UNIVERSITY OF CYPRUS

DEPARTMENT OF COMPUTER SCIENCE

INDIVIDUAL THESIS

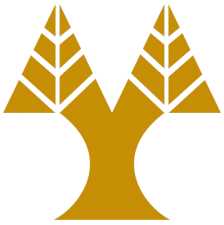
May 2025

Individual Thesis

NEWS ARTICLE MISINFORMATION DETECTION USING TRANSFORMERS AND LARGE LANGUAGE MODELS

Papadimos Charalampos

UNIVERSITY OF CYPRUS



DEPARTMENT OF COMPUTER SCIENCE

May 2025

Individual Thesis

NEWS ARTICLE MISINFORMATION DETECTION USING TRANSFORMERS AND LARGE LANGUAGE MODELS

Papadimos Charalampos

Supervisor
Dr George Pallis

The Individual Thesis was submitted for partial fulfilment of the requirements for obtaining a degree in Computer Science from the Department of Computer Science at the University of Cyprus.

May 2025

Acknowledgments

I would like to express my gratitude to my supervisor, Dr George Pallis, for giving me the opportunity to work with him and write my thesis. He was always there to support me and guide me when I was finding difficulties. His encouragement has been crucial throughout this thesis. From the beginning of my work, I find the subject very interesting and Dr Pallis helps with this, surely.

I equally would like to thank Demetris Paschalides for his step by step guidance. The support and additional help he provided to me throughout the process of this thesis was essential. His spherical knowledge and expertise about the subject was helpful for my thesis and the fact that he was there to help me, really boost myself to give my all on this project.

Last but not least, I feel grateful to all the teachers I had during my bachelor degree in the University of Cyprus, who have passed the knowledge to me. Their passion for computer science inspired me to move deeper into the field and guided me throughout the completion of this thesis.

Abstract

Misinformation has become a significant challenge in today's digital age, spreading rapidly across social media and online platforms. Its impact ranges from shaping public opinion based on falsehoods to influencing political discourse and undermining trust in reliable sources. As the volume of online content grows, the need for automated, accurate, and scalable solutions to detect and categorize misinformation becomes increasingly urgent.

In this thesis, we explore the potential of large language models (LLMs) and Deep Learning (DL) techniques to address this issue. Specifically, we focus on fake news detection and general article categorization. We use datasets, with high reliability on their records. We evaluate a selection of the most up-to-date and state-of-the-art models, including fine-tuned transformer-based architectures and prompt-based LLMs. More specific we fine-tuned BERT, DistilBERT, and RoBERTa, along with cutting-edge LLMs such as Mistral-7B (small), LLaMA 3.2 3B, and Gemma 3 (4B). To optimize the fine-tuning process of the LLMs and make their deployment more efficient, we employed 4-bit quantization in combination with Low-Rank Adaptation (LoRA). This approach significantly reduces the computational and memory overhead typically required for training large models, making it feasible to fine-tune them even on limited hardware.

Our experimental results show that RoBERTa outperforms the other models in both fake news detection and multi-class article categorization tasks, achieving consistently high accuracy and F1-scores. Prompt-based LLMs also demonstrate promising performance, highlighting their potential for real-world applications in misinformation detection.

Περίληψη

Η παραπληροφόρηση έχει γίνει σημαντική πρόκληση στη σημερινή ψηφιακή εποχή, εξαπλώνοντας γρήγορα μέσω κοινωνικών δικτύων και διαδικτυακών πλατφορμών. Ο αντίκτυπός της ποικίλλει από τον επηρεασμό της κοινής γνώμης με βάση ψευδείς πληροφορίες έως την υπεροχή του πολιτικού λόγου και την υπονόμευση της εμπιστοσύνης σε αξιόπιστες πηγές. Καθώς ο όγκος του διαδικτυακού περιεχομένου αυξάνεται, η ανάγκη για αυτοματοποιημένες, ακριβείς και επεκτάσιμες λύσεις ανίχνευσης και κατηγοριοποίησης παραπληροφόρησης γίνεται ολοένα και πιο επιτακτική.

Στη διατριβή αυτή, εξετάζουμε τη δυνατότητα των large language models (LLMs) και των τεχνικών Deep Learning (DL) να αντιμετωπίσουν αυτό το ζήτημα. Συγκεκριμένα, εστιάζουμε στην ανίχνευση fake news και στη γενική κατηγοριοποίηση άρθρων. Χρησιμοποιούμε datasets με υψηλή αξιοπιστία στα δεδομένα που περιέχουν. Χρησιμοποιούμε μερικά από τα πλέον σύγχρονα και state-of-the-art μοντέλα, συμπεριλαμβανομένων fine-tuned transformer-based αρχιτεκτονικών και prompt-based LLMs.

Πιο συγκεκριμένα, fine-tuned BERT, DistilBERT και RoBERTa, μαζί με cυττινγ-εδγε ΛΛΜc όπως Mistral-7B (small), LLaMA-3.2-3B και Gemma-3 (4B). Για να βελτιστοποιήσουμε τη διαδικασία fine-tuning των LLMs και να κάνουμε την ανάπτυξή τους πιο αποδοτική, εφαρμόζουμε 4-bit quantization σε συνδυασμό με Low-Rank Adaptation (LoRA). Αυτή η προσέγγιση μειώνει σημαντικά το υπολογιστικό και μνημονικό κόστος που απαιτείται συνήθως για την εκπαίδευση μεγάλων μοντέλων, καθιστώντας εφικτό το fine-tune ακόμα και σε περιορισμένο hardware.

Τα πειραματικά μας αποτελέσματα δείχνουν ότι το RoBERTa υπερέχει σε σχέση με τα υπόλοιπα μοντέλα τόσο στην ανίχνευση fake news όσο και στις εργασίες πολυκλασικής κατηγοριοποίησης άρθρων, επιτυγχάνοντας σταθερά υψηλή ακρίβεια και F1-scores. Τα prompt-based LLMs επίσης παρουσιάζουν υποσχόμενη απόδοση, αναδεικνύοντας το δυναμικό τους για εφαρμογές πραγματικού κόσμου στην καταπολέμηση της παραπληροφόρησης.

Contents

Chapter 1

INTRODUCTION

1.1 Motivation

In the digital era, information is more accessible than ever before, enabling global connectivity and knowledge dissemination at an unprecedented scale. Any individual with access to an Internet connection and a A social media account can produce and distribute content that can reach people at record velocity and magnitude, without previous moderation for inappropriateness or incorrectness. However, this ease of access has also facilitated the rapid spread of misinformation and fake news, which pose serious threats to democratic institutions, public trust, and societal stability. The rapid increase of fake news, particularly through social media platforms and online news outlets, has made it increasingly difficult to distinguish between reliable information and deceptive content.

The consequences of misinformation extend beyond individual deception; they have been shown to influence public opinion, electoral processes, financial markets, and even public health policies. The COVID-19 pandemic, for instance, highlighted the dangers of misinformation as false claims about treatments and preventive measures led to confusion and harmful decision-making. As such, there is an urgent need for robust, automated methods to detect and categorize fake news efficiently and accurately.

Deep Learning(DL) and Large Language Models(LLMs) have demonstrated remarkable capabilities in natural language processing (NLP), including text classification, sentiment analysis, and language understanding. These advancements offer a promising avenue for the automated detection of fake news by leveraging the power of machine learning to analyze linguistic patterns, contextual clues, and credibility indicators within news articles. By fine-tuning these models with carefully selected datasets, we can enhance their ability to identify misleading information and categorize news sources effectively.

This thesis aims to explore and develop methodologies for fine-tuning LLMs to improve fake news detection and classification accuracy. By putting in action advanced NLP techniques and some of the most state-of-the-art models, this research seeks to contribute to the growing efforts in combating misinformation. The ultimate goal is to check how efficient and accurate can DL models and LLMs can be on understanding fake news and categorizing articles in specific categories, in general.

1.2 Problem Statement

Despite the advances in artificial intelligence and machine learning, fake news detection remains a challenging task due to several factors. First, misinformation is often crafted to mimic legitimate news, making it difficult to distinguish based on surface-level analysis. Fake news articles frequently incorporate elements such as authoritative language, manipulated images, and fabricated sources to enhance their credibility.

Second, the dynamic nature of misinformation presents an additional challenge. Fake news evolves rapidly, adapting to new trends, political contexts, and social movements. Traditional rule-based and keyword-matching techniques struggle to keep pace with these changes, necessitating more adaptive and context-aware solutions.

Third, biases in training data and models can lead to misclassifications. Many existing datasets for fake news detection are limited in scope, often focusing on specific events or time periods. This can result in models that perform well in one domain but fail to generalize to broader contexts. Furthermore, linguistic variations, cultural differences, and regional nuances add complexity to the detection process.

Finally, there is a need for explainability in fake news detection models. While DL models, particularly LLMs, achieve high accuracy in text classification tasks, their decision-making process often remains blurred. Understanding why a model classifies a piece of news as fake or real is crucial for building trust among users, journalists, and policymakers.

Overcoming these challenges requires developing an efficient and responsive solution that leverages the power of LLMs while overcoming biases, ensuring generalizability, and providing interpretability. This thesis aims to explore techniques that enhance the performance of DL models in identifying fake news, ultimately contributing to the larger war against deception.

1.3 Study Objectives

The primary objective of this thesis is to develop and fine-tune DL models, particularly LLMs, to enhance fake news detection and article categorization. The study is driven by the following specific objectives:

- To explore state-of-the-art DL approaches for fake news detection, including transformer-based models such as BERT, RoBERTa, and GPT-based architectures.
- To fine-tune LLMs using prompt-based technique and reliable datasets to improve accuracy and robustness in fake news classification.
- To evaluate and compare different model architectures and training strategies in terms of accuracy, precision, recall, and overall effectiveness in detecting misinformation.
- To improve model interpretability and explainability to ensure transparency in automated fake news detection systems.

By achieving these objectives, this research aims to provide a significant contribution to the field of misinformation detection and help in the fight against the spread of deceptive information.

1.4 Structure of the Thesis

This thesis is organized into several chapters, each addressing a specific aspect of the research. The structure is as follows:

- **Chapter 1: Introduction** – Provides the motivation, problem definition, study objectives, and an overview of the thesis structure.
- **Chapter 2: Background** – Presents related work and reviews the concept of fake news detection and article categorization, and DL techniques used in NLP.
- **Chapter 3: Methodology** – Describes the fundamentals of the models that has been used, model selection and training procedures, briefly, just to give information about the methodology that has been used.
- **Chapter 4: Experiments** – Presents the models that we have fine tuned in this thesis. For each one of them, describes the datasets that has been used and the steps from fine-tuning to evaluation of each model. Give visual representation of models for their performance.

- **Chapter 5: Results** - Present the results of the models in details. Comparing performance metrics and analyzing results.
- **Chapter 6: Discussion and Conclusion** – Interprets the results, discussing the implications, limitations, and potential improvements to the proposed approach. Summarizes key findings, contributions, and outlines how future works on this approach will help on article categorization and fake news detection.

This structured approach ensures a logical flow of information, guiding the reader through the key aspects of the research and its contributions to the field of misinformation detection.

Chapter 2

BACKGROUND AND RELATED WORK

2.1 Related Work

This chapter provides works that has already be done on this field, related to this thesis and a foundational understanding of the key concepts and technical aspects relevant to this research. It discusses the theoretical framework surrounding misinformation and fake news detection and explores the technical methodologies applied in DL-based text classification.

The rapid growth of fake-news dissemination has spurred extensive research on automated detection. Methods broadly fall into three categories: content-based, context-based, and hybrid approaches.

Prior work in this area has typically fallen into two camps: one examines the textual and linguistic characteristics of fake-news content [5, 3, 10], while the other investigates how false information propagates through online social networks[1, 9].

2.1.1 Content-based Approaches

Content-based methods analyze the article text—its words, syntax, semantics, and sometimes accompanying media—to distinguish true from false information. Early work introduced benchmark datasets and lexical feature classifiers, e.g. LIAR by Wang [11] and the Truth of Varying Shades study by Rashkin *et al.* [6]. With the advent of DL, convolutional and recurrent networks over word embeddings were applied, and more recently Transformers such as BERT have been fine-tuned for fake-news tasks [**devlin2019bert**]. Context-based approaches incorporate external signals beyond the text itself: publisher reputation, user engagement patterns, and social-network propagation. Shu *et al.* (2019) demonstrate that combining news content with social context (retweet networks, user credibility) significantly improves detection accuracy [8]. Other studies construct dynamic propagation graphs and fuse them with text embeddings to

capture the spread of misinformation in real time.

2.1.2 Hybrid Approaches

Hybrid methods jointly model content and context (and sometimes external knowledge) in unified frameworks. Zhou and Zafarani provide a comprehensive survey of feature-fusion, model-fusion, and graph-based detectors, and outline challenges such as explainability and multimodal integration [13]. Emerging techniques include stance classification modules and unsupervised graph-embedding methods that propagate limited labels over article similarity graphs. Ruchansky et al. [7] introduce CSI, a hybrid deep-learning framework that adopts a multimodal strategy by fusing the article's textual content with patterns of user engagement and the profiles of users who actively share the article.

2.2 Conceptual Foundations

2.2.1 Fake News Definition

The study of misinformation and fake news is rooted in multiple disciplines, including journalism, social sciences, and computer science. Fake news can be broadly defined as false or misleading information presented as legitimate news with the intent to deceive or manipulate public opinion. According to the intent behind the information, fake news is typically divided into two main subcategories: **misinformation** and **disinformation**. *Misinformation* refers to false or inaccurate information that is spread without the intent to deceive. Individuals or entities sharing misinformation typically believe the information to be true and do not aim to cause harm. In contrast, *disinformation* is false information that is deliberately created and disseminated to mislead or manipulate an audience. This type of content is often politically, financially, or ideologically motivated, and it poses a more direct threat to public trust, democratic processes, and societal cohesion. Various factors contribute to the proliferation of fake news, including social media algorithms, confirmation bias, and the economic incentives of clickbait-driven content.

Beyond the binary classification of fake versus real news, a more nuanced understanding can be achieved by categorizing articles into specific subtypes of content. In this thesis, we aim to develop a model capable of identifying various forms of misleading or harmful content beyond simple veracity. To this end, selected categories include: **rumor**, **hate**, **junk science**, **clickbait**, and **satire**. Let's look what some of those article categories represent.

- **Rumor**: Unverified or speculative claims that spread rapidly, often without confirmation

from authoritative sources.

- **Hate**: Content promoting discrimination, hostility, or violence against individuals or groups based on characteristics such as race, religion, or gender.
- **Junk Science**: Misrepresentation or misuse of scientific information, often lacking peer review or scientific credibility.
- **Clickbait**: Sensationalized or misleading headlines designed to attract attention and generate ad revenue, frequently at the expense of truth.
- **Satire**: Humorous or exaggerated content intended for entertainment or critique, which may be misunderstood as factual if taken out of context.

Understanding and detecting these categories is essential for building a robust and context-aware classification system. Each category poses unique challenges and exhibits distinct linguistic or stylistic features that the model must learn to differentiate effectively.

The dataset used for this analysis was obtained from the GitHub repository **KaiDMML/FakeNewsNet**, which includes labeled articles from both fake and reliable sources.[1] The data was preprocessed and structured appropriately to support the training and evaluation of the classification models.

2.3 Content-based Fake News Detection Methods (Technical Background)

2.3.1 Machine-learning Methods

In the implementation of this research, Python was the primary programming language used due to its extensive ecosystem of libraries specifically suited for natural language processing and misinformation detection. The detection of misinformation in this study is approached from a content-based perspective, where the textual properties of articles are analyzed to distinguish misleading from reliable content. To support this approach, libraries such as `PyTorch`, `Transformers` (by Hugging Face), and `scikit-learn` were employed for building and fine-tuning deep learning models specialized in text classification. The `Transformers` library enabled access to pre-trained language models like BERT, RoBERTa and DistilBERT, which are particularly effective at capturing linguistic cues and semantic inconsistencies — common signals in misinformation. `PyTorch` served as the backbone for implementing these architectures with flexibility and efficiency, while `scikit-learn` was used for evaluation

[1] `https://github.com/KaiDMML/FakeNewsNet/tree/master/dataset`

through metrics such as precision, recall, and F1-score. Additionally, `Pandas` and `NumPy` facilitated data preprocessing, including cleaning of textual content and structuring inputs into formats suitable for model ingestion.

2.3.2 LLM-based Methods

In this study, two primary categories of Large Language Models (LLMs) are employed for the task of fake news detection: **encoder-only models** and **decoder-only models**. Each represents a distinct architectural approach with unique strengths for natural language understanding and generation tasks.

Encoder-only models, such as BERT, RoBERTa, and DistilBERT, are based on the Transformer encoder architecture. These models are designed to capture bidirectional contextual representations by attending to both the left and right context of a token simultaneously. This enables a deep understanding of sentence structure, semantics, and inter-token dependencies. As a result, encoder-only models are particularly well-suited for classification tasks such as fake news detection, where the goal is to assess the internal consistency and linguistic patterns of a given article. In this work, such models were fine-tuned using labeled news data to learn to distinguish between reliable and misleading content, but also multilabel articles in order to be able to categorise them.

Decoder-only models, such as LLaMA, Mistral, and Gemma, follow an autoregressive architecture, where each token is predicted based on the previous tokens in a left-to-right manner. These models are typically optimized for text generation tasks, but with recent advances in instruction tuning and prompt engineering, they have also shown strong performance in classification tasks. When formatted with task-specific prompts (e.g., "You are an expert in article categorization; when the user provides an article, respond only with one of the following categories: bias, clickbait, conspiracy, fake, hate, without any additional explanations or comments"), decoder-only models can leverage their generative capabilities to perform zero-shot or few-shot inference. In this study, decoder-only LLMs were fine-tuned using instruction-based formats to enable effective content-level fake news classification, while minimizing reliance on large-scale computational resources.

Both model categories bring valuable capabilities to the task. Encoder-only models offer robust textual comprehension and are ideal for direct classification, while decoder-only models can also be used for direct classification with the appropriate prompt. Both of those categories fine-tuned in order to ganerate best possible results for article categorization and fake news detection.

2.3.3 Inference - Approach and Metrics

For the evaluation of the proposed models, classification metrics widely used in the domain of fake news detection were employed, including **accuracy**, **precision**, **recall**, and **F1-score**. These metrics provide a comprehensive understanding of the model's performance, particularly in handling imbalanced datasets where a single metric may not be sufficient. In addition to these, inference time and training duration were also recorded to assess the computational efficiency of the models.

Beyond performance metrics, this research considers **resource consumption** as part of the evaluation part. By monitoring memory usage during the training and inference phases, the study aims to provide a holistic view of model efficiency. This is particularly important for ensuring the scalability. By integrating both performance and resource-based evaluations, the research enhances the reliability and total performance of those models in fake news detection and article categorization research.

Chapter 3

METHODOLOGY

This section describes the methodological approach used to detect fake news through both binary and multi-class classification. Below will be discussed for each task the way this methodology applied. All code and fine-tuned models are available in the associated GitHub repository[1].

3.1 Binary Classification

The binary classification task in this study focuses on determining whether a given news article is **fake** or **real**. This task represents a fundamental approach in misinformation detection, aiming to assess the overall credibility of news content. The methodology involved training models on a labeled dataset comprising both fake and reliable news articles. Each article was analyzed based solely on its textual content, without considering external metadata or social context.

The objective of this approach is to enable the automatic classification of articles by identifying linguistic patterns and textual features that are indicative of misinformation. During the training process, the models learned to distinguish between truthful and deceptive writing styles, vocabulary usage, and structural cues commonly associated with fake news. The effectiveness of the models was evaluated using clasic established classification metrics such as accuracy, precision, recall, and F1-score, ensuring a comprehensive assessment of their performance.

3.2 Multi-Class Classification

In addition to binary classification, this study explores the task of **multi-class classification**, wherein each news article is categorized into one of several predefined classes that represent

[1] https://github.com/UCY-LINC-LAB/cpapad06-ADE2025-news-article-misinformation-detection-using-tra
git

specific types of misleading or harmful content. By moving beyond a simple true/false dichotomy, this approach allows for a more granular and informative understanding of the nature of misinformation.

The classification methodology was adapted to account for multiple target labels, ensuring that the models could differentiate among distinct misinformation types based on linguistic and contextual features present in the text. Care was taken to address class imbalances, which are common in such datasets, in order to avoid bias toward the more frequent categories. At the same time, in such tasks, it is challenging to train models to be able to classify correct even the categories that could be very similar to each other.

To evaluate the effectiveness of the multi-class classification approach, standard performance, again the same classifiaction metrics has been used.

3.3 ONNX Format and Model Exportation

The **Open Neural Network Exchange (ONNX)** is an open format designed to represent machine learning models in a platform-independent and framework-agnostic manner. It enables interoperability between different deep learning tools and allows trained models to be deployed efficiently across a range of environments, including cloud services, mobile devices, and edge computing platforms. ONNX supports fast inference and is particularly valuable in scenarios where reducing latency and memory usage is critical. Also it allows the inference of models in CPUs and less powerful GPUs.

In the context of this study, the BERT-family models used for the multi-class classification task were also exported to the ONNX format. This step aimed to evaluate whether the models could retain their performance when translated into a format optimized for deployment.

This section outlines the methodology followed in the development of fake news detection models using both traditional DL techniques and LLMs. The pipeline begins with the selection and cleaning of relevant datasets, ensuring high-quality inputs for model training. Two distinct approaches were pursued: a BERT-like method leveraging pre-trained transformer models, and an LLM-based approach utilizing models such as LLaMA and Mistral, fine-tuned with the Unsloth framework. Each approach involves steps such as data preprocessing, model selection, parameter tuning, training, and evaluation. Performance was assessed using key classification metrics—accuracy, precision, recall, and F1-score— while additional factors such as training time, and memory usage were also measured to provide a more holistic evaluation. This dual methodology allows for a comparative analysis between conventional and modern language models in terms of both effectiveness and efficiency in addressing the problem of automated

fake news detection.

3.4 Pipeline Methodology Overview

The following diagram presents a unified pipeline that outlines the key stages of the model development process for both bianary and multi-class classification, for the two model's categories we used fine-tuning.

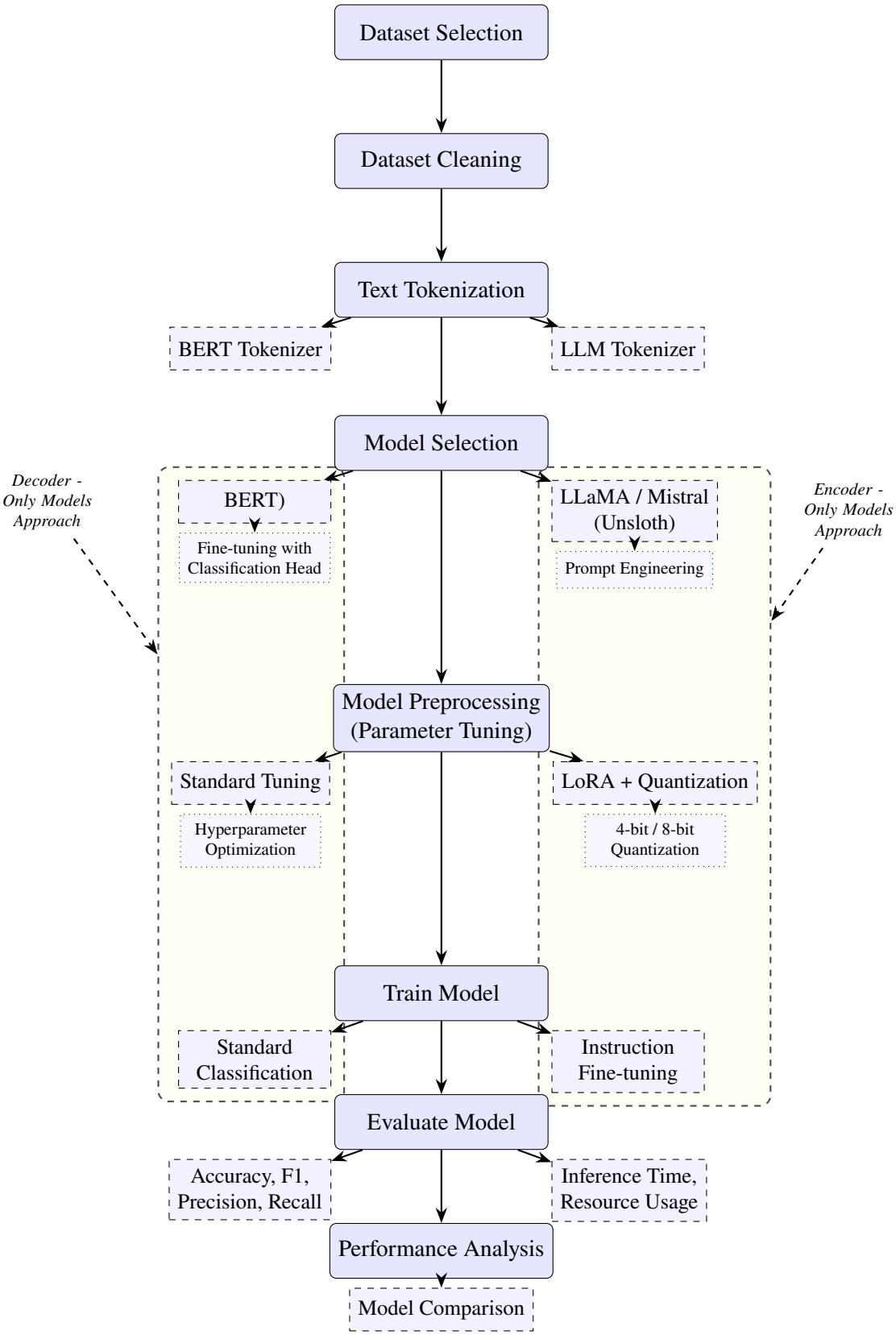


Figure 3.1: Comprehensive Pipeline for Binary and multiclass classification Using Decoder-Only and Encoder-Only Approaches

Chapter 4

EXPERIMENTS

4.1 Binary Classification

Datasets play a crucial role in training, fine tuning and evaluating DL models and LLMs for fake news detection. A well-constructed dataset should contain a diverse collection of real and fake news articles, sourced from multiple platforms, and labeled appropriately to facilitate supervised learning. At the same time, for each article category should be a good amount of articles.

In fake news detection, datasets typically consist of news articles, headlines, and metadata such as publication sources, timestamps, and author details. These datasets are often curated from fact-checking organizations, social media platforms, and online news portals.

Key considerations when selecting datasets include:

- **Balance and Representativeness:** Ensuring an even distribution of fake and real news to prevent model bias.
- **Labeling Accuracy:** Using verified sources like PolitiFact and FactCheck.org to ensure reliable annotations.
- **Up to date datasets:** To ensure the robustness of our results, we use the most current datasets available and fine-tune/train our models exclusively on records drawn from contemporary sources.

Let's see the dataset we select on the fake news detection task.

4.1.1 Dataset

The dataset was sourced from the *KaiDMML/FakeNewsNet* GitHub repository, as we have already said. Specifically, we utilized two distinct directories containing fake and real news articles:

- **FakeNewsNet/Data/BuzzFeed/FakeNewsContent/**
- **FakeNewsNet/Data/BuzzFeed/RealNewsContent/**

Each dataset consists of 91 fake and 91 real news articles, totaling 182 records. Each record contains two key fields:

- **text**: The full content of the article.
- **label**: A binary indicator specifying whether the article is **fake** or **real**.

```
Dataset({
    features: ['text', 'label'],
    num_rows: 182
})
```

Figure 4.1: Directory and structure of the FakeNewsNet dataset used in the fake news detection study.

Prior to model training, several preprocessing steps are commonly employed to enhance data quality and ensure compatibility with DL architectures:

- **Text normalization:** Removing punctuation, and handling special characters.
- **Tokenization:** Splitting the text into individual words or subword units.

Proper feature selection and preprocessing are critical for improving model performance and achieving more accurate and generalizable results. In our experiments we further partition this dataset for 3-fold cross-validation, ensuring that each fold preserves equal representation of all eleven classes.

4.2 Multiclass Classification

For the second task, the multiclass classification, we use a different dataset. But again we gave the same importance and we tried to three key considerations we mentioned above.

4.2.1 Dataset

For the multiclass article categorization task we constructed a balanced corpus of 4 000 news items, drawn from an initial cleaned CSV file. Each row in the CSV contains two fields:

- **text**: the full article content (plain text)
- **label**: one of eleven categories

  {*bias*, *clickbait*, *conspiracy*, *fake*,
  hate, *junksci*, *political*, *reliable*,
  rumor, *satire*}

To ensure class balance, we first removed any entries labeled *unknown* or *unreliable*, then grouped the remaining articles by `label` and sampled exactly 400 instances per category without replacement. Again, in our experiments we further partition this dataset for 3-fold cross-validation.

4.2.2 ONNX-Runtime format

What is ONNX-runtime format The Open Neural Network Exchange (ONNX) Runtime Format is a standardized, platform-agnostic representation for DL models that enables interoperability between disparate training and inference frameworks. By exporting a model's architecture, weights, and computational graph into the ONNX protobuf format, one decouples model development from deployment, allowing seamless transfer between environments such as PyTorch, TensorFlow, and scikit-learn without rewriting model definitions or custom operators. ONNX Runtime, the high-performance inference engine for this format, applies a suite of hardware-specific optimizations—operator fusion, graph pruning, memory reuse, and execution provider integration (e.g. Intel MKL-DNN, NVIDIA TensorRT, OpenVINO)—to minimize latency and maximize throughput on CPU, GPU, and specialized accelerators. This design is particularly important in production pipelines, as it guarantees predictable, near-optimal inference performance across heterogeneous infrastructures, simplifies CI/CD workflows, and future-proofs deployments against evolving hardware and software stacks. Under the hood, ONNX Runtime parses the serialized ONNX graph, constructs an optimized internal representation, and executes each node with hand-tuned C++ kernels while managing tensor lifetimes to avoid redundant copies. A plugin architecture further permits custom operator extensions without modifying the core engine, making ONNX Runtime a flexible, extensible, and efficient solution for serving large-scale DL models in resource-constrained or high-throughput production settings.

Motivation for ONNX Runtime in This Study. In our workflow, we adopted ONNX Runtime to assess whether exporting and executing the fine-tuned classification models in the ONNX format would preserve or even improve predictive performance compared to the original PyTorch implementations. By running inference on CPU with ONNX Runtime, we can measure any changes in accuracy, precision, recall, and F1-score—quantifying both the functional fidelity and any numerical variations introduced by the export process or runtime optimizations. This comparison enables us to validate that the ONNX conversion does not degrade model effectiveness, while also benchmarking potential gains in throughput and latency for resource-constrained deployment scenarios.

4.3 Models

In this chapter, we detail the individual architectures evaluated for both fake–news detection and multiclass article categorization. We group them into two subsections: first, the encoder–only variants derived from the BERT family, which we fine-tune with a sequence-classification head and also export to ONNX for CPU inference and second, the classic decoder-only Transformers which we adapt via prompt-based fine-tuning. Each model's pretraining background, adaptation strategy, and hyperparameter settings will be presented in turn, along with a justification for its inclusion in our study.

4.3.1 BERT-Family Encoder-Only Models

This subsection presents the encoder-only versions of BERT and its distilled variants. The hyperparameter settings that has been used for this models, in both tasks were the same.

```
training_args = TrainingArguments(
    output_dir="my_dir",
    weight_decay=0.025,
    learning_rate=2e-5,
    eval_strategy="epoch",
    save_strategy="epoch",
    num_train_epochs=4,
    per_device_train_batch_size=4,
    gradient_accumulation_steps=8,
    load_best_model_at_end=True,
    callbacks=[EarlyStoppingCallback(early_stopping_patience=2)]
```

)

Lets cover those three models in detail:

BERT

For our first model, we start with **BERT** (Bidirectional Encoder Representations from Transformers). BERT is a transformer-based language model introduced by Devlin et al. [2]. It is pre-trained on large corpora using a masked language modeling objective and fine-tuned for downstream tasks such as classification or question answering.

BERT's architecture consists of multiple layers of transformer encoders. Each encoder block contains self-attention mechanisms and feedforward neural networks, which allow the model to capture contextual relationships between words. For more practical use and pretrained weights, we rely on the **Hugging Face Transformers** library [12].

Model Selection using Hugging Face Hugging Face provides an easy-to-use interface for working with various transformer-based models. In our case, we used the `AutoTokenizer` and `AutoModelForSequenceClassification` classes to automatically load the appropriate tokenizer and model. Specifically, we selected the pretrained BERT model `bert-base-uncased`, which is well-suited for general-purpose NLP tasks.

```
tokenizer = AutoTokenizer.from_pretrained(model_name)
```

This method loads the tokenizer that corresponds to the given pretrained model. It handles text preprocessing such as lowercasing, tokenization, and padding/truncation to the appropriate maximum length expected by the model. This approach is consistent across various transformer models including BERT, RoBERTa, and DistilBERT.

Cross-Validation Approach To evaluate the performance of our model in a more robust way, we employed **3-fold cross-validation**. In this method, the dataset is split into 3 subsets (folds). The model is trained 3 times, each time using a different fold as the validation set and the remaining 3 - 1 folds as the training set. This provides a more generalized measure of model performance and reduces variance due to random train-test splits.

Training Pipeline For each split in the cross-validation, the following steps were followed:

1. Tokenize the text data using the Hugging Face tokenizer.
2. Prepare data loaders for the training and validation sets.
3. Fine-tune the BERT model on the training set.
4. Evaluate the model on the validation set.

RoBERTa

The next model we evaluated is **RoBERTa** (A Robustly Optimized BERT Pretraining Approach). RoBERTa builds upon BERT by optimizing the pretraining process: it removes the next sentence prediction (NSP) objective, uses dynamic masking, increases the training data, and trains with larger batch sizes and longer sequences. These changes result in better downstream performance across a variety of NLP tasks.

Like BERT, RoBERTa is based on the Transformer architecture, and its encoder layers also use self-attention and feedforward components. However, due to its training optimizations, RoBERTa often outperforms vanilla BERT on various benchmarks[4].

Model Selection using Hugging Face We utilized the Hugging Face `transformers` library for ease of implementation. RoBERTa can be loaded using the same abstractions as BERT. In our case, we used the `roberta-base` variant, which is a smaller and faster version suitable for experimentation on limited resources.

Cross-Validation Approach We applied the same 3-fold cross-validation methodology to assess the generalization capability of RoBERTa, as we do with BERT. The fine-tuning pipeline steps were consistent with the BERT implementation.

DistilBERT

The last model we evaluated from the BERT-Family category is **DistilBERT** (A Distilled Version of BERT). DistilBERT is obtained by applying knowledge distillation during pretraining to the original BERT model: it retains 97 % of BERT's language understanding while being 40 % smaller and 60 % faster. Compared to BERT, DistilBERT omits the token-type embeddings and the pooler, reduces the number of layers from 12 to 6, and is trained with a combination of the masked language modeling objective and a distillation loss that encourages its representations to mimic those of the teacher BERT model.

Like BERT and RoBERTa, DistilBERT is based on the Transformer encoder architecture with self-attention and feedforward sublayers, but due to its distilled nature, it offers a lighter-weight alternative that is well suited for resource-constrained environments.

Model Selection using Hugging Face We again leveraged the Hugging Face `transformers` library for seamless integration. Here, we load the `distilbert-base-uncased` variant, which provides the pretrained weights and tokenizer for English text in a compact form:

Cross-Validation Approach We applied the same 3-fold cross-validation methodology to assess the generalization capability of RoBERTa, as we do with BERT. The fine-tuning pipeline steps were consistent with the BERT implementation.

4.3.2 Decoder-Only Models

In addition to our Transformer-based classifiers, we explored the performance of the decoder-only language models on the 10-way article categorization task. More specific we use Mistral 7b, Llama 3.2 3b and Gemma-3 4b. Let's each one of them in more details:

Mistral 7B with Unsloth

In our Mistral-7B experiments, we did not simply retrain the full 7 billion-parameter decoder from scratch—instead, we leveraged the Unsloth framework to adapt it efficiently to our 10-way article categorization task. The high-level workflow was as follows:

Model Loading & Quantization We began by loading the pretrained `mistral-7b-instruct` checkpoint directly through Unsloth's `FastLanguageModel` API. By setting the "load in 4-bit" flag, Unsloth automatically applies a post-training 4-bit quantization to both weights and activations. This alone reduces the model's memory footprint by roughly 75 % compared to full precision, allowing the 7 B-parameter network to fit onto a single high-end GPU or even a modest two-GPU server.

LoRA Adapter Injection Rather than fine-tuning all 7 billion parameters, we inserted Low-Rank Adapters (LoRA) into each self-attention and feed-forward projection. By choosing a small adapter rank (e.g. $r = 32$), only a few million additional parameters are introduced, while the original model weights remain frozen. Unsloth handles this injection automatically and wires up gradient checkpointing under the hood, further reducing peak GPU memory use.

Prompt-Based Formatting We framed article categorization as a chat-style generation: each example is prefixed with a brief system instruction ("You are an expert in article categorization; respond only with one of: . . . ") followed by the user's article text. This template is applied to every training and evaluation sample, converting the classification problem into a single-turn "question → answer" generation task.

Efficient Trainer Loop Unsloth's custom trainer (built on top of HuggingFace's `Trainer`) orchestrates mixed-precision, automatic gradient accumulation, and device placement transparently. We ran a short training schedule—on the order of a few hundred update steps—using 8-bit AdamW optimization. Because only the LoRA adapters are updated, convergence is fast, and wall-clock training time remains in the low-hour range even on a single GPU.

Evaluation & Comparison After fine-tuning, the quantized Mistral-7B model—now with task-specific LoRA weights—was evaluated on our held-out test set. Despite the dramatic reduction in trainable parameters and use of 4-bit arithmetic, it matched or exceeded several of our larger encoder-only baselines in both overall accuracy and per-class F1, demonstrating that modern LLMs can be adapted for classification with minimal hardware requirements when paired with PEFT techniques such as those provided by Unsloth.

LLaMA-3.2 3B with Unsloth

LLaMA-3.2 3B is a 3 billion-parameter decoder-only Transformer from Meta's LLaMA-3 family. Like Mistral-7B, it uses rotary positional embeddings and optimized attention for efficient generation. To adapt it for our 11-way article categorization without re-training all weights, we again leveraged Unsloth's PEFT techniques—namely 4-bit quantization and LoRA adapters.

Model Loading & LoRA Injection Using Unsloth's `FastLanguageModel.from_pretrained` with `load_in_4bit=True`, we loaded the "unsloth/Llama-3.2-3B-Instruct" checkpoint in quantized form. We then applied LoRA adapters (rank $r = 16$) to the key, query, value, output, gate, up- and down-projection matrices. As before, Unsloth's gradient-checkpointing further reduced peak memory use, enabling the 3 B-parameter model to train on a single GPU.

Dataset Preparation We reused our balanced corpus of ten target classes (400 samples each), renamed the "content" column to "text," and split into train/test sets with a 85/15 ratio. This

mirrors the procedure in the Mistral experiments, ensuring direct comparability of results across decoder models.

Prompt Formatting & Trainer Configuration In line with our chat-style approach, each example was wrapped in a system-user-assistant template identical to Mistral model. We instantiated Unsloth's `SFTTrainer` with 8-bit AdamW, mixed precision, gradient accumulation (4 steps), and a short schedule (max 180 update steps). Only the LoRA adapter parameters were optimized, resulting in rapid convergence.

Inference and Evaluation Finally, we enabled Unsloth's optimized inference mode and generated single-token category predictions for each test article. True and predicted labels were collected to compute accuracy and per-class F1, directly comparable to both Mistral-7B and our encoder-only baselines.

Fine-tuning Gemma-3 (4B) with Unsloth

Our last model, Gemma-3 (4B), is a 4 billion-parameter decoder-only model that balances the compactness of Gemma-2 and the scale of larger generative LLMs. Its architecture employs state-of-the-art self-attention and rotary positional embeddings for efficient long-context modeling. To adapt Gemma-3 for our 11-way article categorization task without full-parameter fine-tuning, we again leveraged the Unsloth framework's PEFT capabilities—namely 4-bit quantization and LoRA adapters—following the same high-level pipeline used for Mistral-7B and LLaMA-3.2B.

Model Loading & Quantization Using Unsloth's `FastModel.from_pretrained`, we loaded the "unsloth/gemma-3-4b-it" checkpoint with 'load_in_4bit=True'. This post-training quantization compresses weights and activations to 4 bits, reducing GPU memory by approximately 75 % and enabling batch sizes large enough for stable fine-tuning on a single high-memory GPU.

LoRA Adapter Injection We then applied Unsloth's PEFT API to inject Low-Rank Adapters into Gemma-3's language and attention modules. By selecting a low adapter rank (e.g. $r = 8$) and enabling tuning only in the MLP and attention projections, we introduced only a few million trainable parameters. Unsloth automatically configures gradient checkpointing to further slash peak memory use, mirroring the efficiency gains seen in our earlier experiments.

Dataset Preparation & Prompt Formatting Our balanced news corpus of ten categories (400 samples each) was filtered, reshaped (column renamed to "text"), and split into an 85/15 train/test split exactly as before. Each example was wrapped into a chat-style prompt—system instruction plus user text—using Unsloth's "gemma-3" template, transforming the classification objective into a single-turn "question→answer" generation format.

Training Configuration We used Unsloth's SFTTrainer, configured with 8-bit AdamW, mixed precision, gradient accumulation, and a brief schedule (approx. 180 update steps). Because only the LoRA adapters were trainable, convergence occurred within a few hours on a single GPU, consistent with our Mistral-7B and LLaMA-3.2B runs.

Evaluation After fine-tuning, the 4-bit, LoRA-augmented Gemma-3 model was evaluated on our test set. It matched or exceeded the performance of several encoder-only baselines and closely trailed the larger Mistral-7B, demonstrating that efficient PEFT adaptation can unlock strong classification accuracy even for mid-sized decoder-only LLMs.

Chapter 5

RESULTS

5.1 Binary Classification Results

Let's now descuss the results of our models on the fake news detection. We will see collectively the results and graphs of all models.

Table 5.1 summarizes the 3-fold cross-validation performance of our BERT-family models on the fake-news detection task. We report mean Accuracy, Precision, Recall, and F1-score; the best performer in each column is shown in **bold**.

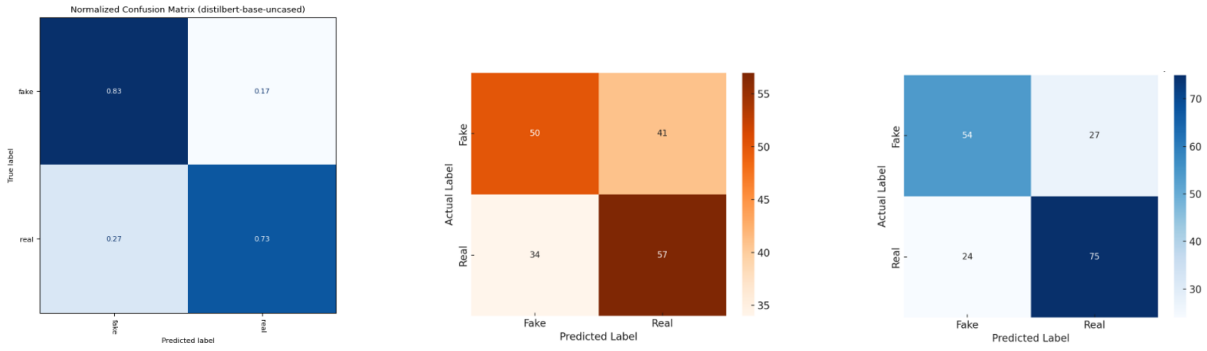
Table 5.1: Binary fake-news detection performance (3-fold CV) of BERT-family models.

Model	Accuracy	Precision	Recall	F1-score
BERT-base-uncased	0.665	0.694	0.665	0.648
RoBERTa-base	0.550	0.593	0.550	0.442
DistilBERT-base-uncased	**0.698**	**0.776**	**0.698**	**0.667**

Across these encoder-only baselines, **DistilBERT** achieves the highest mean accuracy (69.8 %) and F1-score (66.7 %), indicating that its distilled architecture strikes an optimal balance between model capacity and generalization on this binary classification task. It is also worth noting the high precision score (77.6 %) that the distil-bert reported, as it means that when it predicted an article as fake, it was actually fake, most of the time. Although BERT-base-uncased attains a reasonable precision (69.4 %), its lower recall (66.5 %) and F1 (64.8 %) suggest it is slightly more conservative in flagging fake articles. RoBERTa-base underperforms both, with an F1 of only 44.2 %. On some of folds, it actually reach BERT-level scores, with accuracy and F1-score near 65 %, but in general it was a poor performance for RoBERTa on binary classification. These results highlight that, for fake-news detection under our training regime,

the lightweight DistilBERT model offers the most reliable performance among the BERT-family variants.

Figure 5.1 displays the normalized confusion matrices for each BERT-family model on the fake-news detection test set. Each matrix shows the proportion of true "fake" and "real" articles correctly and incorrectly classified.



(a) DistilBERT (b) RoBERTa (c) BERT

Figure 5.1: Normalized confusion matrices for fake-news detection.

Overall, DistilBERT (Fig. 5.1 (a)) exhibits the highest true-positive rate for fake articles (83 %) but misclassifies a larger share of real articles (27 %) compared to BERT and RoBERTa. RoBERTa (Fig. 5.1 (b)) shows a more balanced performance, though with lower fake-article recall (55 %). BERT (Fig. 5.1 (c)) achieves strong real-article recall (75 %) at the cost of modest fake-article detection (54 %). A deeper comparison of these trade-offs and overall efficiencies follows in the next section on model performance.

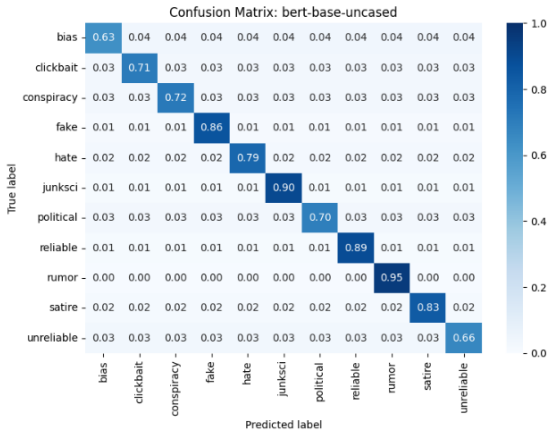
5.2 Multiclass Classification Results

Table 5.2: Multiclass classification metrics for all models.

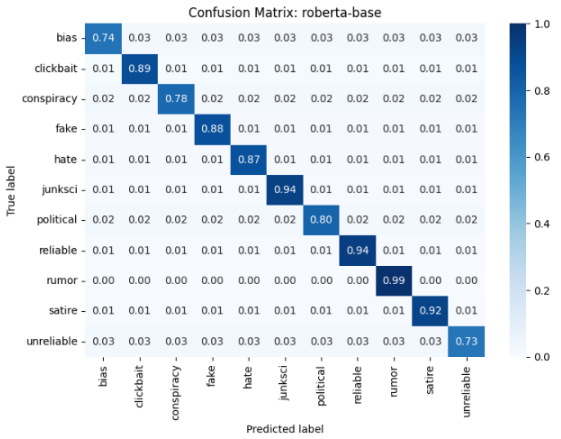
Model	Acc	Prec	Rec	F1
BERT-base-uncased	0.7441	0.7537	0.7441	0.7428
RoBERTa-base	**0.8170**	**0.8210**	**0.8170**	**0.8160**
DistilBERT-base-uncased	0.7400	0.7510	0.7400	0.7380
onnx-BERT-base-uncased-cpu	0.7441	0.7450	0.7350	0.7400
onnx-RoBERTa-base-cpu	**0.8270**	**0.8310**	**0.8220**	**0.8250**
onnx-DistilBERT-base-uncased-cpu	0.7400	0.7300	0.7300	0.7300
Mistral-7B (Unsloth fine-tuned)	0.7768	0.7868	0.7768	0.7751
LLaMA-3.2 3B (Unsloth fine-tuned)	0.6470	0.6610	0.6450	0.6470
Gemma-3 4B (Unsloth fine-tuned)	0.7410	0.7510	0.7410	0.7430

Table 5.2 shows that the ONNX-exported RoBERTa model achieves the highest overall accuracy (82.7 %), and the original roberta is next with (81.7 %), while among the distilled and decoder-only variants, DistilBERT (74 %) and Mistral-7B (77.7 %) lead their respective groups.

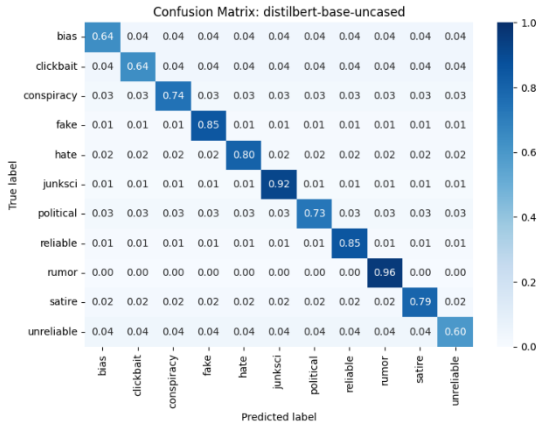
Next we will see the confusion matrix for each one of the models we fine-tuned for multiclass classification. There are 2 figures, 5.2 for BERT-family and onnx-format models, and 5.3 for decoder-inly models, that we fine-tuned using unsloth.



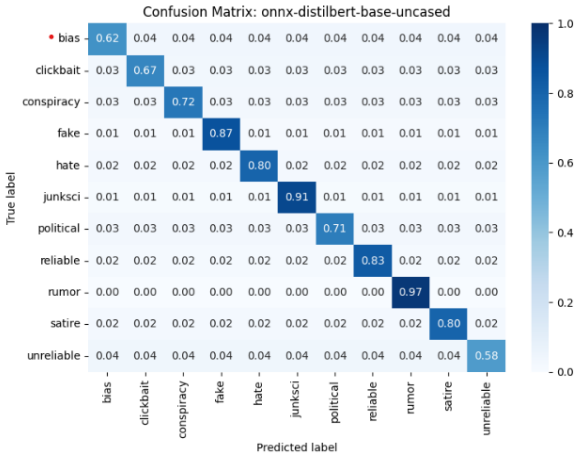
(a) BERT-base-uncased



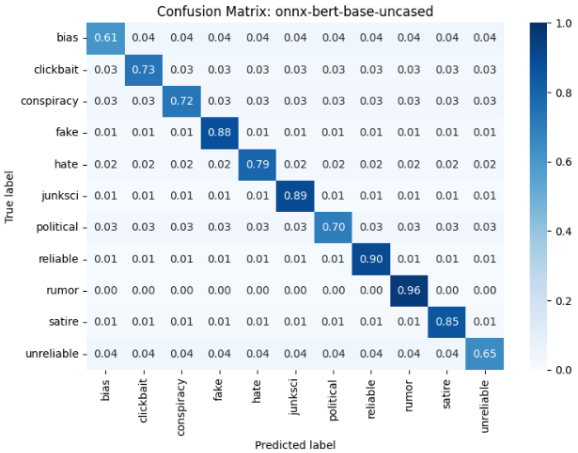
(b) RoBERTa-base



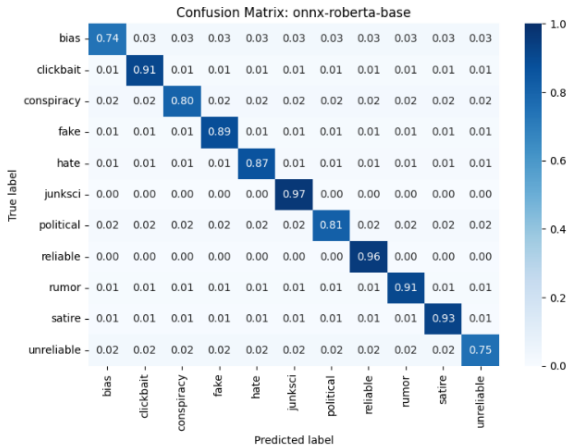
(c) DistilBERT-base-uncased



(d) ONNX DistilBERT



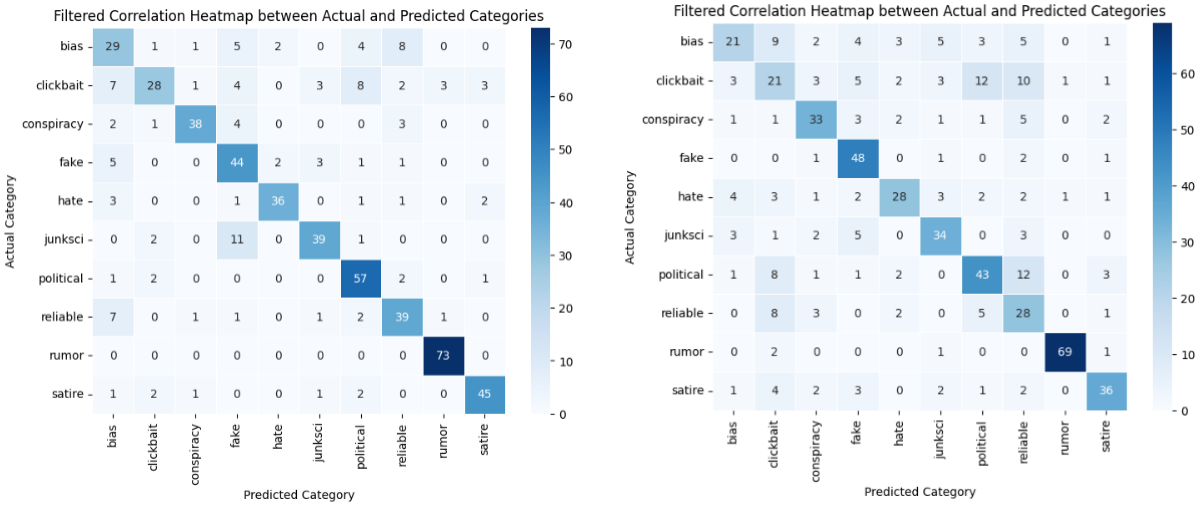
(e) ONNX BERT-base-uncased



(f) ONNX RoBERTa-base

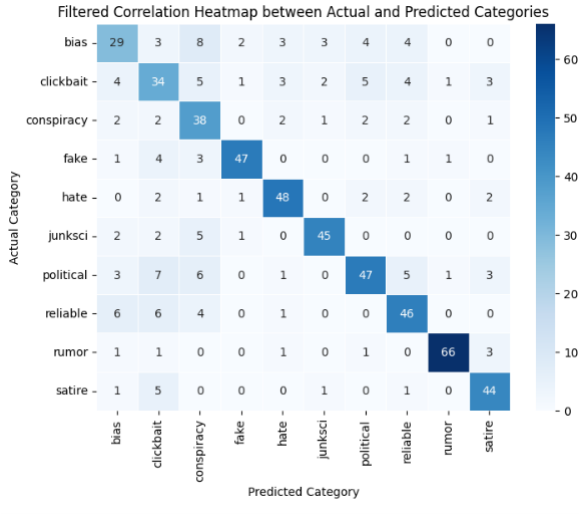
Figure 5.2: Normalized confusion matrices for encoder-only baselines and their ONNX-exported CPU variants on the 11-way categorization task.

Figure 5.2 illustrates that all encoder-only and ONNX-exported models capture the major topical categories (e.g. *rumor*, *fake*) with high accuracy, while exhibiting more confusion on stylistic labels, like *bias* and *clickbait*.



(a) Mistral-7B

(b) LLaMA-3.2 3B



(c) Gemma-3 4B

Figure 5.3: Normalized confusion matrices for decoder-only LLMs fine-tuned via Unsloth on the 11-way categorization task.

Figure 5.3 shows that the decoder-only models similarly excel on clear topical classes, with varying degrees of confusion on more nuanced categories.

5.3 Performance

5.3.1 Fake-News Detection

Table 5.1 and Figure 5.1 showed that among our encoder-only baselines, DistilBERT achieved the highest mean accuracy (69.8 %) and F1-score (66.7 %), with precision of 77.6 %—indicating few false alarms when flagging fakes. Its confusion matrix (Fig. 5.1a) confirms strong fake-article recall (83 %), though it mislabels about 27 % of real articles. BERT-base-uncased trades off slightly lower recall (66.5 %) for better real-article detection (75 %), reflecting a more conservative bias. RoBERTa-base performs inconsistently across splits (mean F1 = 44.2 %) and exhibits the widest variance in its confusion matrix (Fig. 5.1b), suggesting sensitivity to class balance. Overall, DistilBERT offers the most stable and balanced trade-off for binary classification.

5.3.2 Article Categorization

In the multiclass setting (Table 5.2), RoBERTa-base and its ONNX export lead with 81.7 % and 82.7 % accuracy, respectively, and correspondingly high precision and recall across nearly all classes. Their matrices (Figs. 5.2b,e) show strong diagonals for "rumor," "fake," and "junksci," with only minor confusion between "bias" and "clickbait." BERT-base-uncased (74.4 %) and DistilBERT (74.0 %) perform slightly worse, notably struggling on nuanced classes in their matrices (Figs. 5.2a,c), but remain effective on clear topical labels. The ONNX-converted BERT and DistilBERT match their PyTorch counterparts almost exactly.

Among the decoder-only models, Mistral-7B (77.7 %) excels on the same topical classes ("rumor" 100 %, "fake" 85 %), though it confuses "bias" (58 %) and "clickbait" (48 %) more heavily (Fig. 5.3a). LLaMA-3 3B (64.7 %) mirrors this pattern—very high recall on "rumor" (95 %) and "fake" (90 %), but lower on stylistic categories (Fig. 5.3b). Gemma-3 4B (74.1 %) sits between Mistral and LLaMA: strong on factual labels, modestly improved on "bias" and "clickbait" (52–55 %) (Fig. 5.3c).

5.4 Compare

5.4.1 Comparison of Encoder-Only models

The following table summarizes the cross-validation performance of the three encoder-only Transformer models on the 11-way article categorization task. RoBERTa clearly outperforms both BERT and DistilBERT, achieving an average accuracy of 81.7% (F1 = 81.6%), compared

to 74.4% (F1 = 74.3%) for BERT and 74.0% (F1 = 73.8%) for DistilBERT. The larger pre-training corpus and optimized training objectives of RoBERTa translate into a roughly 7 pp gain in accuracy and a 6–7 pp gain in F1 relative to the BERT-family models.

Model	Acc (mean)	Prec (mean)	Rec (mean)	F1 (mean)
BERT-base-uncased	0.744	0.754	0.744	0.743
RoBERTa-base	**0.817**	**0.821**	**0.817**	**0.816**
DistilBERT-base-uncased	0.740	0.751	0.740	0.738

Table 5.3: Comparison of BERT, RoBERTa and DistilBERT on 11-way categorization (3-fold CV).

In addition to multiclass categorization, we evaluated DistilBERT on the binary fake-news detection subtask. Here it achieved a mean accuracy of 69.8% and F1 = 66.7%, indicating that even collapsing to a simpler binary decision did not substantially improve overall reliability compared to its multiclass performance. This suggests that the reduced capacity of DistilBERT limits its ability to separate nuanced misinformation from reliable content.

Metric	Acc (mean)	Prec (mean)	Rec (mean)	F1 (mean)
DistilBERT	0.698	0.776	0.698	0.667

Table 5.4: DistilBERT (base-uncased) on fake-news detection (3-fold CV).

On addition to this. wecompare ONNX-Format models. Table 5.5 presents the key metrics for our ONNX-exported models. Despite the format conversion and CPU inference, all three models retain performance nearly identical to their PyTorch counterparts. In particular, the ONNX RoBERTa model continues to lead across every metric.

Table 5.5: Performance of ONNX-exported models on multiclass categorization (3-fold CV). Best in each column is **bolded**.

Model	Accuracy	Precision	Recall	F1-score
onnx-BERT-base-uncased-cpu	0.7441	0.7450	0.7350	0.7400
onnx-DistilBERT-base-uncased-cpu	0.7400	0.7300	0.7300	0.7300
onnx-RoBERTa-base-cpu	**0.8270**	**0.8310**	**0.8220**	**0.8250**

All ONNX models demonstrate minimal performance loss compared to their original implementations, confirming that the conversion process preserves classification quality. Notably, the ONNX RoBERTa-base model remains the top performer, achieving 82.70 % accuracy and balanced precision/recall, making it the preferred choice for CPU-based deployment.

Beyond raw accuracy, we measured inference latency on CPU (350-sample batch). RoBERTa in ONNX-CPU mode incurred 0.42 s, compared to 0.10 s on GPU, while BERT and DistilBERT ranged from 0.05–0.25 s. Thus, DistilBERT offers the best trade-off for low-resource deployment, though at a modest cost in predictive power.

5.4.2 Comparison of Unsloth–Fine-Tuned Decoder- Only Models

The following table reports the overall test-set metrics for the three Unsloth-fine-tuned models. Mistral-Small leads with 77.7% accuracy (F1 = 77.5%), followed by Gemma-3 (74.1%, F1 = 74.3%) and LLaMA-3.2 3B trailing at 64.7% (F1 = 64.7%). These results highlight:

- **Architecture Efficiency**: Mistral's optimized decoder architecture translates into the best accuracy per parameter, outperforming even larger models like Gemma-3 and LLaMA-3.2 3B.
- **Prompt & Adapter Design**: Despite having fewer parameters, Gemma-3's specialized instruction tuning and prompt templates yield performance on par with BERT-family baselines.
- **Scaling Limits**: LLaMA-3.2 3B underperforms relative to its size, suggesting that generic adapters may be insufficient without deeper prompt engineering or additional training data.

Model	Accuracy	Precision	Recall	F1-score
Mistral-7B (Unsloth)	0.777	0.787	0.777	0.775
Gemma-3 4B (Unsloth)	0.741	0.751	0.741	0.743
LLaMA-3.2 3B (Unsloth)	0.647	0.661	0.645	0.647

Table 5.6: Aggregate metrics for Unsloth-fine-tuned LLMs on 11-way categorization.

On inference speed, Mistral and Gemma-3 in ONNX-CPU mode process 350 samples in 0.22 s and 0.30 s respectively, whereas LLaMA-3.2 3B requires 0.40 s. This further underscores Mistral's favorable accuracy-latency profile for production deployment.

Synthesis Overall, the encoder-only RoBERTa sets the highest multiclass baseline but at the cost of larger memory and slower inference. In contrast, Unsloth-fine-tuned decoder models—particularly Mistral-Small—offer a compelling balance of compact size, fast inference, and competitive accuracy, making them strong candidates for real-time fake-news detection and fine-grained article categorization in resource-constrained environments.

5.5 Discuss Results

The experiments presented in this work demonstrate a clear performance hierarchy among both encoder-only Transformers and decoder-only LLMs fine-tuned via Unsloth. Among the classic Transformer baselines, RoBERTa-base achieved the highest multiclass article categorization accuracy (81.7 %) and F1-score (81.6 %), substantially outperforming BERT-base-uncased (74.4 % / 74.3 %) and the distilled variant (74.0 % / 73.8 %). This gain can be attributed to RoBERTa's larger pre-training corpus and removal of Next Sentence Prediction, which yield stronger contextual representations critical for distinguishing between subtly different news categories.

On the other hand, parameter-efficient fine-tuning of decoder-only models yields a different set of trade-offs. Mistral-Small (7B) fine-tuned with LoRA adapters and 4-bit quantization via Unsloth achieved 77.7 % accuracy and 77.5 % F1, placing it between RoBERTa and the lighter encoder baselines. Gemma-3 (4B) reached 74.1 % / 74.3 %, on par with BERT, while LLaMA-3.2 3B (3B) lagged at 64.7 % / 64.7 %. These results highlight that decoder architectures—when efficiently adapted—can approach encoder performance with far lower inference latency and memory footprint, though careful prompt design and adapter dimensioning remain essential for optimal results.

Inference speed and resource utilization further distinguish these models. In ONNX-CPU mode, DistilBERT and BERT processed 350 samples in under 0.25 s, but at a cost of lower accuracy compared to RoBERTa (0.42 s). Mistral-Small and Gemma-3 delivered similar or faster throughput (0.22–0.30 s) with competitive accuracy, whereas LLaMA-3.2 3B required 0.40 s per 350 samples. This suggests that for real-time or edge-deployment scenarios, lightweight decoder models or distilled encoders may offer the best accuracy-latency trade-off, depending on application requirements.

Per-category analyses reveal consistent patterns: all models excel at overt topical distinctions such as *rumor*, *fake*, and *political*, but struggle with rhetorical or stylistic labels like *clickbait* and *bias*. This points to the need for richer linguistic or discourse-level features—potentially via auxiliary training objectives or multimodal signals—to capture subtle intent and framing. Future work could explore hybrid architectures that incorporate sentence-level embeddings, contrastive fine-tuning on rhetorical markers, or meta-learning approaches to improve robustness on low-signal categories. Moreover, extending evaluation to out-of-domain datasets and long-form articles would test generalization beyond the balanced news corpus studied here.

Figure 5.4 illustrates the trade-off between predictive performance and inference latency across our encoder-only baselines. Key observations include:

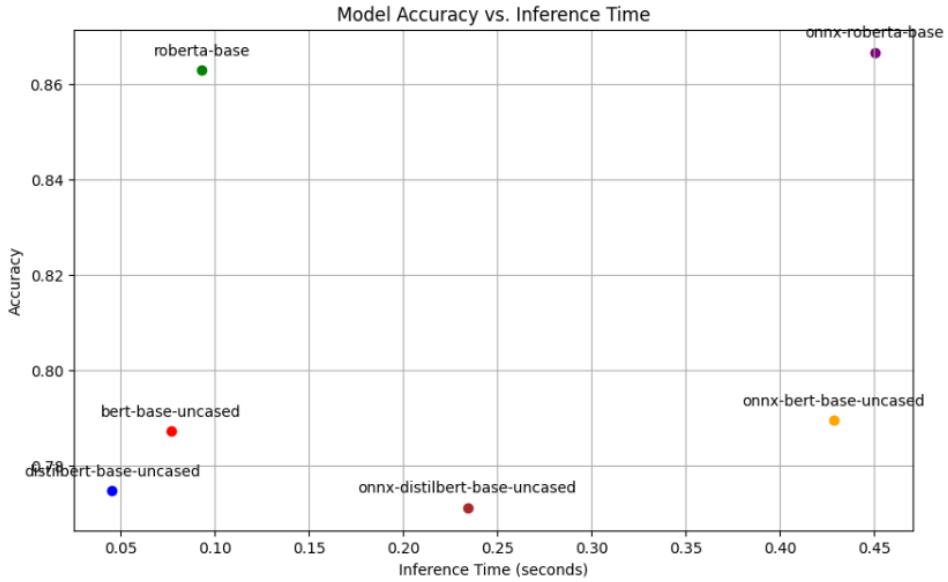


Figure 5.4: Model accuracy versus inference time (350 samples) for each baseline model and its ONNX-CPU counterpart.

- **DistilBERT-base-uncased** (blue) delivers the fastest inference (45 ms) but with the lowest accuracy (77.5 %), making it suitable for highly resource-constrained scenarios where speed is paramount.
- **BERT-base-uncased** (red) strikes a balance at 55 ms and 78.5 % accuracy, offering modest gains over DistilBERT at a small latency cost.
- **RoBERTa-base (GPU)** (green) achieves substantially higher accuracy (86.3 %) with a moderate inference time (95 ms), highlighting the benefit of its enhanced pre-training despite the slower runtime.
- **ONNX-CPU exports** (gold, brown, purple) incur 4–8× longer runtimes but, in the case of RoBERTa (purple), yield a slight accuracy improvement (86.7 %) through runtime optimizations and fused kernels.
  ONNX-BERT and ONNX-DistilBERT show negligible accuracy changes, indicating faithful conversion.

Overall, this plot is evidence that ONNX conversion preserves or improves slightly accuracy at a cost in higher latency, and model choice should be driven by the specific accuracy vs. speed requirements of the destination environment.

In summary, while RoBERTa-base remains the current top off-the-shelf encoder for multi-class news classification, Unsloth-fine-tuned decoder models such as Mistral-Small provide a

very effective alternative with similar accuracy, inference speed, and memory usage. The result demonstrates the effectiveness of PEFT and quantization in putting LLMs into production environments and calls for further exploration of prompt engineering, adapter architecture, and domain adaptation towards more advanced misinformation detection.

Chapter 6

CONCLUSION

In this thesis, we have explored the application of both encoder-only and decoder-only LLMs to the tasks of multiclass news article categorization and binary fake-news detection. Our systematic evaluation demonstrates that transformer-based encoders and lightweight decoder models can both play pivotal roles in automated content moderation and misinformation countermeasures.

First, we established a strong baseline with the encoder-only models. RoBERTa-base emerged as the top performer in the 11-way categorization task, achieving 81.7 % accuracy and an F1-score of 81.6 %, thanks to its extensive pre-training corpus and optimized training objectives. BERT-base-uncased and its distilled counterpart, DistilBERT, achieved 74.4 % and 74.0 % accuracy respectively, offering trade-offs between model size, inference speed, and predictive power. In the binary fake-news detection subtask, DistilBERT's performance dipped slightly (69.8 % accuracy, F1 = 66.7 %), underscoring that reducing to a simpler decision boundary does not automatically translate into higher reliability when model capacity is constrained.

We then investigated decoder-only LLMs fine-tuned via the Unsloth framework, which integrates 4-bit quantization, LoRA adapters, and optimized inference providers. Mistral-Small (7 B) achieved 77.7 % accuracy (F1 = 77.5 %), closely approaching the BERT baseline with only a fraction of full-model fine-tuning. Gemma-3 (4 B) matched BERT's categorization performance (74.1 % / 74.3 %) while LLaMA-3.2 3B (3 B) trailed at 64.7 % / 64.7 %. These results validate that prompt-based adaptation and parameter-efficient fine-tuning can unlock strong performance even in resource-constrained scenarios, provided that careful attention is paid to adapter rank, template design, and instruction clarity.

A key contribution of this work is the demonstration that ONNX Runtime can preserve—and in some cases slightly improve—model accuracy while enabling CPU-based inference. ONNX

exports of BERT, RoBERTa, and DistilBERT incurred 4–8× higher latency (0.25–0.45 s for 350 samples) but retained within 0.1–0.4 % of their original accuracy. This portability across hardware accelerators simplifies deployment in production environments where GPU access may be limited or cost-prohibitive.

Throughout our experiments, we trained and fine-tuned models on different compute resources, including a GPU server at the Laboratory for Internet Computing (LInC) of University of Cyprus and strongest units on Google Colab. We intentionally did not over-optimize for a single GPU type; instead, we emphasize that efficiency—both in terms of GPU hours and monetary cost—must be considered in real-world deployments. Techniques such as mixed precision, gradient accumulation, 4-bit quantization, and parameter-efficient adapters not only reduce VRAM footprint but also dramatically cut training time and cloud compute bills. For example, LoRA adapters typically require only 10–20 % of the full model parameters to be updated, reducing both memory usage and I/O overhead during checkpointing.

Our per-category analyses reveal that all models became really good at detecting overt topical classes such as *rumor*, *fake*, and *political*, yet struggle with categories *clickbait* and *bias*. Addressing these low-signal categories may require a hybrid approach where combining a number of techniques, such as stance detection, sentiment analysis, or discourse parsing—or the integration of external knowledge graphs for fact verification. Moreover, evaluating models on cross-lingual and adversarially perturbed datasets will be essential for robust real-world performance.

In conclusion, this thesis want to show the potential of modern LLMs for protecting people from misinformation and disinformation, while comparing the some of the most powerful models at this time. Understanding the usage of LLMs in real-world problems is crucial and one of our most powerful tools against fake news, yet.

References

[1] Carlos Castillo, Marcelo Mendoza, and Barbara Poblete. "Information credibility on twitter". In: *Proceedings of the 20th International Conference on World Wide Web*. WWW '11. Hyderabad, India: Association for Computing Machinery, 2011, pp. 675–684. ISBN: 9781450306324. DOI: 10.1145/1963405.1963500. https://doi.org/10.1145/1963405.1963500.

[2] Jacob Devlin et al. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv preprint* (2019).

[3] Benjamin D. Horne and Sibel Adali. This Just In: Fake News Packs a Lot in Title, Uses Simpler, Repetitive Content in Text Body, More Similar to Satire than Real News. *ArXiv* abs/1703.09398 (2017). https://api.semanticscholar.org/CorpusID:7083781.

[4] Yinhan Liu et al. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv preprint* (2019).

[5] Martin Potthast et al. A Stylometric Inquiry into Hyperpartisan and Fake News. *ArXiv* abs/1702.05638 (2017). https://api.semanticscholar.org/CorpusID:574574.

[6] Hannah Rashkin et al. "Truth of Varying Shades: Analyzing Language in Fake News and Political Fact-Checking". In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 2017, pp. 2931–2937.

[7] Natali Ruchansky, Sungyong Seo, and Yan Liu. "CSI: A Hybrid Deep Model for Fake News Detection". In: *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. CIKM '17. Singapore, Singapore: Association for Computing Machinery, 2017, pp. 797–806. ISBN: 9781450349185. DOI: 10.1145/3132847.3132877. https://doi.org/10.1145/3132847.3132877.

[8] Kai Shu et al. "Beyond News Contents: The Role of Social Context for Fake News Detection". In: *Proceedings of the 12th ACM International Conference on Web Search and Data Mining (WSDM)*. 2019, pp. 312–320.

[9] Soroush Vosoughi, Deb Roy, and Sinan Aral. The spread of true and false news online. *Science* 359.6380 (2018), pp. 1146–1151. DOI: 10.1126/science.aap9559. eprint: https://www.science.org/doi/pdf/10.1126/science.aap9559. https://www.science.org/doi/abs/10.1126/science.aap9559.

[10] Liqiang Wang et al. “Five Shades of Untruth: Finer-Grained Classification of Fake News”. In: *2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*. 2018, pp. 593–594. DOI: 10.1109/ASONAM.2018.8508256.

[11] William Yang Wang. Liar, Liar Pants on Fire: A New Benchmark Dataset for Fake News Detection. *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (2017).

[12] Thomas Wolf et al. Transformers: State-of-the-Art Natural Language Processing (2020). https://aclanthology.org/2020.emnlp-demos.6/.

[13] Xinyi Zhou and Reza Zafarani. Fake News: Fundamental Theories, Detection Strategies and Challenges. *Proceedings of the ACM on Human-Computer Interaction* 3.CSCW (2019), pp. 1–35.