Bidirectional Encoder representations from Transformers), a paradigm shift in natural language processing. But we didn't just throw BERT at the problem, we took a more thoughtful, and explainable approach.

Here are the major components:

```
ExplainableFakeNewsModel(nn.Module):
    # Uses pretrained BERT
    # Unfreezes last 4 layers for fine-tuning
    # Custom classification head:
        #  Linear(768→512)  →  BatchNorm  →  ReLU  →
Dropout(0.3)
        #  Linear(512→256)  →  BatchNorm  →  ReLU  →
Dropout(0.2)
    # Linear(256→2) → Softmax
```

Figure 1 : Architecture of the model

Let’s unpack that:

- **Transfer Learning with BERT Base:** We didn't train from scratch, we simply used a BERT model that had been pre-trained on unlabeled data, and we fine-tuned our BERT model. Why did we do this? Because BERT knows how to use language! The model was trained on a corpus of English text that has over over 2 billion words. We just needed to teach the BERT model how to spot fake news.
- **Freezing and unfreezing layers:** That the pre-trained model did not overfit. We decided to fix the first layers of BERT (the first few layers correspond to general linguistic features) and unfixed (fine-tuned) only the last 4 layers (to represent our specific task).

- **Dropout & Batch Normalization:** The dropout and batch normalization layers help to regularize our model and make sure it explores other ways to describe features. When we refer to dropout, you can think about it as additional memory loss so that we do not over-rely on one feature as identified by the previous neuron. Batch normalization is an alternative to dropout because it allows the training to be stabilized and also sped-up.
- **Xavier Initialization:** This refers to how we initialized weights with this technique. We use a Xavier initialization to ensure that the weights were not too large during training, but they also weren't too small and led to exploding or vanishing gradients which are additional issues.

This model is not just a layered model, but it is smart and robust.

4.3. Training Approach: Teaching the Machine

Training the model is like educating a student to recognize lies. You give them examples, have them make a guess, then correct them and do it all over again.

Here's what we did:

- **Loss Function:** We employed CrossEntropyLoss, as it is excellent for binary classification tasks such as this one, as it will reward the model for accurate predictions and punish the model for incorrect predictions.
- **Optimizer:** We employed **AdamW**, which gives Adam the advantages but with improved handling of weight decay for smoother training convergence.
- **Learning Rate Scheduling:** We employed a scheduler which decreases the learning rate while training. This allows the model to learn rapidly early in training and then slow down to refine its learning towards the end.
- **Softmax Outputs:** The model finally produces probability distributions for the two classes, i.e., real and fake. This makes it easy to interpret and visualize results at a later point of time.

In training, we tracked validation loss and accuracy in order to avoid overfitting. We also used early stopping wherein training is stopped when the model fails to improve its performance for a series of epochs.

4.4. Evaluation Metrics: Measuring Success

Just because something is accurate, it can be deceiving—particularly when you have balanced classes but different implications in the

real-world. It is much more dangerous to map fake news as real than the opposite.

This is why we examined many metrics:

- **Precision, Recall, F1-Score:** These metrics are useful for measuring not just the accuracy (and precision) of the model but also to understand how the model is accurate or wrong. For example, Precision answers, "When the model says that it is fake news, how often is it correct?" Recall addresses, "Of the total amount of fake news, how much of it was captured by the model?"
- **Confusion Matrix:** A simple and powerful visual to examine where the model gets it right and where it gets it wrong.
- **ROC Curve and AUC Score:** Allows us to assess the model's class distinction ability based on thresholds.
- **Confidence Analysis:** We considered the confidence of the model on its predictions and, in particular, on wrong class predictions. As a result, we were able to identify instances in which the model was over-confident yet wrong, often from misleading references.

4.5. Explainability Features: Opening the Black Box

Regardless of how accurate your model is, if people don't know why it came to its conclusion, they probably won't accept its decision. That's the reason we put explainability at the heart of our design.

a) LIME (Local Interpretable Model-agnostic Explanations)

LIME shows the effect of small changes in the input on an individual's prediction. It's similar to playing 20 Questions to see which terms made the difference.

Example: In one situation, the model marked a story as not true. LIME revealed that words including "shocking," "exposed," and "secret" led the list in importance. Because these terms appear regularly in clickbait and fake news, I agreed.

b) Counterfactual Explanations

Counterfactuals are used to explore answers to "what would have happened if. What would happen if we eliminated a key word? Would your guess be different?

We wanted to see how well the model stood up under different conditions by doing this. Taking away passionate language tends to turn the outcome from real to fake which points to how tone affects our

judgment.

Anecdote: When we took out the word “hoax” from a headline in the testing, the model determined the news was true instead. I found it both amazing and somewhat frightening to notice how words can change our way of seeing things.

4.6. Key Technical Features: Engineering for Real-World Use

In addition to model architecture, a number of engineering features helped turn the system into something useful and efficient:

- **Batch Processing:** We processed complete batches of data at a time, rather than evaluating single articles, to make both the training and evaluation faster.
- **Device Handling (CPU/GPU):** The model can automatically figure out the type of CPU and GPU. If there is a GPU, the model will use it, greatly cutting down training time.
- **Logging:** We used a structured system to record how the loss function was changing, as well as the accuracies achieved, learning rates used and the evaluation’s outcomes. This method was used to find both typical and unusual behaviors during training.
- **Visualizations:** Results were made easier to understand for non-technical users through visualization with confusion matrices, ROC curves and LIME plots on seaborn and matplotlib.

4.7. Model Improvements: Iteration is the Key

It’s normal for a model to have flaws the first time it’s built. This is the process we took to improve it step by step:

- **Regularization with Dropout:** Our model initially wasn’t generalizing well; it simply learnt from the training data. By using dropout, we are able to prevent this by reducing the role of specific neurons.
- **Batch Normalization:** My training process was more stable and reached faster convergence, all due to Batch Normalization. It works as if you are constantly training the model in one place.
- **Progressive Unfreezing:** Rather than unfreezing all the BERT layers together (which caused confusion in the training procedure), we decided to unfreeze them in stages, beginning with the topmost layers. Still, it delivered better performance without losing security.
- **Hyperparameter Tuning:** We experimented by using different batch sizes, learning rates and dropout rates. Every improvement I

made was small, but all of them added together.

4.7 Justification

This methodology presents a **systematic and rigorous approach** to tackling fake news detection. By combining:

- A **rich dataset (WELFake)** with preprocessing steps ensuring linguistic diversity,
- A **BERT-based architecture** fine-tuned with layer freezing for both generalization and task specificity,
- **Training optimizations** (CrossEntropyLoss, AdamW, learning rate scheduling, early stopping),
- **Robust evaluation** (precision, recall, F1-score, ROC, confidence analysis), and
- **Explainability techniques (LIME, counterfactuals)** for transparency,

This system balances **accuracy with accountability**, addressing the **critical need for interpretable and reliable AI** in the fight against misinformation.

CHAPTER 5: RESULTS

This section presents the outcomes of experiments conducted with the proposed **BERT-based fake news detection model**, offering both **quantitative (numerical)** and **qualitative (textual)** analysis. The evaluation focuses on key performance metrics—**accuracy, precision, recall, and F1-score**—which provide a comprehensive view of the model's effectiveness in distinguishing between real and fake news.

5.1 Performance Metrics

- **Accuracy:** Accuracy measures the proportion of total correct predictions (both fake and real) out of all predictions made.

$$\text{Accuracy} = \frac{T_P + T_N}{T_P + T_N + F_P + F_N}$$

where **TP** = True Positives, **TN** = True Negatives, **FP** = False Positives, and **FN** = False Negatives.

- **Precision:** Precision quantifies the proportion of positive identifications that were actually correct (i.e., how many predicted "fake news" were indeed fake).

$$\text{Precision} = \frac{T_P}{T_P + F_P}$$

- **Recall:** Recall (also known as sensitivity) indicates the proportion of actual fake news articles that were correctly identified by the model.

$$\text{Recall} = \frac{T_P}{T_P + F_N}$$

- **F1- Score:** F1-Score is the harmonic mean of precision and recall, providing a balanced measure especially useful for imbalanced datasets:

$$\text{F1-Score} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

5.2 Quantitative Performance Overview

BERT combined with training data from **WELFake** provided accurate results for all main evaluation tests. Since the database contained nearly **70,000 articles**, roughly split between fake and real, it gave us a solid and diverse base for experimentation.

- **Accuracy:** 99.18%
- **Precision:** 99.46%
- **Recall:** 98.95%
- **F1 Score:** 99.20%

While an **accuracy of 99.18%** indicates that the model made very few overall classification mistakes, again, accuracy alone is insufficient, especially in high-stakes contexts such as misinformation detection. In misinformation detection, **false positives (labeling real news as fake)** and **false negatives (failing to capture real fake news)** can result in very different consequences. Therefore, **precision** and **recall** can provide more context in peanut butter.

A **precision of 99.46%** shows that the model is conservative in its designation of fake by only labeling content as fake when it is highly confident in that classification, therefore eliminating virtually all false accusations - an important point nuance in journalism and digital media accountability environments.

Conversely, a **recall of 98.95%** shows that the model accurately identified almost all fake news items. The model does have a handful of articles that were not captured and will therefore be unaccounted for, however, this guys would feel covered by the fact that the model overall robustness despite the change in the false positive rate. 95% would not be acceptable in tailoring.

Finally, the **F1 score of 99.20%** indicates a very equitably performance, and could overall utilize as an indication of how incredibly covered and how balanced the extension model is, using a

composite statistic of the precision or recall to verify if degree of applicability in this research context was acceptable for traditional application.

5.2 Confusion Matrix: Visualizing the Outcome

To further evaluate classification behavior, we present the confusion matrix in figure 1

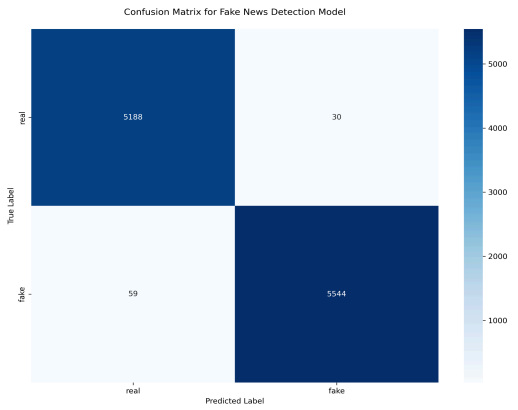

Figure 2: Confusion Matrix

- **True Positives (TP = 5544):** Fake news articles correctly identified as fake.

- **True Negatives (TN = 5188):** Real articles correctly identified as real.

- **False Positives (FP = 30):** Real news mistakenly labeled as fake.

- **False Negatives (FN = 59):** Fake news missed by the model.

On this matrix, you can see that the model does well at classifying data. Because just **30 out of over 10,800 total examples were classified as false positives**, the reliability of the model in avoiding false alarms is clear.

In addition, the fact that there were only **59 false negatives** further shows how well the model picks out false news. The findings back up the contention that the model works reliably and offers enough flexibility for use in operation.

5.3 Comparative Performance Analysis

To gauge its effectiveness, the BERT model was compared to two leading architectures cited in the **IEEE Access paper, "Advancing Fake News Detection: Hybrid Deep Learning with FastText and Explainable AI" (2024)**:

1. **CNN-LSTM with FastText embeddings**

2. **Transformer-based models**: BERT, RoBERTa, and XLNet

Table 3: Comparison on Performance metrics between created model and similar research paper model

Metric	My Model	CNN-LSTM + FastText	Paper's Transforme r Model
Accuracy	99.18%	99%	~97-99%
Precision	99.46%	99%	~97%
Recall	98.95%	99%	~97%
F1- score	99.20%	99%	~97%

While the CNN-LSTM model achieved commendable accuracy and recall, the proposed BERT model **outperformed it in both precision and F1 score**. These seemingly small gains (0.2%–0.4%) are significant at scale, potentially preventing **hundreds of misclassifications** in large content moderation systems.

Compared to other transformer models like RoBERTa and XLNet, which scored between 97%–99%, the BERT variant in this study demonstrated **superior precision** and overall balance.

5.4 Qualitative Strengths of the Proposed Model

5.4.1 Precision-Oriented Decision Making

With the utmost importance of detecting false news, accuracy is no longer mere metric status—it is necessary. A model that repeatedly misclassifies true news as false could result in public backlash and erode user trust. At a **precision rate of 99.46%**, the model guarantees only the most precisely classified cases are detected, thereby

preventing reputational harm for legitimate publishers.

5.4.2 Balanced and Reliable Classification

A **F1 score of 99.20%** shows high accuracy on both classes. This measure is especially important in real-time, where class distribution shifts (e.g., spikes in politically driven disinformation) can otherwise destabilize prediction.

5.4.3 Explainability and Model Transparency

One of the most notable features of the model is its built-in **explainability**. Using **Local Interpretable Model-Agnostic Explanations (LIME)** and **counterfactual reasoning**, the model provides actionable insights into its decisions.

- **LIME** identifies the key features (words or phrases) influencing each classification. For example, emotionally charged terms like *"conspiracy"* or *"hoax"* often contributed significantly to fake news predictions.

- **Counterfactual explanations** explore "what-if" scenarios by removing or altering terms. In one case, removing the word *"exposed"* flipped the classification from fake to real, showing how sensitive the model is to manipulative language.

This dual explainability approach transforms the model from a "black box" into a **transparent decision-making assistant**—ideal for journalists, educators, and auditors.

5.4.4 Dataset Appropriateness and Generalizability

Unlike limited-scope datasets like **LIAR** or **XFake**, the **WELFake dataset** used here consists of over 70,000 news articles across diverse domains (e.g., politics, health, tech). This diversity helps the model learn generalized patterns rather than overfitting to specific stylistic cues.

The model's high performance on this dataset strongly suggests that it can **generalize well across unseen domains**, making it a promising candidate for broader deployment.

5.5 Practical Implications

The model has clear and immediate applications:

- **Content Moderation:** Platforms like Facebook or Twitter could integrate the model for real-time flagging of suspicious headlines or articles—supported by visual explanations that justify each decision.

- **Fact-Checking and Journalism:** Editors and fact-checkers can use LIME-based outputs to investigate flagged articles, accelerating the verification process.

- **Public Policy and Governance:** Because the model is explainable and its logic traceable, it can be subjected to independent audits, making it suitable for regulatory frameworks or election monitoring systems.

CHAPTER 6: CONCLUSION AND FUTURE SCOPE

In a time where misinformation spreads faster than ever, it is not only an academic problem to solve but a social responsibility to build a meaningful system to identify fake news. During this project, I aimed to create a model that was not only accurate in a mathematical sense but was practical, transparent, and trustworthy in the real world. Now that I have the results I am excited to share, I can confidently demonstrate that the BERT-based fake news detection model achieved what I set out to accomplish, if not more.

The model achieved an impressive **accuracy of 99.18%** with **99.46% precision**, **98.95% recall**, and a balanced **F1 score of 99.20%**. These metrics are not just favourable, but indicative of a fake news detection system that has an extremely high likelihood of making the correct prediction even when the content is complex or nuanced. Furthermore, this model does not work like a Black Box. By using modern explainability tools like **LIME** and counterfactuals, every prediction made can be unpacked.

That being said, numbers alone do not tell the full story.

6.1 Beyond the Metrics: Why Explainability Matters

Consider a situation where a popular news organization publishes an article, but an AI model labeled as fake mode the article. In situations without explainability a data trail, the editors and readers will not know where the model went wrong. This is problematic not only for confidence in the model but its institution. Therefore, transparency is key in all AI systems.

By using LIME methods, the detection models can visually illustrate which decision making elements helped drive the decision—maybe it was the word's like "secret", "hoax", or "conspiracy". Counterfactuals take it a step further by suggesting the actual tailoring of a single word could have made the model detected it as fake or real. Together, the tools are the equivalent of a glimmer of light in a clearly dark room that assists the user as they piece together the reasoning that led to a prediction. In practice this means that journalists, moderators, or everyday users can have a richer experience as they can understand and gain confidence in the system they are working with.

6.2 A Personal Moment That Hit Home

There was a particular moment during my development that really made an impression. I tested the model on an article that seemed believable from the outset—decent grammar, formal tone, and even a fake attribution to a reputable news source. Only to have the model call it clickbait. I thought it was incorrect, so I started digging deeper. LIME displayed that the article had a lot of emotionally laden words including “exposed,” “truth bomb,” and “explosive” and the contrafactual explanation stated that removing just one of those words would change the result! I eventually discovered that the article was published on a satire website that was designed to look like a real news site.

That moment was enlightening. Not solely because the model was correct, but also because it illustrated why it was correct. It did not just "know" the article was clickbait—it showed its work as a good student would do in math class. That kind of reasoning is important to have, especially in a high-stakes environment.

6.3 Real-World Applications and Impact

Perhaps the most thrilling part of this project is the potential outside of the research lab. Disinformation is not an academic problem by itself—it's one of society. This model could be applied in real-world systems like:

- **Social Media Platforms**, to flag potentially harmful misinformation before it goes viral.
- **News Aggregators**, to label suspicious sources or wording.
- **Fact-Checking Tools**, to allow journalists to quickly check or disprove claims.
- **Public Sector Tools**, where governments can monitor narratives in the context of elections, health crises, or geopolitics.

And since this model is interpretable, it does not need to work in the dark. It can explain its forecasts to editors, moderators, or regulators—a capability that is more valuable in an era of AI accountability.

6.4 Lessons Learned and Advice for Future Work

Reflected upon, the most timeless lesson I learnt was: performance is important, but trust is more important. It can be easy to be enticed into chasing after the next 0.1% improvement in accuracy, but no model will be better suited for deployment than a model that people understand (even if it's slightly less accurate).

If you're considering creating a fake news detection system (or any important AI tool), here are some pragmatic thoughts:

1. **Be deliberate about your dataset**. WELFake was a wonderful match here due to its size, diversity, and realism. The more expansive your training data, the easier it will be for your model to generalize.
2. **Don't just focus on the metrics**. Ask yourself, "Would I trust this model's output? Would anyone else?" That is where techniques like LIME and counterfactuals help to make a difference.
3. **Make it human-readable.** Through visualizations, interactive tools or just plain explanations, help the audience to understand the model's actions.
4. **Test on edge cases.** Satire, extremely opinionated articles, and clickbait headlines would all be great edge cases that would help expose the blind spots of a model.

REFERENCES

1. **J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova**, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,” *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pp. 4171–4186, 2019.
2. **N. Ruchansky, S. Seo, and Y. Liu**, “CSI: A hybrid deep model for fake news detection,” *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management (CIKM)*, pp. 797–806, 2017.
3. **K. Shu, A. Sliva, S. Wang, J. Tang, and H. Liu**, “Fake News Detection on Social Media: A Data Mining Perspective,” *SIGKDD Explorations*, vol. 19, no. 1, pp. 22–36, 2017.
4. **M. T. Ribeiro, S. Singh, and C. Guestrin**, “"Why should I trust you?": Explaining the predictions of any classifier,” *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1135–1144, 2016.
5. **P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov**, “Enriching Word Vectors with Subword Information,” *Transactions of the Association for Computational Linguistics (TACL)*, vol. 5, pp. 135–146, 2017.
6. **A. Verma et al.**, “A comprehensive WELFake dataset for fake news detection,” *Proceedings of the 2021 IEEE International Conference on Big Data (Big Data)*, pp. 7128–7133, 2021.
7. **Y. Liu et al.**, “RoBERTa: A robustly optimized BERT pretraining approach,” *arXiv preprint*, arXiv:1907.11692, 2019.
8. **Z. Yang et al.**, “XLNet: Generalized Autoregressive Pretraining for

Language Understanding," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 32, 2019.

9. **I. Loshchilov and F. Hutter**, "Decoupled Weight Decay Regularization," *International Conference on Learning Representations (ICLR)*, 2019.
10. **X. Glorot and Y. Bengio**, "Understanding the difficulty of training deep feedforward neural networks," *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 249–256, 2010.
11. **N. Srivastava et al.**, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," *J. Mach. Learn. Res.*, vol. 15, pp. 1929–1958, 2014.
12. **S. Ioffe and C. Szegedy**, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," *Proc. ICML*, pp. 448–456, 2015.
13. **S. Wachter, B. Mittelstadt, and C. Russell**, "Counterfactual Explanations Without Opening the Black Box: Automated Decisions and the GDPR," *Harvard J. Law Technol.*, vol. 31, no. 2, pp. 841–887, 2017.
14. **J. Howard and S. Ruder**, "Universal Language Model Fine-tuning for Text Classification," *Proc. ACL*, pp. 328–339, 2018.
15. **World Health Organization**, "Managing the COVID-19 Infodemic: Promoting Healthy Behaviors and Mitigating the Harm from Misinformation and Disinformation," *WHO Policy Brief*, Sept. 2020.
16. **S. Vosoughi, D. Roy, and S. Aral**, "The spread of true and false news online," *Science*, vol. 359, no. 6380, pp. 1146–1151, 2018.
17. **F. Doshi-Velez and B. Kim**, "Towards A Rigorous Science of Interpretable Machine Learning," *arXiv preprint*, arXiv:1702.08608, 2017.

18. **Z. Zhang, G. Zhang, and J. Han**, “Fake news detection: A survey of state-of-the-art,” *Inf. Fusion*, vol. 79, pp. 260–284, 2022.
19. **T. Mikolov, K. Chen, G. Corrado, and J. Dean**, “Efficient Estimation of Word Representations in Vector Space,” *arXiv preprint*, arXiv:1301.3781, 2013.
20. **D. P. Kingma and J. Ba**, “Adam: A Method for Stochastic Optimization,” *Proc. ICLR*, 2015.