UNIVERSITY OF PIRAEUS

SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGIES

DEPARTMENT OF DIGITAL SYSTEMS

MASTER OF SCIENCE

"INFORMATION SYSTEMS AND SERVICES"

"Fake News Detection in a Headline-Only Setting: A Comparative Study of Machine Learning and Deep Learning Models on the GossipCop Dataset"

by

Dionysios Sotiropoulos

Submitted

in partial fulfilment of the requirements for the degree of

Master of ……

at the

UNIVERSITY OF PIRAEUS

April 2026

Thesis Supervisor: Michael Filippakis

Title: Professor

Author . . . . . .

ΣΕΛΙΔΑ ΕΓΚΥΡΟΤΗΤΑΣ

ΟνοματεπώνυμοΦοιτητή/Φοιτήτριας: ……..

Τίτλος Μεταπτυχιακής Διπλωματικής Εργασίας: …….

Η παρούσα Μεταπτυχιακή Διπλωματική Εργασία υποβάλλεται ως μερική εκπλήρωση των απαιτήσεων του Προγράμματος Μεταπτυχιακών Σπουδών "Πληροφοριακά Συστήματα & Υπηρεσίες" του Τμήματος Ψηφιακών Συστημάτων του Πανεπιστημίου Πειραιώς και εγκρίθηκε στις …… [ημερομηνία έγκρισης] από τα μέλη της Εξεταστικής Επιτροπής.

Εξεταστική Επιτροπή

Επιβλέπων/ουσα(Τμήμα Ψηφιακών Συστημάτων, Πανεπιστήμιο Πειραιώς)……[ονοματεπώνυμο, βαθμίδα]

Μέλος Εξεταστικής Επιτροπής: ……[ονοματεπώνυμο, βαθμίδα]

Μέλος Εξεταστικής Επιτροπής: ……[ονοματεπώνυμο, βαθμίδα]

ΥΠΕΥΘΥΝΗ ΔΗΛΩΣΗ ΑΥΘΕΝΤΙΚΟΤΗΤΑΣ

Ο/Η……., γνωρίζοντας τις συνέπειες της λογοκλοπής, δηλώνω υπεύθυνα ότι η παρούσα εργασία με τίτλο «……», αποτελεί προϊόν αυστηρά προσωπικής εργασίας και όλες οι πηγές που έχω χρησιμοποιήσει, έχουν δηλωθεί κατάλληλα στις βιβλιογραφικές παραπομπές και αναφορές. Τα σημεία όπου έχω χρησιμοποιήσει ιδέες, κείμενο ή/και πηγές άλλων συγγραφέων, αναφέρονται ευδιάκριτα στο κείμενο με την κατάλληλη παραπομπή και η σχετική αναφορά περιλαμβάνεται στο τμήμα των βιβλιογραφικών αναφορών με πλήρη περιγραφή.

Επιπλέον *δηλώνω υπεύθυνα ότι η συγκεκριμένη Μεταπτυχιακή Διπλωματική Εργασία έχει συγγραφεί από εμένα προσωπικά και δεν έχει υποβληθεί ούτε έχει αξιολογηθεί στο πλαίσιο κάποιου άλλου μεταπτυχιακού ή προπτυχιακού τίτλου σπουδών, στην Ελλάδα ή στο εξωτερικό.*

Παράβαση της ανωτέρω ακαδημαϊκής μου ευθύνης αποτελεί ουσιώδη λόγο για την ανάκληση του πτυχίου μου. Σε κάθε περίπτωση, αναληθούς ή ανακριβούς δηλώσεως, υπόκειμαι στις συνέπειες που προβλέπονται τις διατάξεις που προβλέπει η Ελληνική και Κοινοτική Νομοθεσία περί πνευματικής ιδιοκτησίας.

Ο/Η ΔΗΛΩΝ/ΟΥΣΑ

Ονοματεπώνυμο:

Αριθμός Μητρώου:

Υπογραφή:

Table of Contents

List of Figures

List of Tables

Abstract

The rapid spread of misinformation through digital platforms has made fake news detection an important problem in contemporary data-driven research. This thesis investigates the effectiveness of Machine Learning and Deep Learning approaches for fake news detection in a constrained headline-only setting, where the available textual information is limited and the dataset is significantly imbalanced. The study is based on the GossipCop subset of FakeNewsNet and focuses exclusively on headline text in order to evaluate model behavior under controlled content-based conditions.

A comparative experimental framework was developed including four Machine Learning models, namely Logistic Regression, Linear Support Vector Classification, Random Forest, and XGBoost, as well as four Deep Learning approaches: Convolutional Neural Networks, Bidirectional Long Short-Term Memory networks, DistilBERT, and a hybrid DistilBERT + XGBoost model. For the Machine Learning models, headlines were represented using 300-dimensional Doc2Vec embeddings and evaluated with stratified 10-fold cross-validation, while class imbalance was handled through SMOTE applied only to the training folds. For the Deep Learning models, tokenized or transformer-based headline representations were used within a holdout evaluation framework, with class weighting employed where appropriate.

The results show a clear performance gap between the two model families. The Machine Learning baselines, especially Logistic Regression and LinearSVC, exhibited weak discriminative performance, while Random Forest and XGBoost improved overall accuracy but remained ineffective in recovering fake news instances. In contrast, the Deep Learning models achieved substantially stronger and more balanced results. DistilBERT provided the best overall balance across ROC-AUC, Cohen's Kappa, fake-class F1-score, and weighted F1-score. The hybrid DistilBERT + XGBoost model achieved the highest overall accuracy, although at the cost of lower fake recall, whereas BiLSTM demonstrated the strongest ability to recover fake news instances.

The findings indicate that transformer-based and other Deep Learning approaches are more suitable than traditional Machine Learning methods for headline-based fake news

detection. At the same time, the study highlights the strong influence of class imbalance and the inherent limitations of headline-only datasets, which restrict contextual depth and constrain model performance. Overall, the thesis shows that fake news detection based solely on headlines remains a relevant but inherently difficult classification problem.

1. Introduction

1.1. Scope of the Thesis

In the contemporary digital environment, the rapid dissemination of misinformation and fake news through online platforms has emerged as a critical challenge for the public sphere, social stability, and the formation of public opinion. The widespread adoption of social media and digital news platforms has significantly lowered the barriers to content creation and distribution, enabling false or misleading information to spread rapidly and reach large audiences within a short time frame.

The scale and velocity of online information diffusion render manual verification processes largely impractical, as the volume of published content far exceeds the capacity of human fact-checking mechanisms. Consequently, the detection and classification of fake news increasingly rely on automated computational approaches. Such methods aim to assist journalists, fact-checkers, and digital platforms by identifying potentially misleading content in an efficient, scalable, and systematic manner.

Within this context, the present thesis focuses on the automatic detection of fake news using machine learning and deep learning techniques, with particular emphasis on textual information derived from news headlines. The study examines the effectiveness of different modeling approaches and highlights their limitations when applied to short-text datasets, where the available linguistic signal is inherently constrained.

1.2. Importance of Fake News Detection

Fake news detection has become a matter of significant importance due to the profound influence that misinformation can exert on modern society, including public opinion, public health, political decision making, and social cohesion. Rather than representing isolated events, fake news often reflects coordinated attempts to manipulate information flows by

exploiting structural and algorithmic characteristics of digital platforms (Shu et al., 2017, Rai et al., 2022).

The persistent circulation of inaccurate or deceptive information can result in widespread misinformation, erosion of trust in public institutions, and the amplification of social polarization. In sensitive domains such as healthcare and politics, exposure to false information may lead to harmful individual behavior and large-scale societal consequences.

For these reasons, the development of reliable automated methods for fake news detection represents a critical step toward strengthening the integrity of digital information ecosystems. By leveraging advances in natural language processing, machine learning, and deep learning, such approaches aim to improve the early identification of misleading content and contribute to a more informed and resilient digital public discourse.

1.3. Fake News and Social Media

The rapid expansion of social media platforms has significantly transformed the way information is produced, shared, and consumed. Unlike traditional media environments, social media enable fast and decentralized dissemination, allowing users not only to access news content but also to redistribute, comment on, and amplify it at scale. This structural shift has made social media a particularly influential environment for the circulation of both legitimate and misleading information (Shu et al., 2017; Shu et al., 2018).

Fake news spreads efficiently in social media ecosystems because visibility is often driven by speed, reach, and user engagement rather than editorial verification or source credibility. As a result, false or manipulated information can be disseminated rapidly, while users may further contribute to its spread without systematically verifying its reliability. This dynamic increases the societal impact of misinformation and strengthens the need for effective detection mechanisms (Shu et al., 2017; Shu et al., 2018).

These characteristics also make fake news detection particularly challenging in online environments. As highlighted in the literature, news content alone is often insufficient for reliable classification, especially in short and fragmented forms of communication. Social media posts and headlines typically provide limited contextual depth, which restricts the amount of exploitable semantic information and makes credibility assessment more difficult when only textual content is available (Shu et al., 2017; Shu et al., 2019).

1.4. Objectives and Research Questions

The primary objective of this thesis is to investigate the effectiveness of machine learning and deep learning approaches for the automatic detection of fake news using headline-based textual data. The study aims to assess how different modeling paradigms perform when applied to a dataset characterized by limited textual information and significant class imbalance.

More specifically, the objectives of this work are as follows:

- To evaluate and compare traditional machine learning models and deep learning architectures for fake news detection.
- To analyze the impact of class imbalance on model performance and examine the extent to which common mitigation techniques influence the results.
- To investigate whether deep learning models demonstrate improved capability in identifying the minority class (fake news) compared to traditional machine learning approaches.
- To assess the limitations of headline-only datasets and identify factors that constrain the performance of automated fake news detection systems.

Based on these objectives, the research is guided by the following research questions:

- How do machine learning and deep learning models compare in terms of performance when applied to headline-based fake news detection?
- To what extent does class imbalance affect evaluation metrics, particularly for the detection of fake news?
- Why do different models exhibit converging performance levels despite architectural differences?
- What insights can be derived regarding the suitability of headline-only datasets for automated fake news detection?

1.5. Structure of the Thesis

Chapter 2 presents the theoretical background and reviews related work on fake news detection, including Machine Learning, Deep Learning, transformer-based, and hybrid approaches, as well as the datasets commonly used in this research area.

Chapter 3 describes the dataset employed in this study and details the preprocessing procedures applied for both Machine Learning and Deep Learning models.

Chapter 4 outlines the methodology and experimental setup, including the selected models, evaluation metrics, imbalance-handling strategies, and validation protocols.

Chapter 5 presents the experimental results obtained from the evaluated Machine Learning and Deep Learning models.

Chapter 6 provides a comparative evaluation of the models and positions the findings of the thesis in relation to selected published studies.

Finally, Chapter 7 discusses the results in light of the characteristics of the dataset and the selected experimental design, and outlines the main limitations of the study together with directions for future research.

2. Theoretical Background and Related Work

2.1. Fake News: Definitions and Taxonomy

The term *fake news* has been extensively discussed in the literature and is generally used to describe news content that is intentionally false or misleading while being presented in the format of legitimate journalism. A key characteristic that differentiates fake news from other forms of inaccurate information is the presence of deliberate intent to deceive, which distinguishes it from unintentional misinformation or reporting errors (Shu et al., 2017).

Several studies emphasize that fake news should not be treated as a uniform phenomenon, but rather as a broad category encompassing multiple forms of deceptive content. Common taxonomies distinguish between fabricated news, manipulated content, misleading headlines, and false contextualization, where genuine information is presented in a distorted or incomplete manner (Reis et al., 2019). Satirical content is often included in such taxonomies, although it differs from fake news in that its primary purpose is not deception, but it may still contribute to misinformation when contextual cues are missing.

From a computational perspective, the complexity of fake news has led most studies to adopt simplified problem formulations in order to enable systematic experimentation. As a result, the majority of fake news detection approaches model the task as a supervised classification problem, typically using binary labels such as *fake* and *real* (Shu et al., 2017; Albahar et al., 2021). This abstraction facilitates model comparison and reproducibility, but it also limits the ability to capture nuanced distinctions between different types of deceptive content.

Another important dimension discussed in the literature concerns the textual granularity of the analyzed data. While some studies focus on full-length news articles, others restrict the analysis to shorter textual units such as headlines or brief summaries. Headline-based fake news detection represents a particularly challenging setting, as headlines are often designed to maximize attention rather than to convey comprehensive factual information (Rai et al., 2022). Consequently, linguistic cues available for classification are sparse, ambiguous, and often stylistic rather than semantic.

Despite these limitations, headline-based formulations remain widely used due to data availability constraints and their relevance to real-world social media environments, where users are frequently exposed only to short textual snippets. For this reason, numerous studies continue to investigate the extent to which automated models can reliably distinguish fake from real news under such restrictive conditions (Reis et al., 2019; Shu et al., 2017).

Overall, the definitions and taxonomies proposed in the literature provide a conceptual foundation for fake news detection research. They also highlight the inherent trade-offs between problem simplification and representational fidelity, which directly influence the design and evaluation of computational models reviewed in the subsequent sections.

2.2. Dissemination Mechanisms of Fake News

The dissemination of fake news in online environments is governed by information diffusion mechanisms that prioritize engagement and visibility rather than content credibility. In digital platforms, news propagation emerges from the interaction between users, platform algorithms, and temporal dynamics, resulting in complex diffusion patterns that can amplify deceptive content at scale (Shu et al., 2017).

A central factor in fake news dissemination is *virality*. Prior studies report that misleading or sensational content tends to attract higher user engagement, increasing the likelihood of sharing and further exposure through algorithmic recommendation systems (Reis et al., 2019). This engagement-driven amplification creates feedback loops in which highly interactive content gains disproportionate visibility, irrespective of its factual accuracy.

User behavior also plays a critical role in shaping dissemination dynamics. Users often act as intermediaries in the information flow by reposting or reacting to content without verifying its reliability. Such interactions serve as implicit endorsement signals that influence subsequent distribution, contributing to the rapid spread of false information (Shu et al., 2017). Temporal analyses further suggest that fake news may exhibit faster early-stage

diffusion compared to real news, although long-term engagement patterns can vary across datasets and domains (Reis et al., 2019).

From a modeling perspective, dissemination mechanisms are commonly analyzed using temporal features, diffusion trees, and network-based representations. These approaches aim to capture structural differences between fake and real news propagation, such as cascade depth, spread speed, and interaction diversity (Rai et al., 2022). Empirical results indicate that dissemination-aware models can significantly improve detection performance when such signals are available.

However, the practical use of dissemination features is constrained by data accessibility. Many datasets lack detailed interaction logs or propagation histories due to privacy concerns and platform access restrictions. As a result, a substantial portion of the literature relies on content-only data, abstracting away dissemination dynamics despite their acknowledged importance (Shu et al., 2017; Albahar et al., 2021).

Overall, dissemination mechanisms provide valuable insight into the behavior of fake news in online ecosystems. Nevertheless, the limited availability of diffusion-level data often necessitates simplified modeling assumptions, motivating continued research into approaches that can operate effectively under restricted data conditions.

2.3. Social Media and Fake News Propagation

Digital social platforms have fundamentally reshaped how news content is produced, consumed, and redistributed at scale. Unlike traditional media, social networks enable decentralized and user-driven content sharing, allowing individuals to act simultaneously as information consumers and distributors. Through actions such as reposting, liking, and commenting, users collectively influence the visibility and reach of news items (FakeNewsNet).

The literature highlights that fake news dissemination on social media is not solely driven by textual content, but also by interaction patterns and engagement dynamics. Deceptive or sensational headlines are often designed to maximize attention, increasing the likelihood of user interaction and algorithmic amplification. As a result, fake news may spread

rapidly in its early stages, achieving high exposure despite limited factual credibility (Shu et al., 2017).

To capture these phenomena, several studies emphasize the importance of *social context*, defined as auxiliary information related to user behavior, interaction networks, and propagation structures. Social context features may include repost cascades, temporal diffusion patterns, and user connectivity, all of which can provide discriminative signals for fake news detection beyond textual content alone (Reis et al., 2019).

An influential example of a social-context-aware framework is the FakeNewsTracker system, which was introduced as a research-oriented platform for collecting and organizing fake news data alongside associated social media interactions (Shu et al., 2019). FakeNewsTracker was designed to integrate news articles with user engagement data and propagation information, offering a unified view of content and diffusion dynamics. Although this framework is no longer operational and is not utilized in the present study, it remains highly relevant from a conceptual perspective, as it illustrates how social context can be systematically incorporated into fake news detection pipelines.

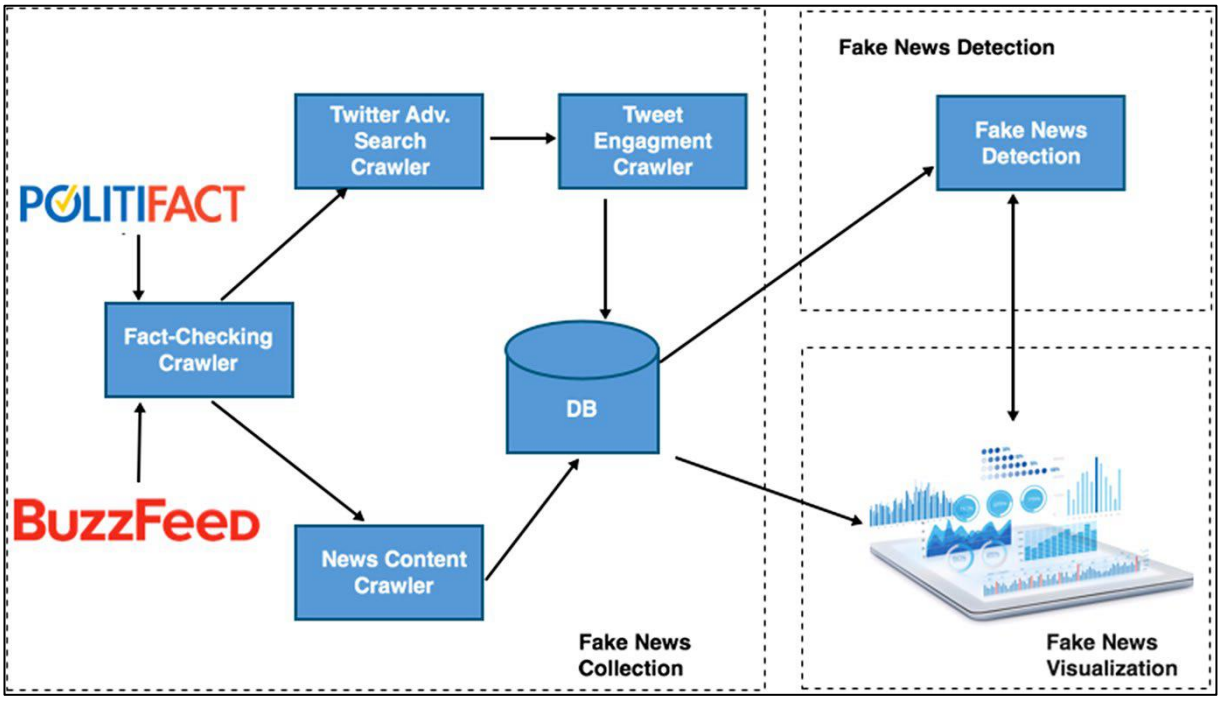

Figure 1. Conceptual illustration of social context and propagation signals in fake news detection, adapted from FakeNewsTracker

(It should be noted that the framework illustrated in Figure X is presented solely for conceptual purposes and reflects a general approach to incorporating social context in fake news detection. The present thesis focuses exclusively on the GossipCop subset of FakeNewsNet and does not utilize fact-checking sources such as PolitiFact or BuzzFeed.)

Despite the demonstrated effectiveness of social-context-based approaches, their practical adoption has become increasingly challenging. Access to detailed interaction data is now heavily restricted due to platform policies and privacy regulations, limiting the availability of social context features for large-scale analysis. Consequently, many recent studies focus on content-only approaches, even while acknowledging that the absence of social signals may constrain detection performance (Albahar et al., 2021).

Overall, social media propagation characteristics constitute a crucial dimension of fake news analysis. However, the trade-off between methodological effectiveness and data accessibility has led to a growing reliance on textual information alone, motivating the exploration of robust content-based and hybrid approaches discussed in subsequent sections.

2.4. Overview of Existing Research Approaches

Research on fake news detection has evolved along three primary methodological directions: content-based approaches, social-context-based approaches, and hybrid models that combine multiple information sources. Each category reflects different assumptions regarding data availability and the nature of misinformation signals.

Content-based approaches focus exclusively on the textual characteristics of news items. Early studies relied on manually engineered linguistic features such as word frequencies, stylistic markers, sentiment indicators, and syntactic patterns, which were subsequently used to train traditional machine learning classifiers including Support Vector Machines, Logistic Regression, and Random Forests (Shu et al., 2017; Albahar et al., 2021).

These methods offer interpretability and computational efficiency but often struggle to capture deeper semantic relations, particularly in short-text settings such as headlines.

With the advancement of representation learning, deep learning models have increasingly replaced handcrafted feature pipelines in content-based fake news detection. Neural architectures such as Convolutional Neural Networks and Recurrent Neural Networks have been employed to automatically learn hierarchical text representations from raw input, demonstrating improved performance compared to traditional machine learning models in several benchmark datasets (Reis et al., 2019). More recently, transformer-based models, including BERT and its variants, have achieved state-of-the-art results by leveraging contextualized word embeddings and large-scale pretraining (Rai et al., 2022).

In parallel, social-context-based approaches incorporate information beyond textual content, exploiting user interactions, propagation structures, and network-level features. Studies adopting this paradigm model fake news diffusion as a graph-based or temporal process, aiming to capture behavioral differences between fake and real news dissemination patterns (Shu et al., 2017). Although these approaches often yield superior performance, their applicability is limited by the availability of social interaction data and access restrictions imposed by online platforms.

Hybrid models seek to bridge the gap between content-only and social-context-based approaches by combining textual representations with auxiliary features derived from user behavior or propagation statistics. Such models have been shown to enhance detection performance by integrating complementary information sources, particularly in settings where textual signals alone are insufficient (Albahar et al., 2021). Hybrid architectures may involve feature-level fusion, model-level ensembling, or sequential integration of deep learning and traditional classifiers.

Despite the diversity of proposed methods, comparative analyses across different datasets reveal that performance gains are often constrained by dataset characteristics, including text length, class imbalance, and annotation quality. As a result, many studies report converging performance levels across distinct modeling paradigms when applied to headline-

only datasets, highlighting the inherent limitations of content-centric fake news detection (Reis et al., 2019; Rai et al., 2022).

This body of work underscores the importance of aligning methodological choices with dataset properties and motivates the systematic evaluation of both traditional and deep learning models conducted in the present study.

2.5. Machine Learning Techniques for Fake News Detection

Traditional machine learning techniques have been widely applied to the task of fake news detection, particularly in early and content-centric studies. These approaches typically rely on transforming textual data into numerical feature representations, followed by the application of supervised classification algorithms. Despite their relative simplicity compared to deep learning models, traditional machine learning methods remain important baselines due to their interpretability, efficiency, and robustness in low-resource settings (Shu et al., 2017; Albahar et al., 2021).

2.5.1. Logistic Regression

Logistic Regression is a widely adopted baseline classifier in text classification tasks and has been extensively used in fake news detection studies due to its simplicity, interpretability, and computational efficiency. Despite its name, Logistic Regression is a linear classification model that estimates the probability of an instance belonging to a given class through a linear decision function combined with a nonlinear activation.

Given an input feature vector $\mathbf{x}$, the model computes a linear score as a weighted sum of the input features and a bias term. This score is then transformed into a probability value through the sigmoid function, which maps real-valued inputs to the interval $[0, 1]$:

$$\boldsymbol{P(y = 1 \mid x) = \sigma(w^{\top} x + b) = \frac{1}{1 + e^{-(w^{\top} x + b)}}}$$

where $\mathbf{w}$ denotes the learned weight vector and $\boldsymbol{b}$ the bias term. Classification is performed by applying a threshold (typically 0.5) to the estimated probability.

Logistic Regression is frequently used as a strong baseline for fake news detection, particularly when headlines are represented through TF IDF vectors, bag of words features, or dense embeddings (Reis et al., 2019). Its main advantage lies in its transparency: each learned weight directly reflects the contribution of a specific feature to the classification decision, allowing for straightforward interpretation of model behavior.

However, Logistic Regression assumes a linear relationship between the input features and the log-odds of the target variable. As a result, it is inherently limited in capturing complex, nonlinear linguistic patterns that often characterize deceptive or misleading content. This limitation becomes particularly pronounced in headline-based fake news detection, where texts are short, context is minimal, and subtle semantic cues play a crucial role (Zhou and Zafarani, 2020).

Consequently, while Logistic Regression provides a strong and interpretable baseline for comparison, its expressive capacity is restricted compared to more advanced machine learning and deep learning models. This makes it especially useful for establishing reference performance levels rather than achieving state-of-the-art results in complex fake news detection scenarios.

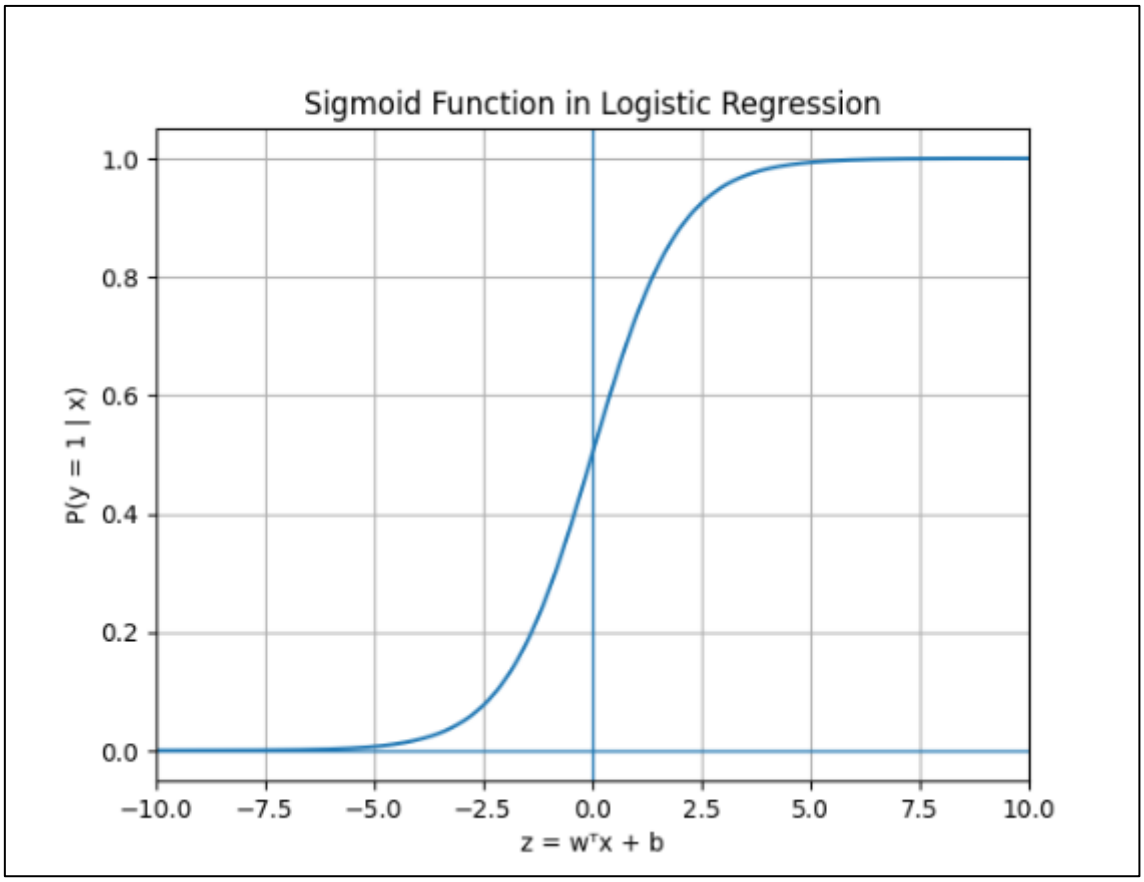

Figure 2. Sigmoid function in logistic regression illustrating the mapping from the linear model output to class probability

2.5.2. Support Vector Machines

Support Vector Machines (SVMs) constitute a class of supervised learning models that aim to construct an optimal decision boundary between classes by maximizing the margin between them. Unlike probabilistic classifiers, SVMs focus on identifying a separating hyperplane that achieves the greatest possible distance from the closest training samples of each class, known as support vectors.

Formally, given a set of labeled training samples $\{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$, where $y_i \in \{-1, +1\}$, the hard-margin SVM optimization problem is defined as:

$$\min_{w,b} \frac{1}{2} \| w \|^2 \text{ subject to } y_i(w^\top x_i + b) \geq 1, \forall i$$

Here, $\mathbf{w}$ represents the normal vector of the separating hyperplane and $\boldsymbol{b}$ its bias term. Maximizing the margin is equivalent to minimizing the norm of $\mathbf{w}$, which leads to improved generalization under certain theoretical assumptions.

In practical applications, especially in text classification, soft-margin SVMs are commonly employed to allow for misclassifications when the data are not perfectly linearly separable. This is achieved through the introduction of slack variables and a regularization parameter C, which controls the trade-off between margin maximization and classification error.

SVMs have been widely adopted in fake news detection research due to their strong theoretical grounding and their effectiveness in high-dimensional feature spaces characteristic of textual data (Shu et al., 2017). In particular, linear SVMs are frequently preferred for text-based tasks, as they scale efficiently with large vocabularies and perform well when features are sparse or embedding-based.

Despite these advantages, SVMs exhibit notable sensitivity to class imbalance. When the training data are skewed toward a majority class, the resulting decision boundary may be biased, favoring correct classification of majority-class instances at the expense of minority-class recall. This limitation is especially relevant in fake news detection, where fake news instances often represent a small fraction of the dataset (Zhou and Zafarani, 2020).

As a result, while SVMs provide a robust and theoretically well-founded baseline, their performance in headline-based fake news detection is constrained by both linear separability assumptions and imbalance-related effects. These characteristics make SVMs suitable for comparative evaluation but highlight the need for more expressive models when subtle semantic distinctions are required.

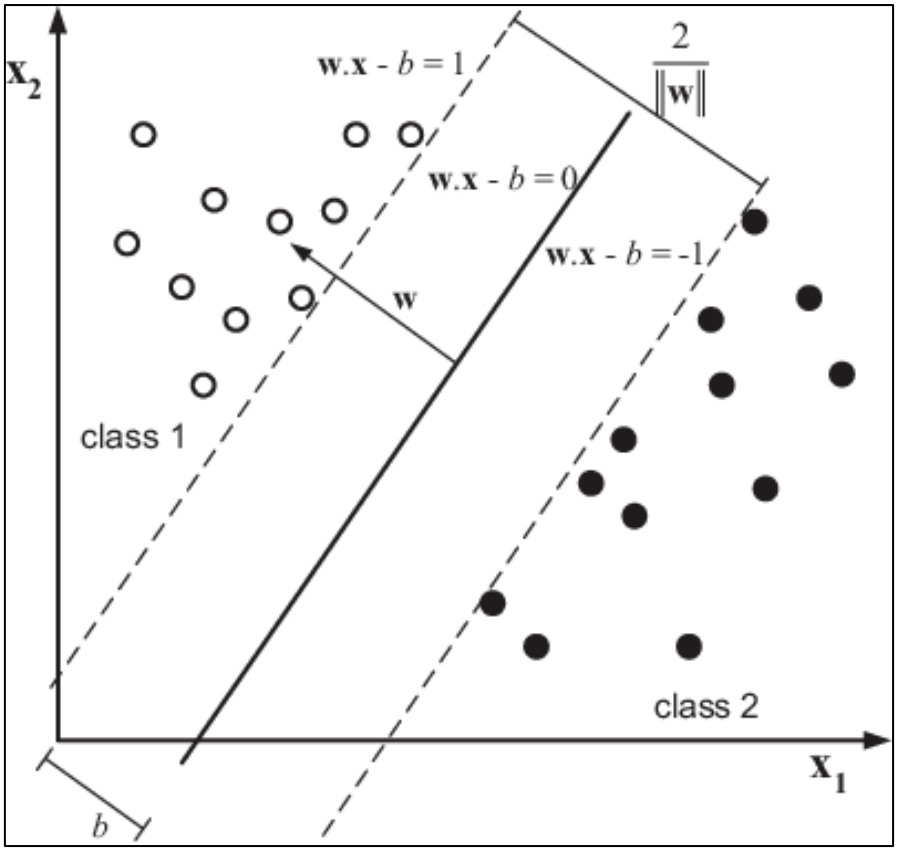

Figure 3. Illustrative diagram of a linear Support Vector Machine classifier showing the separating hyperplane and maximum-margin boundaries.

2.5.3. Random Forest

Random Forest is an ensemble learning algorithm that combines the predictions of multiple decision trees in order to improve classification robustness and generalization performance. Instead of relying on a single tree, the model constructs a collection of trees during training and aggregates their outputs through majority voting in the case of classification tasks.

Each decision tree in a Random Forest is trained on a bootstrapped subset of the original training data, a process known as bagging (bootstrap aggregating). In addition, at each split within a tree, only a random subset of the available features is considered. This dual source of randomness reduces correlation between individual trees and mitigates the risk of overfitting, which is a common limitation of standalone decision trees.

From a modeling perspective, Random Forests are capable of capturing nonlinear relationships and complex feature interactions without requiring explicit feature engineering. This property makes them particularly appealing for fake news detection, where linguistic patterns and semantic cues may interact in nontrivial ways. Prior studies have demonstrated that Random Forest classifiers can achieve competitive performance when applied to both handcrafted linguistic features and dense text representations, such as document embeddings (Albahar et al., 2021).

Despite these strengths, Random Forests are not immune to the challenges posed by imbalanced datasets. When the majority class dominates the training data, individual trees may learn decision rules that favor majority-class predictions, resulting in high overall accuracy but poor recall for the minority class. This behavior is especially pronounced in headline-based fake news detection, where the minority class often contains limited and ambiguous textual signals.

Furthermore, while Random Forests offer improved flexibility compared to linear models, their interpretability decreases as the number of trees increases. Although feature importance measures can provide some insight into model behavior, the ensemble nature of the algorithm makes it difficult to trace individual predictions back to specific decision paths.

In summary, Random Forest represents a powerful non-linear baseline for fake news detection, offering improved modeling capacity over linear classifiers while maintaining reasonable computational efficiency. However, its effectiveness remains constrained by class imbalance and the limited contextual information inherent in headline-only datasets, motivating comparisons with more expressive deep learning approaches.

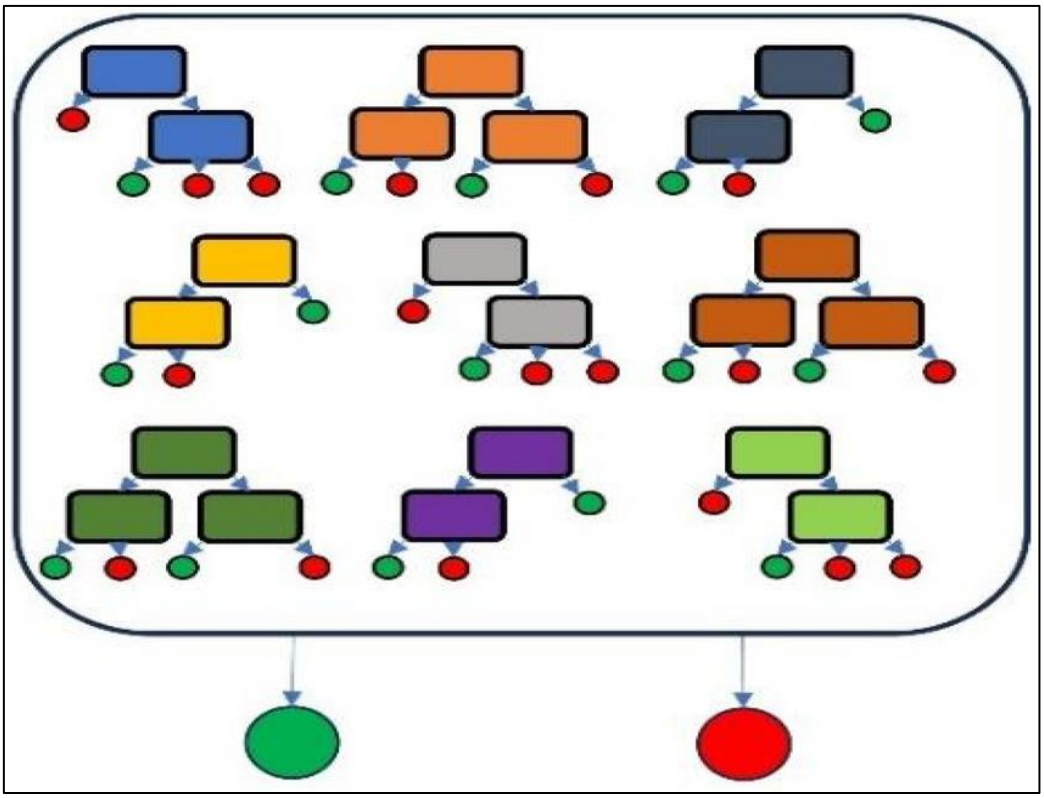

Figure 4. Illustration of a Random Forest model composed of multiple decision trees, where the final classification is obtained through majority voting.

2.5.4. Extreme Gradient Boosting (XGBoost)

Extreme Gradient Boosting (XGBoost) is an advanced ensemble learning framework based on the gradient boosting paradigm. Unlike bagging-based methods such as Random Forests, boosting constructs models sequentially, where each new learner is trained to reduce the errors made by the ensemble built so far. In practice, XGBoost combines multiple weak learners—typically, shallow decision trees—into a strong classifier by iteratively optimizing a global objective function.

Formally, the objective function optimized by XGBoost can be expressed as:

$$\mathcal{L} = \sum_{i} l(y_i, \hat{y}_i) + \sum_{k} \Omega(f_k)$$

where $\boldsymbol{l}(\cdot)$ denotes a differentiable loss function measuring the discrepancy between the true label $y_i$and the predicted output $\hat{\boldsymbol{y}}_{\boldsymbol{i}}$, and $\boldsymbol{\Omega}(\boldsymbol{f}_{\boldsymbol{k}})$ is a regularization term that penalizes model complexity. The regularization component plays a crucial role in controlling overfitting by constraining tree depth, number of leaves, and leaf weights.

A key advantage of XGBoost lies in its efficient handling of structured and tabular data, as well as its robustness to feature interactions and nonlinear relationships. The algorithm incorporates several system-level and algorithmic optimizations, including parallelized tree construction, shrinkage, and column subsampling, which contribute to both improved performance and computational efficiency.

When engineered feature representations are available, XGBoost is often selected for fake news detection because boosting can model non linear interactions more effectively than a single tree (Reis et al., 2019). Its ability to model complex decision boundaries often results in improved generalization compared to single-tree classifiers and linear models.

However, the effectiveness of XGBoost is strongly dependent on the quality and expressiveness of the input feature representation. When applied to headline-only datasets, where semantic information is limited and textual cues are subtle, the model may struggle to adequately discriminate between fake and real news without carefully designed features. As observed in prior studies, gradient boosting methods tend to favor the majority class under severe class imbalance, leading to high overall accuracy but reduced recall for the minority class (Shu et al., 2017).

Moreover, unlike deep learning architectures that learn hierarchical representations directly from raw text, XGBoost does not inherently capture sequential or contextual dependencies. This limitation motivates its comparison with neural models and its use in hybrid configurations, where pretrained language model embeddings are combined with gradient boosting classifiers.

In summary, XGBoost represents a powerful and flexible machine learning approach for fake news detection, particularly when applied to dense and informative feature representations. Nevertheless, its performance in headline-based settings remains

constrained by feature quality and class imbalance, underscoring the need for complementary deep learning approaches explored later in this thesis.

2.5.5. Limitations of Traditional Machine Learning Approaches

Model	Core Principle	Strengths	Limitations
Logistic Regression	Linear probabilistic classifier with sigmoid decision function	Interpretable, computationally efficient, stable baseline	Limited expressive power, linear decision boundary, poor minority-class recall under class imbalance
Linear SVM	Margin-maximizing linear classifier	Effective in high-dimensional spaces, robust to overfitting	Sensitive to class imbalance, limited nonlinear modeling capacity
Random Forest	Ensemble of decision trees using bagging and feature randomness	Captures nonlinear interactions, robust to noise	Bias toward majority class, limited performance on sparse semantic signals
XGBoost	Gradient boosting ensemble optimizing sequential residuals	Strong generalization on structured features, regularization support	Performance heavily dependent on feature quality, limited contextual modeling

Table 1. Summary of machine learning models used in this study, highlighting their main characteristics, strengths, and limitations in headline-based fake news detection

Despite their widespread adoption, traditional machine learning models exhibit inherent limitations when applied to headline-based fake news detection. Their reliance on

fixed, pre-engineered feature representations constrains their capacity to capture deeper contextual and semantic relationships, which are often essential for distinguishing misleading content from legitimate news in short-text settings.

Moreover, the pronounced class imbalance present in real-world datasets further exacerbates these limitations. As observed both in prior studies and in the experimental results of this thesis, different machine learning classifiers tend to converge toward similar performance levels, particularly favoring the majority class. This convergence suggests that, under limited linguistic signal, model architecture plays a secondary role compared to the information content of the features themselves (Rai et al., 2022).

These observations highlight a fundamental ceiling in the effectiveness of traditional machine learning approaches for headline-only fake news detection and motivate the transition toward deep learning architectures, which aim to learn richer and more expressive representations directly from raw textual data.

2.6. Deep Learning Techniques for Fake News Detection

Deep learning approaches have become increasingly prominent in fake news detection research due to their ability to learn complex and hierarchical representations directly from textual data. In contrast to traditional machine learning models, which rely on manually engineered features, deep learning architectures automatically extract semantic, syntactic, and contextual patterns from raw or minimally processed text. This property has motivated their widespread adoption in recent content-based fake news detection studies (Shu et al., 2017; Reis et al., 2019).

In headline-based datasets, where textual information is inherently limited, deep learning models aim to maximize the utility of short input sequences by leveraging distributed word representations and contextual encoding mechanisms. The following subsections

describe the deep learning architectures examined in this thesis, along with their application to fake news detection.

2.6.1. Convolutional Neural Networks (CNNs)

Convolutional Neural Networks (CNNs) are feed-forward neural architectures that were originally introduced for image processing tasks and later adapted to natural language processing applications. In text classification problems, CNNs operate on sequences of word embeddings and apply convolutional filters in order to extract local and position-invariant textual patterns, such as n-grams and short phrase-level features.

Given a sequence of word embeddings representing a headline, a convolution operation applies a sliding window over consecutive tokens. Formally, the convolution operation can be expressed as:

$$\boldsymbol{c_i} = \boldsymbol{f}(\mathbf{w} \cdot \mathbf{x}_{\boldsymbol{i:i+h-1}} + \boldsymbol{b})$$

where $\mathbf{x}_{\boldsymbol{i:i+h-1}}$ denotes a window of $\boldsymbol{h}$ consecutive word embeddings, w represents a learned filter, $\boldsymbol{b}$ is a bias term, and $\boldsymbol{f}(\cdot)$ is a nonlinear activation function. The resulting feature maps capture local lexical patterns that are indicative of class membership.

To reduce dimensionality and retain the most salient features, convolutional layers are typically followed by pooling operations, most commonly max pooling. This mechanism selects the strongest activation for each filter, allowing the model to focus on the most informative phrases regardless of their position within the text. As a result, CNNs are largely insensitive to word order beyond the local context defined by the filter size.

CNN based text classifiers are widely used for content only fake news detection, since they can capture discriminative local patterns that appear in short, attention grabbing headlines. By employing multiple filters with different window sizes, CNNs are capable of capturing patterns of varying granularity, ranging from short expressions to slightly longer

phrase structures. This property makes them particularly suitable for headline-based datasets, where texts are short and global contextual information is inherently limited (Shu et al., 2017; Rai et al., 2022).

Several empirical studies report that CNN-based architectures outperform traditional machine learning classifiers in content-only fake news detection tasks, while maintaining relatively low computational cost and stable training behavior (Albahar et al., 2021). However, CNNs exhibit certain limitations. Their focus on local patterns restricts their ability to model long-range dependencies and broader semantic relationships, which may be necessary for understanding more complex forms of misinformation. Despite this limitation, CNNs remain a strong baseline for short-text fake news detection and are frequently adopted in comparative experimental studies.

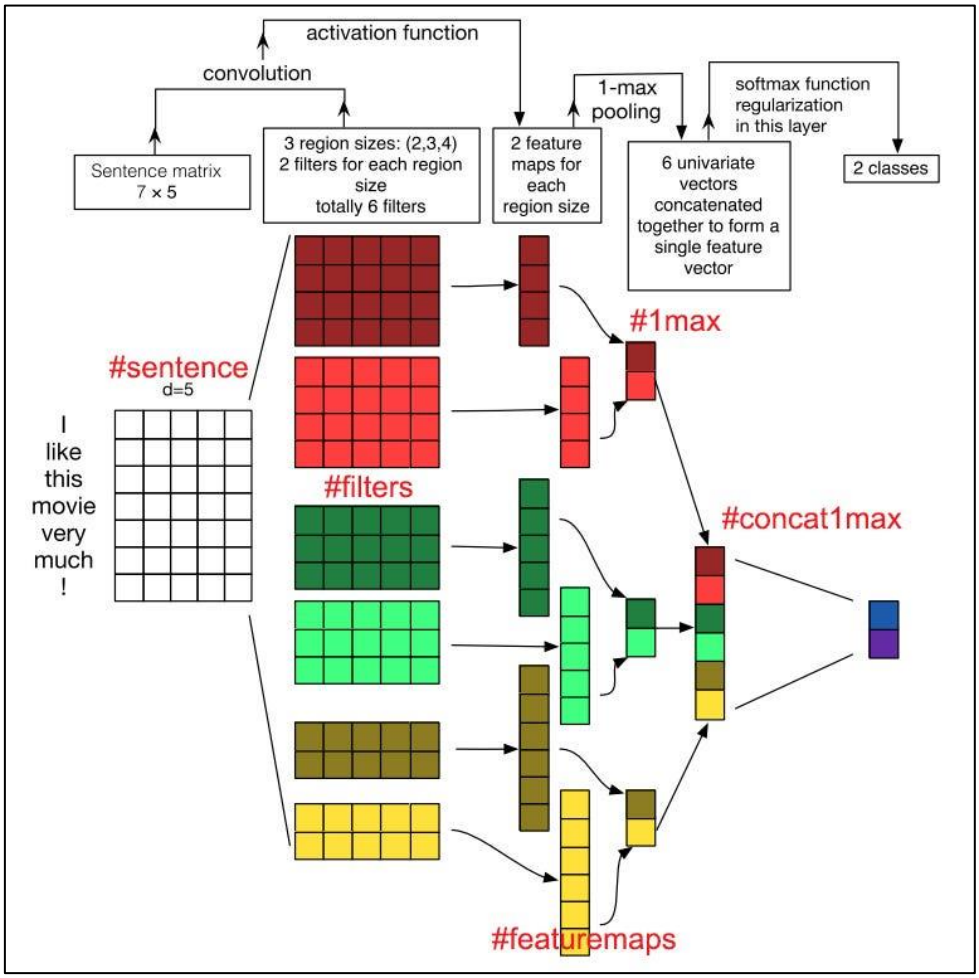

Figure 5. Illustrative architecture of a Convolutional Neural Network for text classification, showing convolution over word embeddings, max-pooling, and feature concatenation.

2.6.2. Recurrent Neural Networks and Bidirectional LSTM

Recurrent Neural Networks (RNNs) are neural architectures specifically designed to model sequential data by maintaining an internal hidden state that evolves over time. This structure allows RNNs to explicitly capture word order and contextual dependencies, making them suitable for natural language processing tasks where the meaning of a word depends on its surrounding context. At each time step, the hidden state is updated based on the current input and the previous hidden state, enabling information to propagate through the sequence.

Despite their theoretical suitability for sequence modeling, standard RNNs suffer from the vanishing gradient problem, which limits their ability to learn long-range dependencies during training. As gradients are propagated backward through time, they may diminish exponentially, preventing the network from effectively capturing relationships between distant tokens.

Long Short-Term Memory (LSTM) networks were introduced to address this limitation by incorporating gating mechanisms that regulate information flow. Specifically, LSTMs employ input, forget, and output gates to control which information is stored, updated, or discarded over time. This design enables LSTMs to preserve relevant contextual information across longer sequences and improves training stability.

Bidirectional LSTM (BiLSTM) architectures further extend this capability by processing the input sequence in both forward and backward directions. This allows the model to incorporate information from both past and future context when generating representations for each token. The resulting hidden representation at each time step is obtained by concatenating the forward and backward hidden states:

$$\boldsymbol{h_t} = [\overrightarrow{\boldsymbol{h_t}}; \overleftarrow{\boldsymbol{h_t}}]$$

where $\overrightarrow{\boldsymbol{h_t}}$ and $\overleftarrow{\boldsymbol{h_t}}$ denote the forward and backward LSTM hidden states, respectively.

BiLSTM architectures are typically applied to fake news detection when the goal is to model word order and contextual dependencies, exploiting sequence information that simpler models may ignore (Reis et al., 2019; Rai et al., 2022). By considering the full bidirectional context of a headline, BiLSTMs can model subtle linguistic cues that are not easily captured by models focusing solely on local patterns.

However, the effectiveness of BiLSTM architectures is influenced by the length and richness of the input text. In headline-only datasets, where sequences are short and contextual depth is inherently limited, the advantages of modeling long-range dependencies may be reduced. As reported in prior studies, BiLSTM-based models often yield moderate performance improvements over simpler architectures, but these gains tend to diminish when applied to short-text fake news detection scenarios (Reis et al., 2019; Rai et al., 2022). Additionally, BiLSTMs incur higher computational cost and longer training times compared to convolutional models, which may further constrain their practical applicability in large-scale experiments.

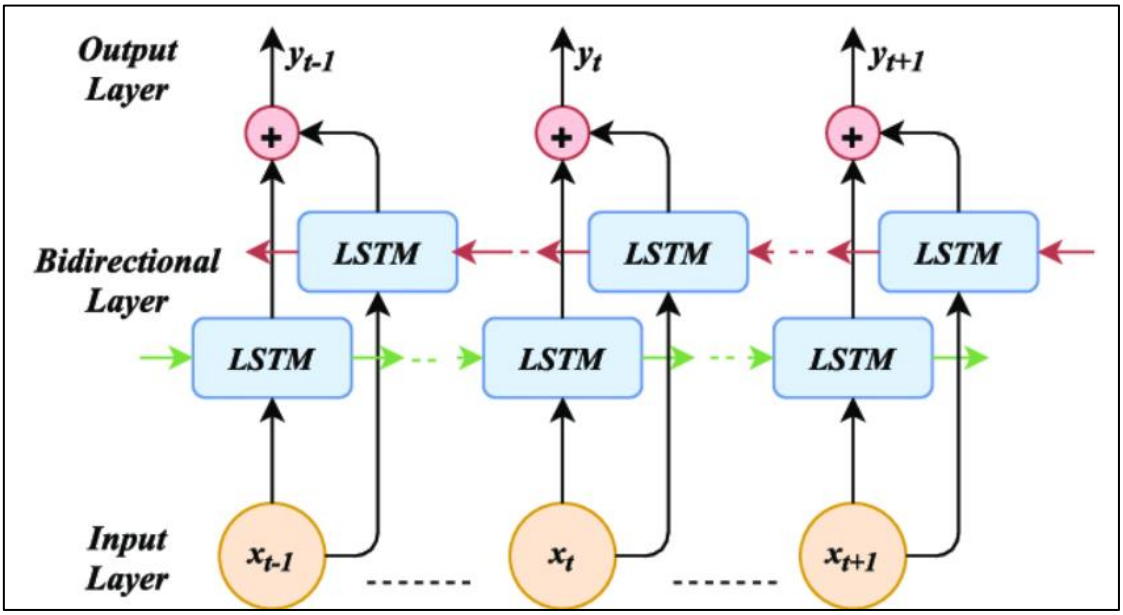

Figure 6. Architecture of a Bidirectional LSTM network processing a text sequence in both forward and backward directions

2.6.3. Transformer-Based Models and DistilBERT

Transformer architectures constitute a major advancement in natural language processing by eliminating recurrent computation and relying exclusively on self-attention mechanisms. Unlike recurrent and convolutional models, Transformers process entire input sequences in parallel, enabling the modeling of long-range dependencies without the limitations imposed by sequential processing.

The core component of the Transformer architecture is the self-attention mechanism, which allows each token to attend to all other tokens in the sequence and dynamically weigh their relevance. Formally, scaled dot-product attention is defined as:

$$\textbf{Attention}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \textbf{softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{K}^{\top}}{\sqrt{\boldsymbol{d_k}}}\right)\boldsymbol{V}$$

where $\boldsymbol{Q}$, $\boldsymbol{K}$, and $\boldsymbol{V}$ denote the query, key, and value matrices, respectively, and $\boldsymbol{d_k}$ is the dimensionality of the key vectors. Through this mechanism, contextual representations are computed by explicitly modeling relationships between all pairs of tokens in the input.

Building upon the Transformer architecture, Bidirectional Encoder Representations from Transformers (BERT) introduced deeply bidirectional contextual representations by jointly conditioning on both left and right contexts during pretraining. BERT is pretrained on large-scale corpora using masked language modeling and next sentence prediction objectives, enabling it to capture rich syntactic and semantic information transferable to downstream tasks.

DistilBERT is a compressed variant of BERT designed to retain most of its representational capacity while significantly reducing computational cost. Through knowledge distillation, DistilBERT is trained to mimic the behavior of a larger BERT model, resulting in a lighter architecture with fewer parameters and faster inference, while maintaining competitive performance (Sanh et al., 2019). This trade-off makes DistilBERT particularly suitable for experimental settings with limited computational resources.

Transformer based models such as DistilBERT are particularly effective for headline classification because they produce contextualized embeddings that can separate subtle semantic differences even in short texts. To be more specific, their contextualized embeddings enable the capture of subtle semantic cues, pragmatic inconsistencies, and lexical interactions that are difficult to model using traditional feature-based approaches or shallow neural architectures. Empirical studies consistently report that BERT-derived models outperform CNN and RNN-based architectures in content-only fake news classification tasks, especially when linguistic distinctions are nuanced and context-dependent (Reis et al., 2019; Shu et al., 2017).

Nevertheless, despite their strong representational power, Transformer-based models are not without limitations. Their performance gains tend to saturate in short-text datasets, where limited input length constrains the amount of exploitable context. Additionally, fine-tuning Transformer models on imbalanced datasets remains challenging, as the models may still exhibit bias toward the majority class unless imbalance-aware training strategies are employed.

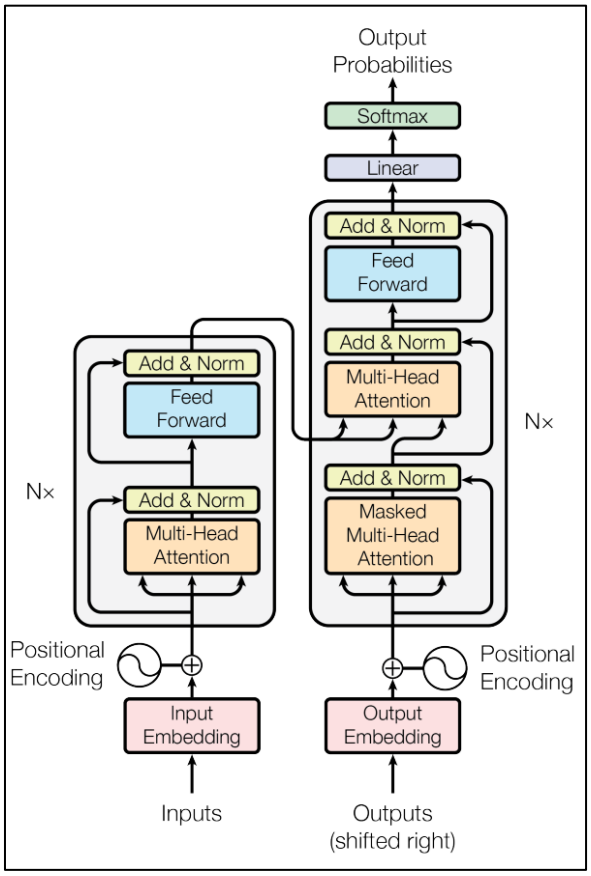

Figure 7. Simplified representation of a Transformer-based architecture for text classification

2.6.4. Hybrid Models: DistilBERT and XGBoost

Hybrid models combine deep learning–based representation learning with traditional machine learning classifiers in a two-stage architecture. In this setting, a pretrained neural model is first employed as a feature extractor, while a separate classifier is responsible for the final decision-making process.

In this thesis, a hybrid architecture combining DistilBERT and XGBoost is investigated. DistilBERT is used to generate contextualized embeddings for each news headline, capturing semantic and syntactic information through self-attention mechanisms. These embeddings are subsequently provided as input to an XGBoost classifier, which performs the final binary classification.

The motivation behind this approach is twofold. First, Transformer-based models excel at extracting rich contextual representations, even from short texts. Second, gradient boosting classifiers such as XGBoost are well-suited for structured, fixed-length feature vectors and are known for their robustness, regularization capabilities, and resistance to overfitting. By decoupling representation learning from classification, hybrid models aim to exploit the strengths of both paradigms.

Prior studies suggest that hybrid architectures may offer improved stability and interpretability compared to fully end-to-end deep learning models, particularly in scenarios where dataset size, class imbalance, or limited linguistic signal constrain neural model performance (Albahar et al., 2021; Rai et al., 2022). However, their effectiveness remains dependent on the quality of the extracted embeddings, and performance gains may be marginal when the underlying dataset provides limited contextual information, as is the case with headline-only fake news detection.

2.6.5. Summary and Limitations of Deep Learning Approaches

The deep learning architectures discussed in this section differ substantially in their representational capacity, computational complexity, and suitability for short-text fake news detection. To facilitate a concise comparison, Table 2 summarizes the primary strengths and limitations of the evaluated deep learning models with respect to headline-based classification tasks.

Model	Strengths	Limitations
CNN	Efficient capture of local n-gram patterns, low computational cost, effective for short texts	Limited ability to model long-range dependencies, insensitive to global context
BiLSTM	Models sequential dependencies and word order; captures bidirectional context	Computationally expensive, limited gains on very short sequences
DistilBERT	Rich contextual representations, strong performance on subtle linguistic patterns pretrained knowledge	High computational cost, performance saturation on headline-only data
DistilBERT + XGBoost	Combines contextual embeddings with robust classification, improved stability	Dependent on embedding quality , added architectural complexity

Table 2. Strengths and limitations of the evaluated deep learning models with respect to headline-based classification tasks

2.7. Limitations of Headline-Only Datasets and Implications for Model Performance

Despite the wide adoption of Machine Learning and Deep Learning approaches in fake news detection, a recurrent limitation identified in the literature concerns the use of headline-only datasets. Although such datasets offer practical advantages in terms of availability, reduced preprocessing complexity, and lower computational cost, they inherently restrict the amount of semantic, contextual, and discourse-level information available to learning algorithms (Shu et al., 2017; Zhou and Zafarani, 2020).

News headlines are intentionally brief, attention-oriented, and often stylistically compressed. As a result, they frequently omit background information, narrative structure, and explanatory context that may be useful for credibility assessment. In headline-based settings, the distinction between fake and real news may therefore depend on subtle lexical or pragmatic cues rather than explicit factual content, which increases the difficulty of the classification task (Wang, 2017; Shu et al., 2017).

This limitation is particularly important in fake news detection on social media, where prior research has shown that news content alone is often insufficient for robust classification. Several studies emphasize that social context, user engagement, and propagation dynamics provide complementary signals that may substantially improve detection performance beyond textual content alone (Shu et al., 2017; Shu et al., 2019). In this respect, headline-only datasets represent a more constrained but also methodologically controlled setting, since they isolate textual evidence from the broader ecosystem in which misinformation spreads.

From a modeling perspective, the restricted informational content of headlines may affect different classes of models in different ways. Traditional Machine Learning methods often rely on surface-level lexical or statistical regularities, which can be weak in short-text settings. Deep Learning models, including recurrent and transformer-based architectures, are generally better equipped to capture latent semantic patterns, but their performance may still be constrained when the available input lacks sufficient context. As a result, architectural

improvements do not necessarily translate into large performance gains in highly compressed textual settings (Khan et al., 2021; Rai et al., 2022).

Another important implication concerns class imbalance. In imbalanced fake news datasets, global metrics such as accuracy may overstate model quality because strong performance on the majority class can conceal weak minority-class detection. For this reason, evaluation in such settings must extend beyond accuracy and include class-sensitive measures such as precision, recall, F1-score, ROC-AUC, and Cohen's Kappa, which provide a more reliable picture of classification behavior under skewed class distributions (He and Garcia, 2009).

Overall, the literature suggests that headline-only fake news detection should be regarded as a constrained classification setting in which both feature richness and contextual depth are inherently limited. This does not reduce its practical relevance, but it does imply that the interpretation of model performance must remain closely tied to the restricted informational conditions under which the models operate (Shu et al., 2017; Shu et al., 2019; Zhou and Zafarani, 2020).

3. Dataset Description and Preprocessing

3.1. Dataset Overview

The experimental analysis conducted in this thesis is based on the **FakeNewsNet** dataset, a large-scale benchmark designed to support research on automated fake news detection. FakeNewsNet integrates verified news content with supplementary information related to news dissemination and user engagement, enabling the study of misinformation under different modeling assumptions (Shu et al., 2018).

Within FakeNewsNet, this study focuses exclusively on the **GossipCop** subset. GossipCop is a fact-checking platform specializing in entertainment-related news, including celebrity gossip, viral rumors, and sensational online stories. Each news item is manually verified by professional fact-checkers and labeled as either *fake* or *real*, providing a reliable ground truth for supervised learning tasks (Shu et al., 2018).

The selection of GossipCop is motivated by its relevance to headline-based fake news detection. Entertainment news frequently relies on short, attention-grabbing headlines, making this subset particularly suitable for evaluating text-based models under constrained informational settings.

3.1.1. FakeNewsNet Dataset

FakeNewsNet provides multiple data modalities, including:

- news textual content,
- social engagement signals,
- user profile information,
- temporal propagation patterns.

Although these modalities enable the exploration of social-context-aware detection approaches, the present study deliberately restricts the analysis to textual information derived from news headlines only. This design choice reflects realistic constraints encountered in

practical applications, where access to social metadata may be limited due to privacy considerations, API restrictions, or platform policies (Zhou and Zafarani, 2020).

3.1.2. GossipCop Subset

The GossipCop subset contains over 20,000 labeled news instances, each associated with a headline and a binary class label (*fake* or *real*). The dataset exhibits a pronounced class imbalance, with real news constituting the majority class. Such imbalance is commonly observed in real-world misinformation datasets and introduces additional challenges for supervised classification models (Shu et al., 2017).

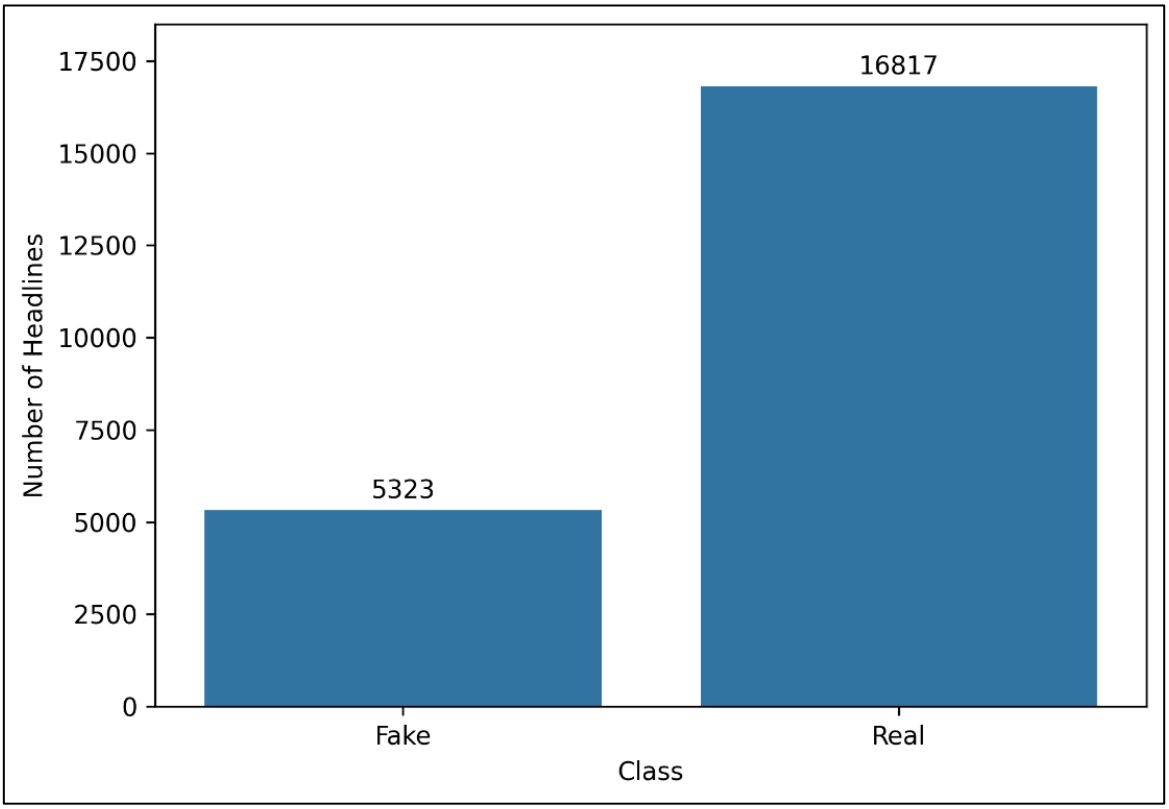

Figure 8. Class distribution of fake and real news headlines in the GossipCop subset

In this thesis, only headline text and corresponding labels are utilized. Full article content, social interactions, and propagation graphs are intentionally excluded to ensure a consistent and controlled experimental setting across all evaluated models.

3.2. Data Characteristics and Class Distribution

The GossipCop subset used in this study contains 22,140 news instances labeled as either fake or real. As shown in the class distribution analysis, the dataset is notably imbalanced, with real news instances substantially outnumbering fake news instances. This imbalance is an important characteristic of the dataset and was taken into account throughout the experimental design, particularly in the selection of validation strategies, imbalance-handling methods, and evaluation metrics.

In addition to class imbalance, the dataset is characterized by the use of short headline text as the primary input for classification. Since the present study deliberately restricts the analysis to headline content, each instance contains limited textual material compared with full news articles or socially enriched fake news datasets. This property defines the scope of the experimental setting adopted in the thesis and motivates the use of both traditional Machine Learning and more expressive Deep Learning models for comparative evaluation.

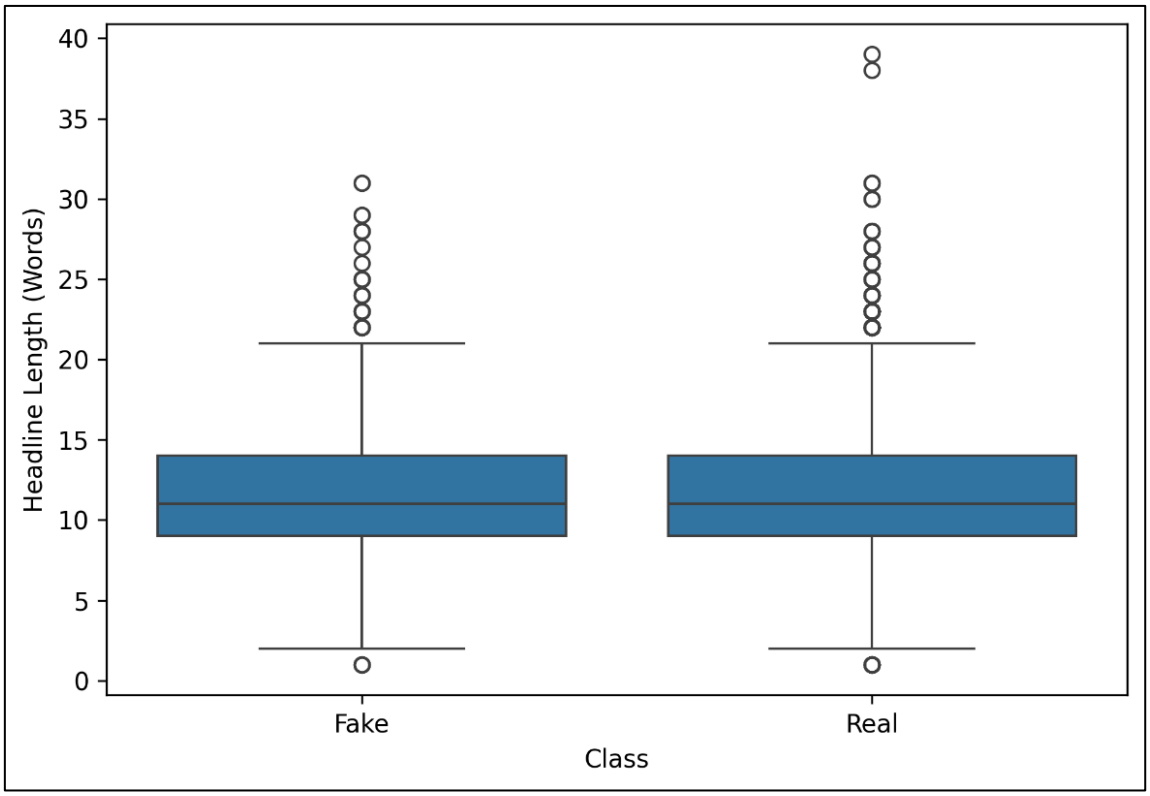

Figure 9. Distribution of headline lengths in the GossipCop subset, measured in number of words for fake and real news headlines

These properties impose structural constraints on automated detection systems and significantly influence both model behavior and evaluation outcomes. The impact of these characteristics is examined in detail in the experimental results and discussion chapters.

3.3. Text Preprocessing Pipeline

Preprocessing plays a critical role in preparing textual data for machine learning and deep learning models, particularly in headline-based fake news detection, where the available linguistic signal is inherently limited. In this study, preprocessing was implemented as a modular and reproducible script-based pipeline, allowing the controlled transformation of the raw GossipCop files into structured inputs suitable for downstream representation learning and classification.

The workflow was organized into a sequence of dedicated scripts, each responsible for a specific transformation stage, including dataset consolidation, tokenization, lexical filtering, and document embedding preparation. This design improved transparency, reproducibility, and consistency across experiments.

3.3.1. Dataset Consolidation

As an initial step, fake and real news instances from the GossipCop subset were merged into a unified dataset containing headline text and corresponding binary labels. Only headline titles were retained for further processing, in line with the experimental focus of this thesis. Full article content and auxiliary metadata were intentionally excluded in order to maintain a consistent input format across all evaluated models.

3.3.2. Text Cleaning and Normalization

All headlines were subjected to basic normalization procedures aimed at reducing noise while preserving informative lexical patterns. These operations included the conversion of all characters to lowercase, the removal of punctuation and non-alphanumeric characters, and the normalization of whitespace. Such preprocessing steps are widely adopted in short-text classification tasks, as they reduce vocabulary fragmentation and improve model robustness without imposing aggressive linguistic constraints (Manning et al., 2008).

3.3.3. Tokenization and Lexical Filtering

Following normalization, headlines were tokenized at the word level. Subsequently, several filtering operations were applied in order to improve the quality of the tokenized text. These included the removal of common English stopwords, the exclusion of purely numeric tokens, and the removal of extremely short tokens with limited semantic content. These filtering steps were designed to reduce noise while retaining semantically meaningful units,

which is particularly important in the context of short headlines, where excessive preprocessing may lead to information loss.

3.3.4. Preparation for Document Embedding

The cleaned and filtered token sequences were subsequently prepared for document-level embedding. Each headline was treated as an independent document and used as input to the Doc2Vec models. This stage produced structured token sequences suitable for learning dense document representations in the following phase of the pipeline.

Figure 9 summarizes the preprocessing workflow used to transform the raw GossipCop files into the final Doc2Vec-based feature representation employed by the Machine Learning models.

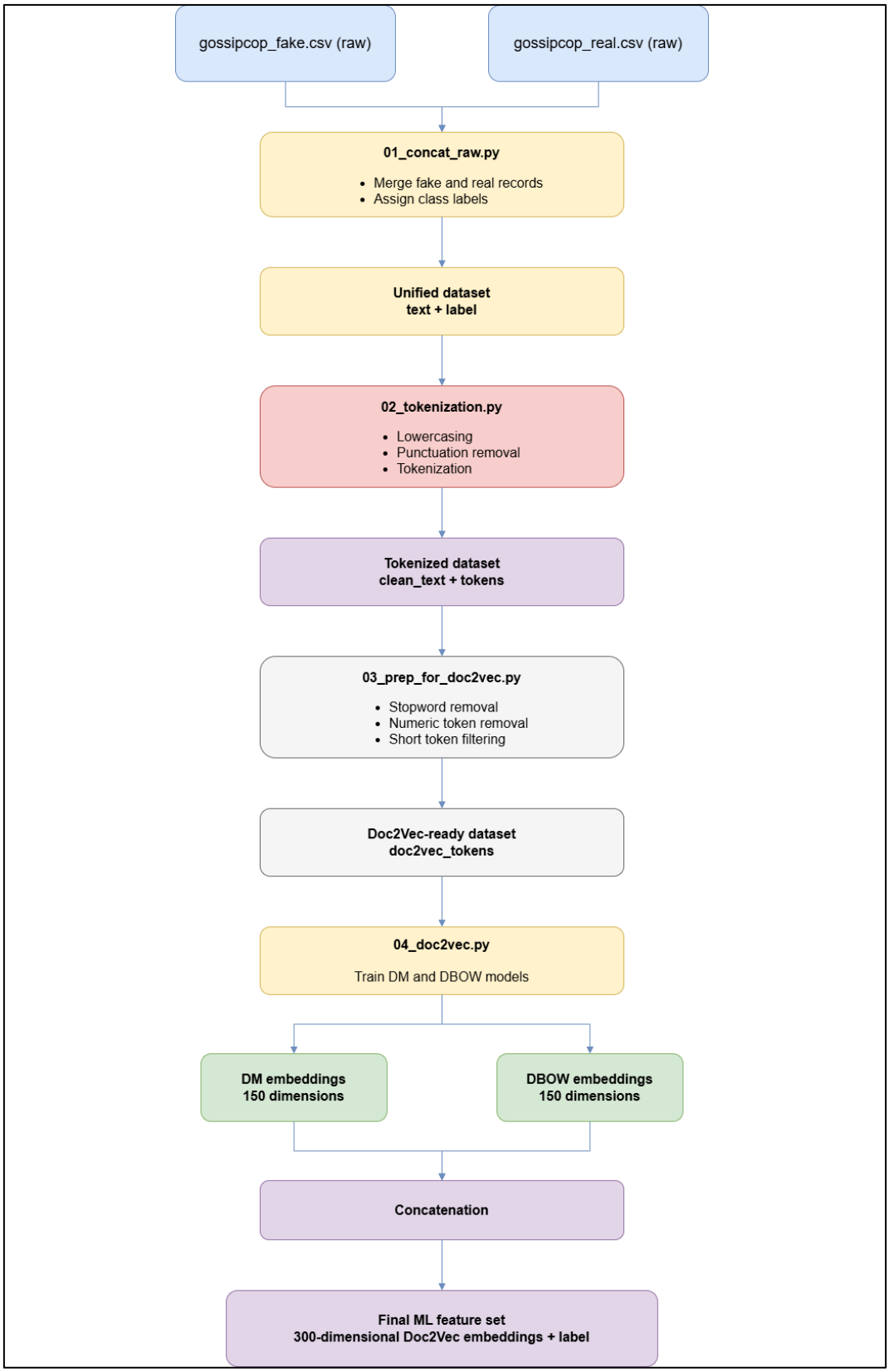

Figure 10. Preprocessing pipeline used to construct the final 300-dimensional Doc2Vec representation from the raw GossipCop fake and real headline files

3.4. Preprocessing for Machine Learning Models

3.4.1. Feature Representation Using Doc2Vec

Machine learning classifiers require fixed-length numerical representations as input. In this study, headline texts were encoded using Doc2Vec embeddings, which provide dense document-level vector representations that capture semantic relationships beyond individual word occurrences.

Doc2Vec was selected after exploratory experimentation with alternative representations, including TF–IDF. While TF–IDF yielded competitive baseline performance, its high dimensionality and sparsity resulted in substantial memory consumption and reduced stability during cross-validation and ensemble training. In contrast, Doc2Vec offers a compact and dense representation that is more suitable for large-scale experimentation under realistic hardware constraints (Le and Mikolov, 2014).

Two complementary Doc2Vec architectures were employed: Distributed Memory (DM) and Distributed Bag of Words (DBOW). As illustrated in Figure X, the final document representation for each headline was obtained by concatenating the vectors produced by both models, resulting in a fixed-length 300-dimensional representation.

3.4.2. Dimensionality and Computational Considerations

The dimensionality of the document embeddings was selected to balance representational capacity and computational efficiency. This choice enabled stable training across all machine learning classifiers evaluated in this study, including Logistic Regression, Support Vector Machines, Random Forests, and XGBoost, without the memory limitations encountered with sparse representations.

No additional handcrafted linguistic features were included in the final machine learning feature set, as exploratory experiments indicated limited performance gains in the headline-only setting.

3.5. Preprocessing for Deep Learning Models

In contrast to the Machine Learning models, which required fixed-length numerical vectors as input, the Deep Learning models used in this study operated directly on textual headline representations derived from tokenization procedures adapted to each architecture.

For the CNN and BiLSTM models, the headline text was converted into tokenized word sequences and padded to a fixed length in order to produce uniform input shapes for training. This representation preserved the sequential structure of the text while remaining compatible with neural architectures designed for short-text classification.

For DistilBERT, preprocessing followed a transformer-based tokenization approach using the corresponding pretrained tokenizer. In this setting, the input headlines were converted into transformer-compatible token sequences with attention masks, allowing the model to process contextualized textual representations rather than simple word-index sequences.

Finally, in the hybrid DistilBERT + XGBoost configuration, DistilBERT was first used as a feature extractor to produce fixed-length contextual embeddings from raw headline text. These embeddings were then provided as input to the XGBoost classifier, combining transformer-based semantic representation with gradient-boosting classification.

3.6. Summary

This chapter presented the dataset selection and preprocessing procedures employed in this study. By focusing exclusively on headline-based textual information from the GossipCop subset of FakeNewsNet, the experimental design emphasizes the challenges of fake news detection under limited contextual information. The methodological implications of these design choices are reflected in the experimental results and further analyzed in the discussion chapter.

Detailed implementation aspects of the preprocessing pipeline are provided in the Appendix A.

4. Methodology and Experimental Setup

4.1. Experimental Design Overview

This chapter presents the methodological framework and experimental design adopted in this thesis for the evaluation of machine learning and deep learning models applied to headline-based fake news detection. The experimental setup is designed to ensure a fair and systematic comparison between different modeling paradigms under consistent conditions, while explicitly accounting for the limitations imposed by short-text data and pronounced class imbalance.

The overall workflow consists of data partitioning, model training, performance evaluation using multiple complementary metrics, and comparative analysis across models. Several methodological decisions were informed not only by established practices reported in the literature (Shu et al., 2018; Zhou and Zafarani, 2020), but also by exploratory experiments conducted during the early stages of this study. These preliminary experiments were instrumental in identifying configurations that were empirically stable and computationally feasible.

4.2. Data Splitting Strategies

Different data splitting strategies were employed for classical machine learning models and deep learning architectures. This distinction reflects both methodological considerations and practical constraints associated with model complexity and training cost.

4.2.1. Machine Learning Models

For traditional machine learning classifiers, model evaluation was performed using stratified 10-fold cross-validation. Stratification ensures that the proportion of fake and real news instances remains consistent across all folds, which is particularly important given the imbalance inherent in the GossipCop dataset.

Cross-validation was selected to obtain robust performance estimates and to reduce sensitivity to a specific train–test split. This evaluation strategy has been widely adopted in prior fake news detection studies involving imbalanced datasets and relatively limited textual input (Shu et al., 2017; Shu et al., 2018).

4.2.2. Deep Learning Models

For deep learning models, a hold-out validation strategy was adopted, consisting of 70% training data, 15% validation data, and 15% test data. This configuration supports hyperparameter tuning and early stopping based on validation performance, which are essential mechanisms for controlling overfitting in neural networks.

Cross-validation was not applied in the deep learning setting due to its high computational cost and limited empirical benefit observed during preliminary experiments. This choice aligns with common practice in deep learning–based fake news detection, where fixed validation splits are typically preferred (Zhou and Zafarani, 2020).

4.3. Handling Class Imbalance

Class imbalance represents a central challenge in the GossipCop dataset, as real news instances significantly outnumber fake news samples. To investigate the impact of imbalance on model performance, different mitigation strategies were applied depending on the model category.

4.3.1. Oversampling for Machine Learning Models

To address the imbalance between fake and real news instances, the Machine Learning models were trained using SMOTE oversampling applied exclusively to the training portion of each fold during cross-validation. This design was adopted in order to avoid data leakage and to ensure that the held-out fold preserved the original class distribution during evaluation.

Although SMOTE is a widely used technique for imbalanced classification, its contribution must be interpreted cautiously in the present setting. In headline-only fake news detection, the limited semantic richness of the input may reduce the effectiveness of synthetic oversampling, since newly generated minority-class samples are still derived from sparse and highly compressed textual representations. For this reason, SMOTE was treated as an imbalance-mitigation strategy within the experimental design rather than as a guaranteed source of performance improvement.

Accordingly, the role of SMOTE in this study was to support a fairer training process for the Machine Learning models under class imbalance, while its actual empirical impact was assessed through the reported evaluation metrics rather than assumed in advance.

4.3.2. Class Weighting for Deep Learning Models

For deep learning models, class imbalance was handled through class weighting rather than data-level oversampling. Class weights were computed based on inverse class frequencies and incorporated directly into the loss function during training.

This approach penalizes misclassification of the minority class more heavily without modifying the original data distribution. In practice, class weighting resulted in more stable training behavior and more consistent improvements in minority-class recall compared to oversampling methods.

4.4. Evaluated Models

The experimental evaluation includes both traditional machine learning classifiers and deep learning architectures, enabling a direct comparison between shallow models and representation-learning approaches.

4.4.1. Machine Learning Models

The following machine learning models were evaluated:

- Logistic Regression
- Support Vector Machines (SVM)
- Random Forest
- XGBoost

All machine learning models were trained using dense document representations derived from Doc2Vec embeddings. Hyperparameters were selected based on commonly adopted configurations in the literature and adjusted through limited experimentation to ensure stable convergence. Extensive hyperparameter tuning was intentionally avoided, as early experiments indicated diminishing returns and limited performance improvements.

4.4.2. Deep Learning Models

The deep learning models evaluated in this study include:

- Convolutional Neural Networks (CNN)
- Bidirectional Long Short-Term Memory networks (BiLSTM)
- DistilBERT
- A hybrid DistilBERT + XGBoost model

CNN and BiLSTM architectures were trained on word-level tokenized headline sequences, allowing the models to learn task-specific representations directly from the data. DistilBERT leveraged pretrained contextual embeddings, enabling the exploitation of linguistic knowledge acquired during large-scale language modeling. The hybrid model combines transformer-based embeddings with a gradient boosting classifier to examine whether discriminative learners can further exploit contextual representations.

4.5. Evaluation Metrics

To obtain a comprehensive assessment of model performance under class imbalance, multiple complementary evaluation metrics were employed.

4.5.1. Accuracy

Accuracy measures the proportion of correctly classified instances:

$$\textbf{Accuracy} = \frac{\boldsymbol{TP + TN}}{\boldsymbol{TP + TN + FP + FN}}$$

While accuracy provides a general overview of performance, it may be misleading in imbalanced datasets, as high accuracy can be achieved by favoring the majority class.

4.5.2. Precision

Precision quantifies the proportion of correctly predicted instances among all instances predicted as a given class:

$$\textbf{Precision} = \frac{\boldsymbol{TP}}{\boldsymbol{TP + FP}}$$

In fake news detection, precision for the fake class reflects the reliability of fake-news predictions.

4.5.3. Recall

Recall measures the proportion of correctly identified instances among all actual instances of a class:

$$\textbf{Recall} = \frac{\boldsymbol{TP}}{\boldsymbol{TP} + \boldsymbol{FN}}$$

Recall is particularly important for fake news detection, as it captures the model's ability to identify misinformation that would otherwise remain undetected.

4.5.4. F1-score

The F1-score is the harmonic mean of precision and recall:

$$\textbf{F1} = 2 \cdot \frac{\boldsymbol{Precision} \cdot \boldsymbol{Recall}}{\boldsymbol{Precision} + \boldsymbol{Recall}}$$

This metric provides a balanced measure of performance, especially in imbalanced classification settings.

4.5.5. ROC–AUC

The Receiver Operating Characteristic (ROC) curve illustrates the trade-off between true positive rate and false positive rate across decision thresholds. The Area Under the Curve (AUC) provides a threshold-independent measure of class separability.

4.5.6. Cohen's Kappa

Cohen's Kappa measures agreement between predicted and true labels while accounting for chance agreement:

$$\boldsymbol{\kappa} = \frac{\boldsymbol{p_o} - \boldsymbol{p_e}}{\mathbf{1} - \boldsymbol{p_e}}$$

Low Kappa values indicate that a model's predictions may not substantially outperform random classification, even when accuracy appears acceptable.

4.6. Implementation Environment and Reproducibility

All experiments were implemented in Python using established libraries, including scikit-learn, XGBoost, TensorFlow/Keras, PyTorch, and the Hugging Face Transformers framework. The preprocessing pipeline, together with the classical Machine Learning experiments based on Logistic Regression, LinearSVC, and Random Forest, was developed and executed in Visual Studio Code (VS Code). In contrast, XGBoost and all Deep Learning experiments, including CNN, BiLSTM, DistilBERT, and the hybrid DistilBERT + XGBoost model, were conducted in Google Colab. This decision was made for practical and computational reasons, as the Colab environment provided access to GPU acceleration, specifically NVIDIA T4 resources, which were particularly useful for transformer-based and other computationally heavier experiments. To enhance reproducibility, fixed random seeds were used wherever applicable. Detailed preprocessing scripts and exploratory configurations are provided in the Appendix.

4.7. Summary

This chapter described the experimental methodology, validation strategies, imbalance-handling techniques, and evaluation metrics adopted in this thesis. The methodological framework supports a fair and informative comparison between machine learning and deep learning approaches for headline-based fake news detection. The following chapter presents the experimental results obtained from the evaluated models.

5. Experimental Results

5.1. Overview of the Evaluation Results

This chapter presents the experimental results obtained from the machine learning and deep learning models evaluated in this study. The results are reported using the evaluation metrics defined in Chapter 4, with particular emphasis on class-wise precision, recall, F1-score, and ROC–AUC, given the class imbalance present in the dataset.

Results are organized into two main sections. Section 5.2 reports the performance of traditional machine learning models, while Section 5.3 presents the results of deep learning and hybrid approaches. All reported results correspond to the final configurations selected after exploratory experimentation.

5.2. Machine Learning Models Results

This section reports the results of the traditional machine learning baselines. All classifiers were trained on Doc2Vec headline embeddings and evaluated with stratified 10-fold cross-validation. To address class imbalance, SMOTE oversampling was applied only to the training split of each fold, while the held-out fold remained untouched. The reported metrics were computed after aggregating predictions across all folds, ensuring that each number reflects the overall performance on the full dataset, rather than fold-specific variation.

Since fake news detection models often struggle with class imbalance, emphasis was placed on evaluating performance for the fake news class, in addition to overall accuracy. For each model, results are presented through a confusion matrix and a ROC curve. In addition to overall accuracy, precision, recall, and F1 score for the fake news class are highlighted, as high accuracy can often be achieved by favoring the majority real news class. The configurations for each model, including hyperparameters and evaluation settings, are detailed in the following sections.

5.2.1. Logistic Regression

Logistic Regression was evaluated using the 300-dimensional Doc2Vec headline embeddings within a stratified 10-fold cross-validation framework. To ensure comparable feature scaling, the input vectors were standardized using StandardScaler prior to classification. Class imbalance was handled with SMOTE, applied only to the training portion of each fold. The classifier was configured with the liblinear solver, max_iter = 500, and random_state = 42. Final predictions were aggregated across all folds in order to compute the overall evaluation metrics.

Parameter	Setting
Input representation	300-dimensional Doc2Vec embeddings
Validation strategy	Stratified 10-fold cross-validation
Number of samples	22140
Number of features	300
Fold shuffling	True
Cross-validation random_state	42
Scaling	StandardScaler
Imbalance handling	SMOTE inside each training fold only
SMOTE k_neighbors	5
SMOTE random_state	42
Model	Logistic Regression
LogisticRegression solver	liblinear
LogisticRegression max_iter	500
LogisticRegression random_state	42

Table 3. Experimental configuration of Logistic Regression

Metric	Value
Accuracy	52.1274%
Cohen's Kappa	0.0111
ROC-AUC	0.5113

Table 4. Overall performance of Logistic Regression

Class	Precision	Recall	F1-score
Fake (0)	0.2462	0.4806	0.3255
Real (1)	0.7646	0.5342	0.6290
Weighted Average	0.6400	0.5213	0.5560

Table 5. Class-wise and weighted metrics for Logistic Regression

Actual / Predicted	Fake	Real
Fake	2558	2765
Real	7834	8983

Table 6. Confusion matrix for Logistic Regression aggregated across folds

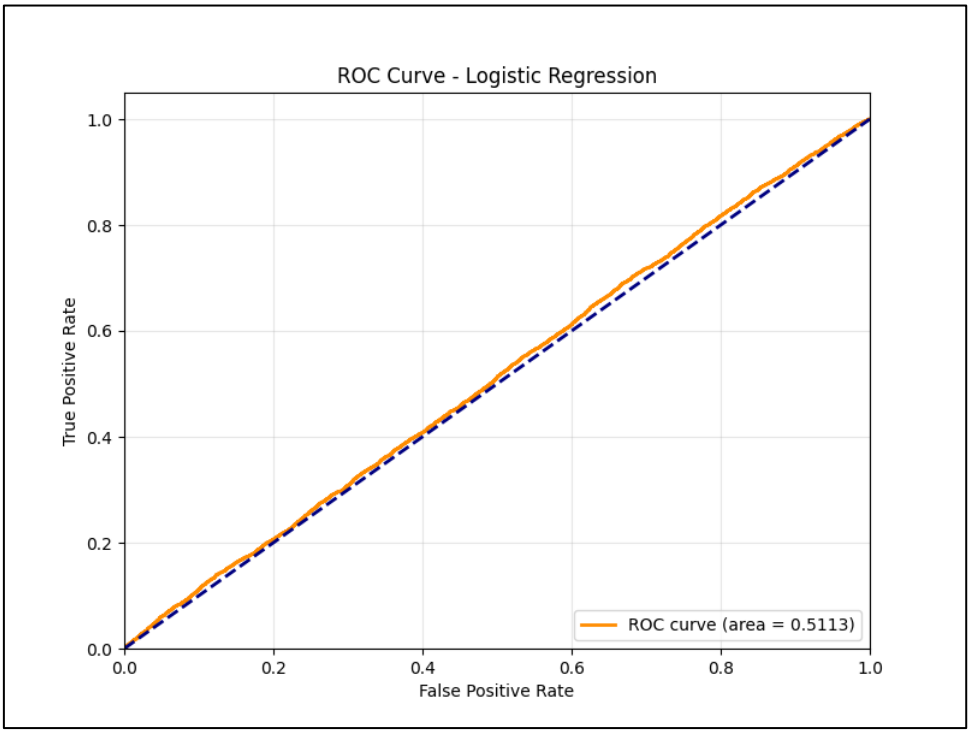

Figure 11. ROC curve for the Logistic Regression classifier on the GossipCop dataset

5.2.2. Linear SVM

Using the 300-dimensional Doc2Vec representation, LinearSVC was evaluated within a stratified 10-fold cross-validation framework. Feature standardization was applied using StandardScaler prior to classification, while class imbalance was handled with SMOTE applied only to the training portion of each fold. The classifier was configured with C = 1.0, max_iter = 2000, and random_state = 42. Since LinearSVC does not provide probability estimates by default, decision function scores were used for ROC-AUC computation. Final predictions and decision scores were aggregated across all folds in order to compute the overall evaluation metrics.

Parameter	Setting
Input representation	300-dimensional Doc2Vec embeddings
Validation strategy	Stratified 10-fold cross-validation
Number of samples	22140
Number of features	300
Fold shuffling	True
Cross-validation random_state	42
Scaling	StandardScaler
Imbalance handling	SMOTE inside each training fold only
SMOTE k_neighbors	5
SMOTE random_state	42
Model	LinearSVC
LinearSVC C	1.0
LinearSVC max_iter	2000
LinearSVC random_state	42

Table 7. Experimental configuration of LinearSVC

Metric	Value
Accuracy	52.1229%
Cohen’s Kappa	0.0112
ROC-AUC	0.5113

Table 8. Overall performance of LinearSVC

Class	Precision	Recall	F1-score
Fake (0)	0.2462	0.4807	0.3256
Real (1)	0.7647	0.5340	0.6289
Weighted Average	0.6400	0.5212	0.5560

Table 9. Class-wise and weighted metrics for LinearSVC

Actual / Predicted	Fake	Real
Fake	2559	2764
Real	7836	8981

Table 10. Confusion matrix for LinearSVC aggregated across folds

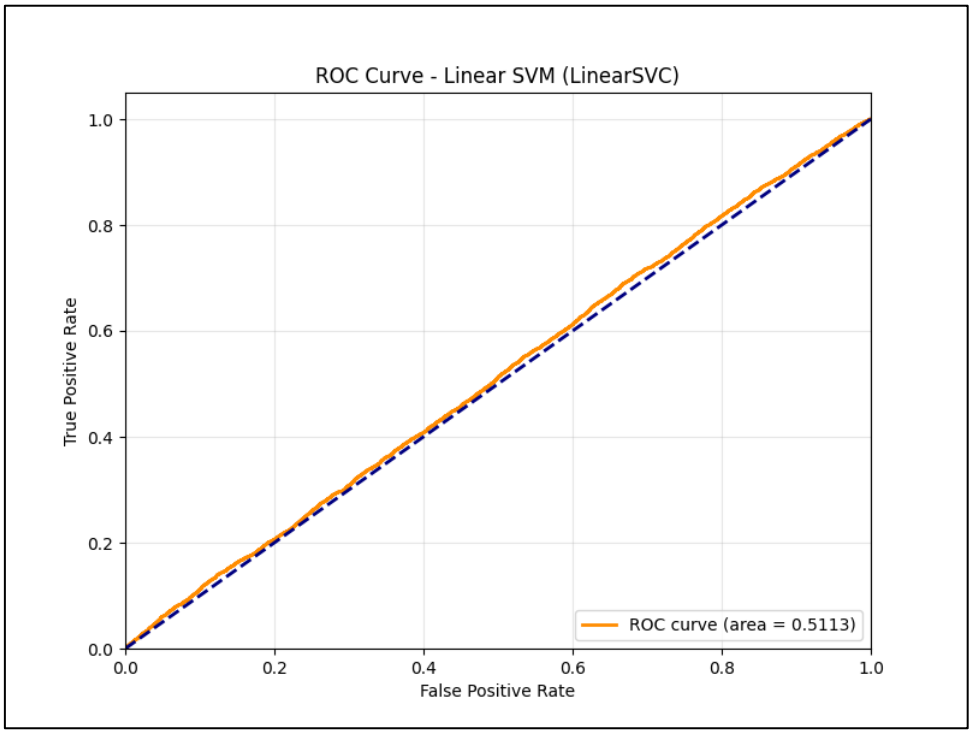

Figure 12. ROC curve for the Linear SVM classifier on the GossipCop dataset

5.2.3. Random Forest

Random Forest was trained on the 300-dimensional Doc2Vec feature space and evaluated using stratified 10-fold cross-validation. Class imbalance was addressed with SMOTE, applied only to the training portion of each fold, while the held-out fold remained unchanged. The classifier was configured with n_estimators = 100, max_depth = None, n_jobs = -1, and random_state = 42. Final predictions and probability scores were aggregated across all folds in order to derive the reported evaluation metrics.

Parameter	Setting
Input representation	300-dimensional Doc2Vec embeddings
Validation strategy	Stratified 10-fold cross-validation
Number of samples	22140
Number of features	300
Fold shuffling	True
Cross-validation random_state	42
Imbalance handling	SMOTE inside each training fold only
SMOTE k_neighbors	5
SMOTE random_state	42
Model	RandomForestClassifier
RandomForest n_estimators	100
RandomForest max_depth	None
n_jobs	-1
RandomForest random_state	42

Table 11. Experimental configuration of Random Forest

Metric	Value
Accuracy	77.5700%
Cohen’s Kappa	0.1596
ROC-AUC	0.5841

Table 12. Overall performance of Random Forest

Class	Precision	Recall	F1-score
Fake (0)	0.6581	0.1396	0.2303
Real (1)	0.7820	0.9770	0.8687
Weighted Average	0.7522	0.7757	0.7152

Table 13. Class-wise and weighted metrics for Random Forest

Actual / Predicted	Fake	Real
Fake	743	4580
Real	386	16431

Table 14. Confusion matrix for Random Forest aggregated across folds

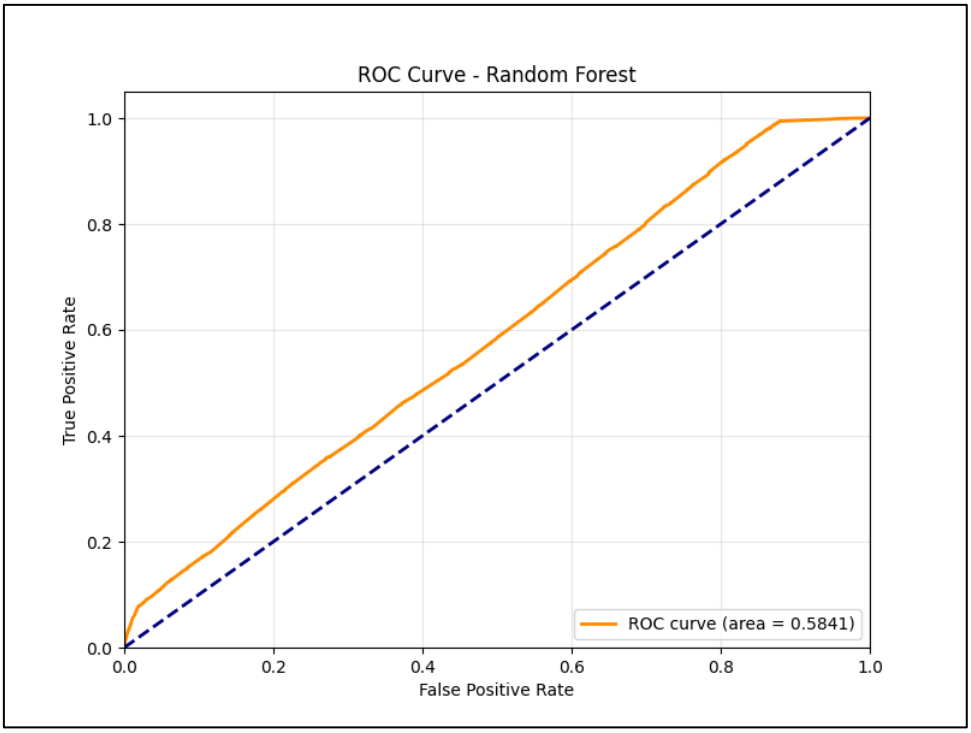

Figure 13. ROC Curve for the Random Forest Classifier on the GossipCop dataset

5.2.4. XGBoost

XGBoost was evaluated on the 300-dimensional Doc2Vec embeddings using stratified 10-fold cross-validation. To address class imbalance, SMOTE oversampling was applied only within the training portion of each fold. The classifier was configured with n_estimators = 300, max_depth = 8, learning_rate = 0.1, subsample = 0.8, colsample_bytree = 0.8, objective = binary:logistic, eval_metric = logloss, tree_method = hist, and random_state = 42. Final predictions and probability scores were aggregated across all folds to compute the overall evaluation metrics.

Parameter	Setting
Input representation	300-dimensional Doc2Vec embeddings
Validation strategy	Stratified 10-fold cross-validation
Number of samples	22140
Number of features	300
Fold shuffling	True
Cross-validation random_state	42
Imbalance handling	SMOTE inside each training fold only
SMOTE k_neighbors	5
SMOTE random_state	42
Model	XGBClassifier
XGBoost n_estimators	300
XGBoost max_depth	8
XGBoost learning_rate	0.1
XGBoost subsample	0.8
XGBoost colsample_bytree	0.8
XGBoost objective	binary:logistic
XGBoost eval_metric	logloss
XGBoost tree_method	hist
XGBoost random_state	42

Table 15. Experimental configuration of XGBoost

Metric	Value
Accuracy	74.9955%
Cohen’s Kappa	0.1371
ROC-AUC	0.5771

Table 16. Overall performance of XGBoost

Class	Precision	Recall	F1-score
Fake (0)	0.4494	0.1777	0.2547
Real (1)	0.7815	0.9311	0.8498
Weighted Average	0.7522	0.7500	0.7067

Table 17. Class-wise and weighted metrics for XGBoost

Actual / Predicted	Fake	Real
Fake	946	4377
Real	1159	15658

Table 18. Confusion matrix for XGBoost aggregated across folds

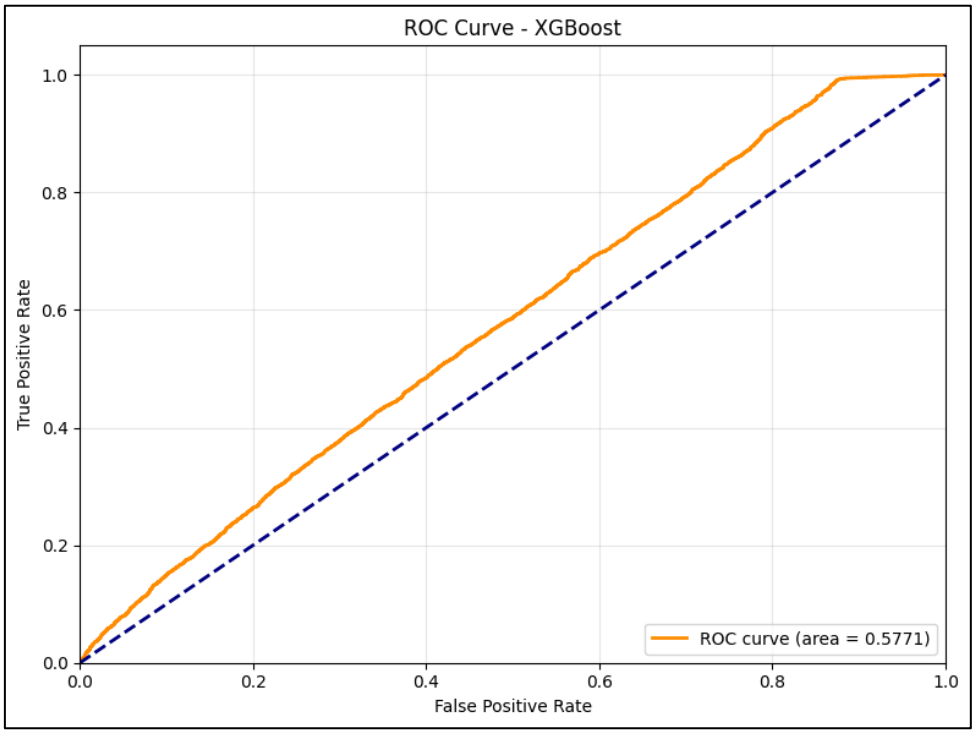

Figure 14. ROC Curve for the XGBoost Classifier on the GossipCop dataset

5.3. Deep Learning Models Results

This section presents the results obtained from the deep learning models evaluated in this study. Unlike the machine learning baselines, these models were trained using a holdout validation scheme with a 70% training set, 15% validation set, and 15% test set. Class imbalance was addressed through class weights computed from the training data, while the held-out validation and test sets remained unchanged. The reported results therefore reflect performance on an unseen test set rather than aggregated outcomes across multiple folds.

The deep learning experiments included sequence-based and transformer-based architectures applied directly to headline text. More specifically, the CNN and BiLSTM models were trained on tokenized and padded headline sequences, whereas DistilBERT and the hybrid DistilBERT + XGBoost model operated on transformer-derived representations. For each model, performance is reported using confusion matrices and ROC curves, together with overall accuracy, Cohen's Kappa, ROC-AUC, and class-wise precision, recall, and F1-score. Particular attention is given to the fake news class, since performance on the minority class provides a more informative view of model effectiveness than accuracy alone. The configuration of each model, including the main architectural and training parameters, is presented in the corresponding subsections.

5.3.1. CNN (Convolutional Neural Networks)

The CNN model was trained on tokenized and padded headline sequences using a holdout validation framework consisting of 70% training, 15% validation, and 15% test data. Class imbalance was addressed through class weights computed from the training set. The model was configured with max_words = 30000, max_len = 50, embedding_dim = 100, filters = 256, kernel_size = 5, dropout rate = 0.5, dense units = 64, epochs = 30, and batch_size = 128. Early stopping was applied with patience = 4. Final predictions and probability scores were computed on the held-out test set in order to derive the reported evaluation metrics.

Parameter	Setting
Input representation	Tokenized and padded headline sequences
Validation strategy	Holdout (70% train, 15% val, 15% test)
Number of samples	22140
Data split random_state	42
Vocabulary size (max_words)	30000
Maximum sequence length (max_len)	50
Embedding dimension	100
Imbalance handling	Class weights computed from the training set
Model	CNN
CNN filters	256
CNN kernel_size	5
CNN dropout rate	0.5
CNN dense units	64
Epochs	30
Batch size	128
EarlyStopping patience	4

Table 19. Experimental configuration of CNN

Metric	Value
Accuracy	81.2406%
Cohen's Kappa	0.5354
ROC-AUC	0.8707

Table 20. Overall performance of CNN

Class	Precision	Recall	F1-score
Fake (0)	0.5837	0.7644	0.6620
Real (1)	0.9174	0.8276	0.8702
Weighted Average	0.8372	0.8124	0.8201

Table 21. Class-wise and weighted metrics for CNN

Actual / Predicted	Fake	Real
Fake	610	188
Real	435	2088

Table 22. Confusion matrix for CNN after Train-Validation-Test split

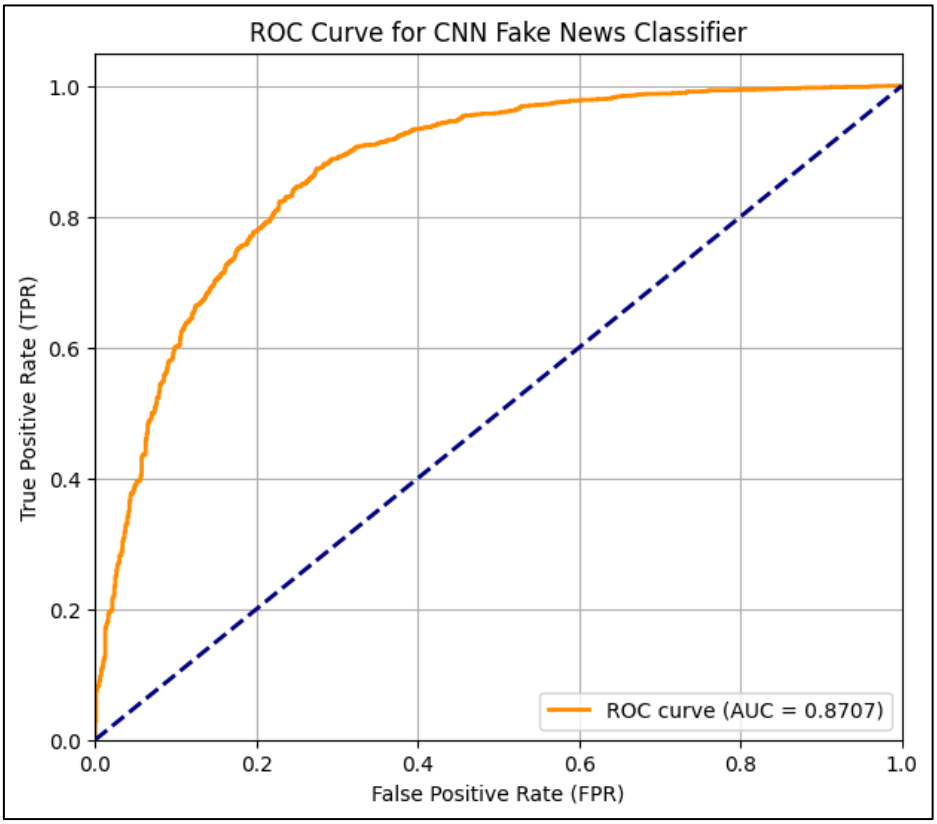

Figure 15. ROC Curve for the CNN on the GossipCop dataset

5.3.2. BiLSTM (Bidirectional Long Short-Term Memory)

For the BiLSTM architecture, the headline text was represented as tokenized and padded word sequences and evaluated using a holdout split of 70% training, 15% validation, and 15% test data. Class imbalance was addressed through class weights computed from the training portion of the dataset. The model used max_words = 20000, max_len = 40, embedding_dim = 100, BiLSTM units = 128, dense units = 64, dropout rate = 0.5, epochs = 12, and batch_size = 64. Early stopping was employed with patience = 3 in order to restore the best model weights according to validation loss. The reported metrics were obtained from predictions on the held-out test set.

Parameter	Setting
Input representation	Tokenized and padded headline sequences
Validation strategy	Holdout (70% train, 15% val, 15% test)
Number of samples	22140
Data split random_state	42
Vocabulary size (max_words)	20000
Maximum sequence length (max_len)	40
Embedding dimension	100
Imbalance handling	Class weights computed from the training set
Model	BiLSTM
BiLSTM units	128
BiLSTM dropout rate	0.5
Dense units	64
Epochs	12
Batch size	64
EarlyStopping patience	3

Table 23. Experimental configuration of BiLSTM

Metric	Value
Accuracy	77.6573%
Cohen's Kappa	0.4821
ROC-AUC	0.8651

Table 24. Overall performance of BiLSTM

Class	Precision	Recall	F1-score
Fake (0)	0.5229	0.8008	0.6327
Real (1)	0.9242	0.7689	0.8395
Weighted Average	0.8278	0.7766	0.7898

Table 25. Class-wise and weighted metrics for BiLSTM

Actual / Predicted	Fake	Real
Fake	639	159
Real	583	1940

Table 26. Confusion matrix for BiLSTM after Train-Validation-Test split

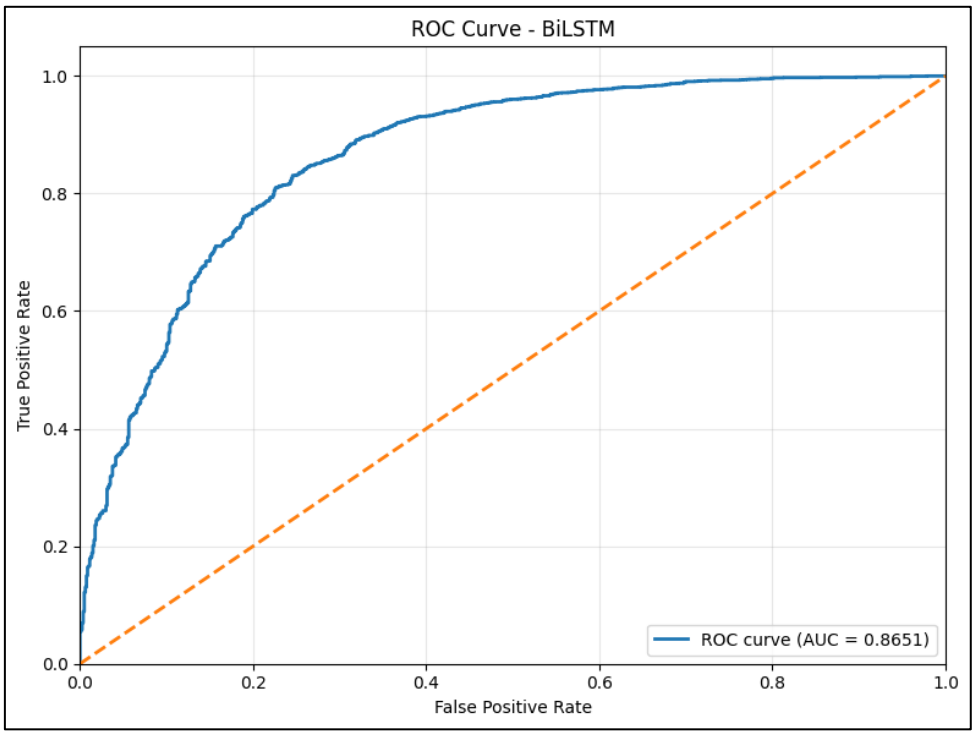

Figure 16. ROC Curve for the BiLSTM on the GossipCop dataset

5.3.3. DistilBERT

DistilBERT was evaluated by fine-tuning a pretrained transformer model on the headline classification task using a holdout validation framework with 70% training, 15% validation, and 15% test data. Class imbalance was handled through class weights derived from the training set. The model was initialized from the distilbert-base-uncased checkpoint and trained with max_length = 128, batch_size = 16, learning rate = 2e-05, CrossEntropyLoss, and 2 epochs. Final predictions and probability scores were computed on the held-out test set to obtain the reported evaluation metrics.

Parameter	Setting
Input representation	Transformer-tokenized headline inputs
Validation strategy	Holdout (70% train, 15% val, 15% test)
Number of samples	22140
Data split random_state	42
Imbalance handling	Class weights computed from the training set
Pretrained model	distilbert-base-uncased
Tokenizer	DistilBertTokenizerFast
Maximum sequence length	128
Model	DistilBERT
Number of labels	2
Optimizer	AdamW
Learning rate	2e-05
Epochs	2
Batch size	16
Loss function	CrossEntropyLoss

Table 27. Experimental configuration of DistilBERT

Metric	Value
Accuracy	82.8365%
Cohen's Kappa	0.5687
ROC-AUC	0.8957

Table 28. Overall performance of DistilBERT

Class	Precision	Recall	F1-score
Fake (0)	0.6131	0.7744	0.6844
Real (1)	0.9222	0.8454	0.8821
Weighted Average	0.8479	0.8284	0.8346

Table 29. Class-wise and weighted metrics for DistilBERT

Actual / Predicted	Fake	Real
Fake	618	180
Real	390	2133

Table 30. Confusion matrix for DistilBERT after Train-Validation-Test split

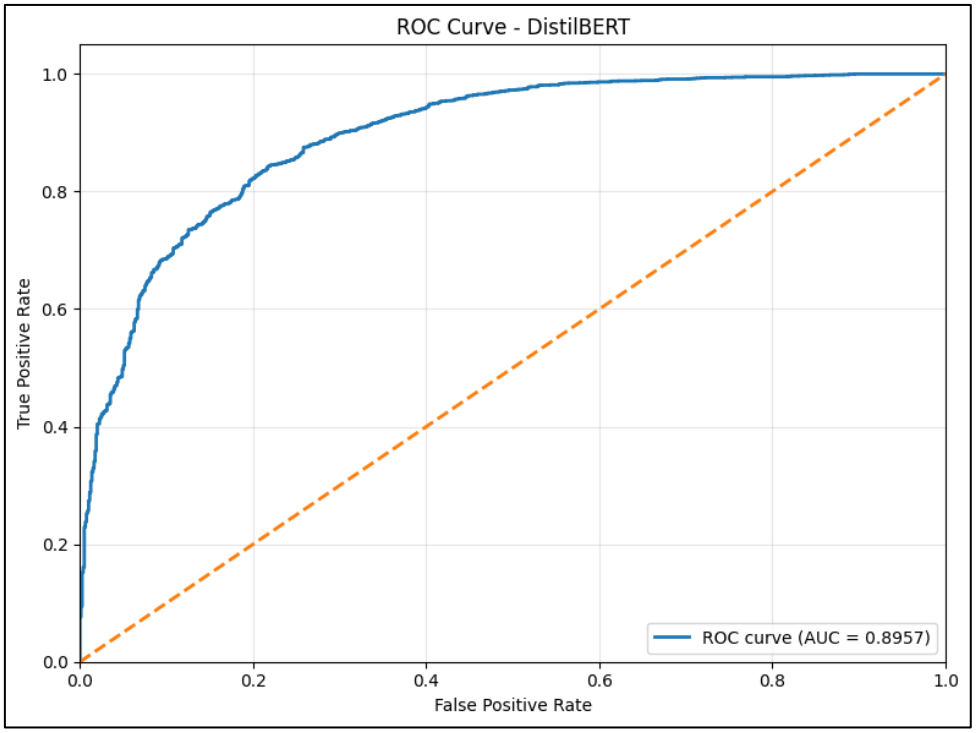

Figure 17. ROC Curve for the DistilBERT on the GossipCop dataset

5.3.4. Hybrid DistilBERT + XGBoost

In the hybrid configuration, raw headlines were first encoded using DistilBERT in order to extract fixed-length contextual embeddings, which were then used as input to an XGBoost classifier. The experiment followed a holdout setup with 70% training, 15% validation, and 15% test data. DistilBERT embeddings were obtained from the distilbert-base-uncased model using the first-token hidden representation, with max_length = 128, embedding batch size = 32, and embedding dimensionality of 768. These representations were subsequently provided to an XGBClassifier configured with n_estimators = 300, max_depth = 6, learning_rate = 0.05, subsample = 0.8, colsample_bytree = 0.8, eval_metric = logloss, and tree_method = hist. No explicit imbalance handling technique was applied in this hybrid setting. The reported results were computed on the held-out test set.

Parameter	Setting
Input representation	DistilBERT embeddings extracted from raw headlines
Validation strategy	Holdout (70% train, 15% val, 15% test)
Number of samples	22140
Data split random_state	42
Imbalance handling	None
Pretrained model	distilbert-base-uncased
Tokenizer	DistilBertTokenizerFast
Embedding extractor	DistilBertModel
Embedding source	last_hidden_state[:, 0, :]
Embedding dimension	768
Maximum sequence length	128
Embedding batch size	32
Model	XGBClassifier
XGBoost n_estimators	300
XGBoost max_depth	6

XGBoost learning_rate	0.05
XGBoost subsample	0.8
XGBoost colsample_bytree	0.8
XGBoost eval_metric	logloss
XGBoost tree_method	hist

Table 31. Experimental configuration of Hybrid DistilBERT + XGBoost

Metric	Value
Accuracy	84.3421%
Cohen's Kappa	0.5020
ROC-AUC	0.8462

Table 32. Overall performance of Hybrid DistilBERT + XGBoost

Class	Precision	Recall	F1-score
Fake (0)	0.7932	0.4712	0.5912
Real (1)	0.8518	0.9612	0.9032
Weighted Average	0.8377	0.8434	0.8282

Table 33. Class-wise and weighted metrics for Hybrid DistilBERT + XGBoost

Actual / Predicted	Fake	Real
Fake	376	422
Real	98	2425

Table 34. Confusion matrix for Hybrid DistilBERT + XGBoost after Train-Validation-Test split

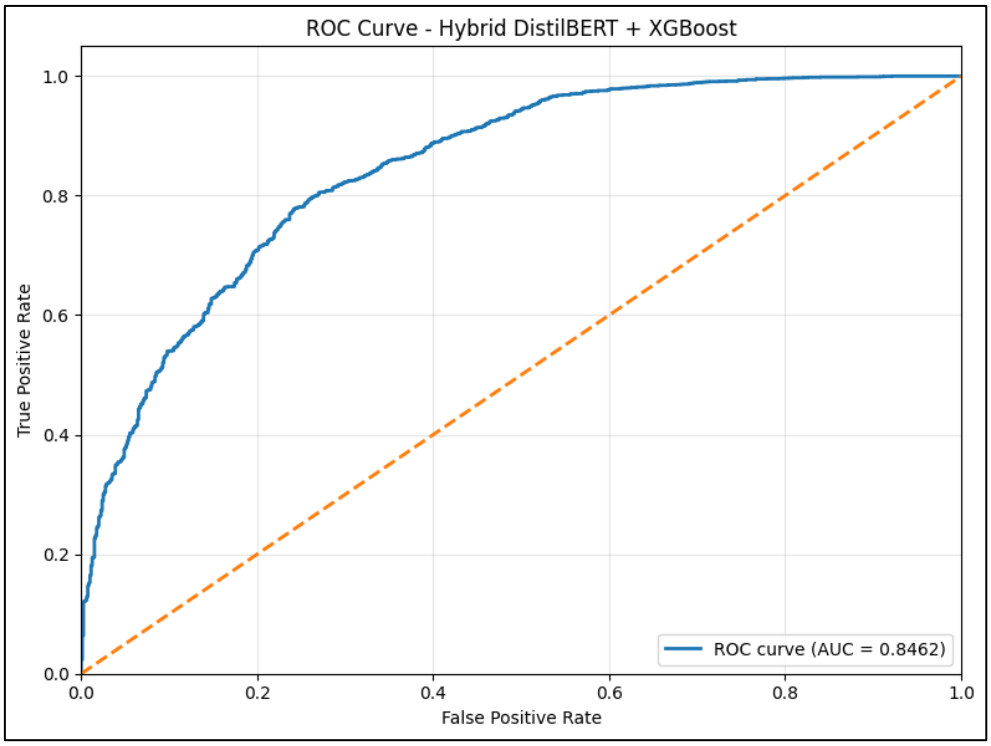

Figure 18. ROC Curve for the DistilBERT + XGBoost on the GossipCop dataset

6. Comparative Evaluation of Models

6.1. Introduction

This chapter presents a comparative evaluation of the models examined in the present study. Whereas Chapter 5 reported the performance of each model separately, the purpose of the current chapter is to compare the evaluated approaches under a common analytical perspective. Particular emphasis is placed on overall performance, discriminative ability, and effectiveness on the fake news class, since the class imbalance of the GossipCop dataset makes minority-class behavior especially important.

The comparison is organized in four stages. First, the Machine Learning models are compared with one another. Next, the Deep Learning models are examined comparatively. This is followed by an overall comparison across all evaluated models. Finally, the findings of this thesis are compared with selected results reported in previous studies on fake news detection.

6.2. Comparative Analysis of Machine Learning Models

Table 35 summarizes the comparative performance of the Machine Learning models evaluated in this study. The analysis focuses on overall accuracy, Cohen's Kappa, ROC-AUC, and fake-class precision, recall, and F1-score, since these metrics provide a more complete view of model behavior under class imbalance than accuracy alone.

Model	Accuracy	Cohen's Kappa	ROC-AUC	Fake Precision	Fake Recall	Fake F1	Weighted F1
Logistic Regression	52.1274%	0.0111	0.5113	0.2462	0.4806	0.3255	0.5560

Model	Accuracy	Cohen's Kappa	ROC-AUC	Fake Precision	Fake Recall	Fake F1	Weighted F1
LinearSVC	52.1229%	0.0112	0.5113	0.2462	0.4807	0.3256	0.5560
Random Forest	77.5700%	0.1596	0.5841	0.6581	0.1396	0.2303	0.7152
XGBoost	74.9955%	0.1371	0.5771	0.4494	0.1777	0.2547	0.7067

Table 35. Comparative performance of the Machine Learning models

The comparison of the Machine Learning models reveals two distinct patterns of behavior. Logistic Regression and LinearSVC produced almost identical results across all reported metrics. Their accuracy remained close to 52%, while Cohen's Kappa was close to zero and ROC-AUC was only slightly above 0.50. This indicates that both linear models had very limited discriminative ability in the Doc2Vec feature space and were only marginally better than random separation of the two classes.

The fake-class results of Logistic Regression and LinearSVC also reflect this weak overall performance. Although both models achieved fake recall values close to 0.48, their fake precision remained very low, resulting in fake-class F1-scores around 0.33. This suggests that the two linear approaches were able to identify a portion of the fake headlines, but did so with a large number of false positive predictions. Their weighted F1 values also remained comparatively low, confirming that their overall predictive quality was limited.

In contrast, the tree-based models, Random Forest and XGBoost, achieved substantially higher overall accuracy than the linear baselines. Random Forest reached the highest accuracy among the Machine Learning models, while XGBoost also showed a clear improvement over the linear approaches. However, this gain in accuracy was not accompanied by equally strong performance on the fake news class. In particular, Random Forest achieved the highest fake precision among the Machine Learning models, but its fake recall was extremely low. This indicates that the model was highly conservative in assigning the fake label, leading to many

missed fake instances. XGBoost showed a slightly more balanced behavior than Random Forest, but its fake recall also remained low and its fake-class F1-score stayed limited.

The ROC-AUC and Cohen's Kappa values further support this pattern. Both tree-based models outperformed the linear baselines, yet their ROC-AUC values remained well below those achieved later by the Deep Learning approaches. Similarly, their Kappa values indicate some improvement in agreement beyond chance, but not at a level suggesting robust and well-balanced discrimination between fake and real headlines.

Overall, the Machine Learning comparison suggests that the linear models were insufficient for the headline-only setting, while the tree-based methods provided stronger global accuracy but remained weak in recovering fake news instances. Among the Machine Learning models, Random Forest and XGBoost achieved the strongest overall results, but neither demonstrated strong minority-class effectiveness when judged by fake recall and fake-class F1-score.

6.3. Comparative Analysis of Deep Learning Models

Table 36 presents the comparative performance of the Deep Learning models evaluated in this study. The comparison includes sequence-based, transformer-based, and hybrid architectures, and focuses on overall accuracy, Cohen's Kappa, ROC-AUC, and the class-wise performance on fake news, in order to assess both general predictive quality and minority-class effectiveness.

Model	Accuracy	Cohen's Kappa	ROC-AUC	Fake Precision	Fake Recall	Fake F1	Weighted F1
CNN	81.2406%	0.5354	0.8707	0.5837	0.7644	0.6620	0.8201
BiLSTM	77.6573%	0.4821	0.8651	0.5229	0.8008	0.6327	0.7898

Model	Accuracy	Cohen's Kappa	ROC-AUC	Fake Precision	Fake Recall	Fake F1	Weighted F1
DistilBERT	82.8365%	0.5687	0.8957	0.6131	0.7744	0.6844	0.8346
Hybrid DistilBERT + XGBoost	84.3421%	0.5020	0.8462	0.7932	0.4712	0.5912	0.8282

Table 36. Comparative performance of the Deep Learning models

The Deep Learning models achieved clearly stronger results than the Machine Learning baselines across almost all evaluation metrics. However, notable differences are also observed among the Deep Learning approaches themselves, particularly with regard to the balance between overall accuracy and fake-class effectiveness.

Among the sequence-based models, CNN and BiLSTM exhibited different strengths. CNN achieved higher overall accuracy and a stronger weighted F1-score than BiLSTM, indicating a more balanced overall predictive behavior across the two classes. At the same time, BiLSTM produced the highest fake recall among all evaluated Deep Learning models, which suggests that it was particularly effective at recovering fake news instances. This stronger sensitivity to the minority class, however, was accompanied by lower fake precision and lower overall accuracy relative to CNN.

DistilBERT produced the strongest overall balance among the Deep Learning models. It achieved the highest ROC-AUC, the highest Cohen's Kappa, and the highest fake-class F1-score, while also maintaining strong overall accuracy and weighted F1. These results indicate that the transformer-based representation was more effective in separating fake and real headlines than the sequence-based architectures, while also preserving a favorable balance between precision and recall for the fake class.

The hybrid DistilBERT + XGBoost model reached the highest overall accuracy among all evaluated Deep Learning models and also achieved the highest fake precision. This indicates that the hybrid configuration was more conservative in assigning the fake label and produced fewer false positive predictions than the other Deep Learning approaches. However, this

advantage was accompanied by a noticeable reduction in fake recall, meaning that a larger proportion of fake instances remained undetected. As a result, its fake-class F1-score remained below that of DistilBERT, CNN, and BiLSTM, despite its strong accuracy.

The ROC-AUC values provide additional evidence for the relative strengths of the Deep Learning models. DistilBERT obtained the highest ROC-AUC, followed closely by BiLSTM and CNN, while the hybrid model achieved a somewhat lower value despite its higher accuracy. This pattern suggests that the hybrid architecture offered stronger threshold-dependent classification performance under the selected decision rule, but weaker overall class separation than the standalone DistilBERT model.

Overall, the comparison among the Deep Learning models shows that each architecture emphasized a different performance trade-off. BiLSTM was strongest in terms of fake recall, the hybrid DistilBERT + XGBoost model achieved the highest accuracy and fake precision, while DistilBERT provided the most balanced overall performance when ROC-AUC, Cohen's Kappa, fake-class F1-score, and weighted F1 are considered jointly.

6.4. Overall Comparison of All Evaluated Models

Table 37 provides an overall comparison of all models examined in this thesis, including both Machine Learning and Deep Learning approaches. Meanwhile, Figure 19 provides a compact visual comparison of all evaluated models across four key performance metrics, highlighting differences in overall predictive performance, discriminative ability, and effectiveness on the fake news class. The comparison combines global performance indicators with fake-class evaluation metrics in order to assess not only general predictive effectiveness, but also the ability of each model to detect the minority fake news class.

Model	Type	Accuracy	Cohen's Kappa	ROC-AUC	Fake Precision	Fake Recall	Fake F1	Weighted F1
Logistic Regression	ML	52.1274%	0.0111	0.5113	0.2462	0.4806	0.3255	0.5560
LinearSVC	ML	52.1229%	0.0112	0.5113	0.2462	0.4807	0.3256	0.5560
Random Forest	ML	77.5700%	0.1596	0.5841	0.6581	0.1396	0.2303	0.7152
XGBoost	ML	74.9955%	0.1371	0.5771	0.4494	0.1777	0.2547	0.7067
CNN	DL	81.2406%	0.5354	0.8707	0.5837	0.7644	0.6620	0.8201
BiLSTM	DL	77.6573%	0.4821	0.8651	0.5229	0.8008	0.6327	0.7898
DistilBERT	DL	82.8365%	0.5687	0.8957	0.6131	0.7744	0.6844	0.8346
Hybrid DistilBERT + XGBoost	DL	84.3421%	0.5020	0.8462	0.7932	0.4712	0.5912	0.8282

Table 37. Overall comparison of all Evaluated Models

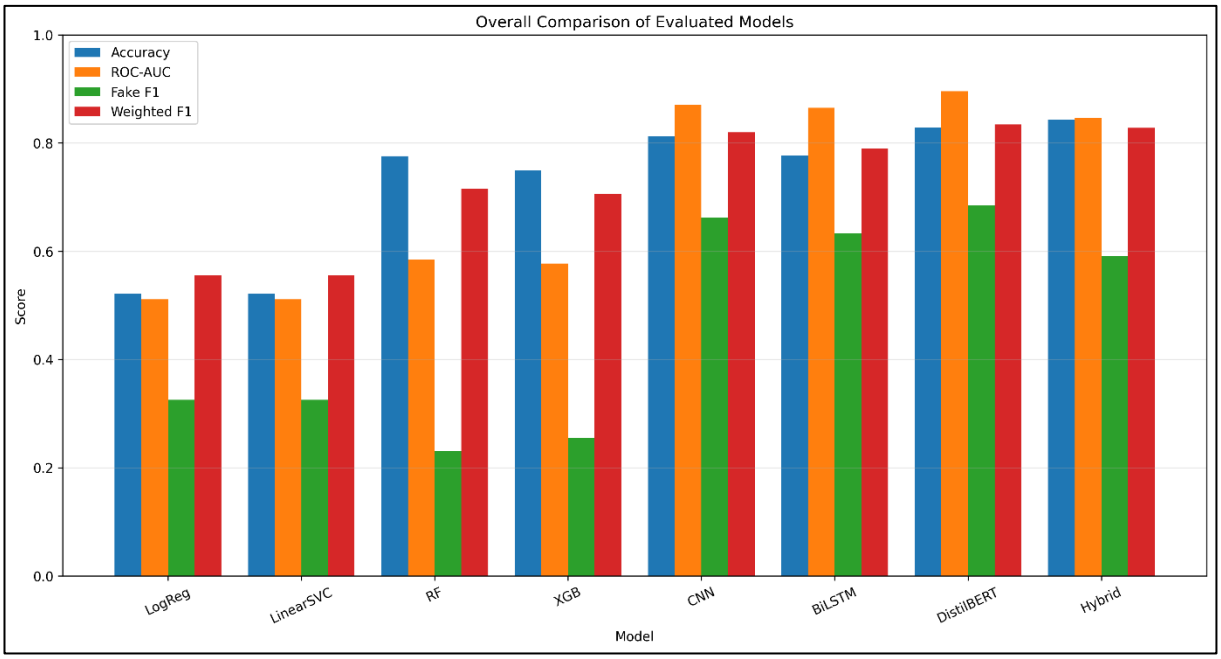

Figure 19. Overall comparison of Evaluated Models

Figure 20 highlights the differences among the evaluated models with respect to fake-class precision, recall, and F1-score, making the trade-offs in minority-class detection more directly visible.

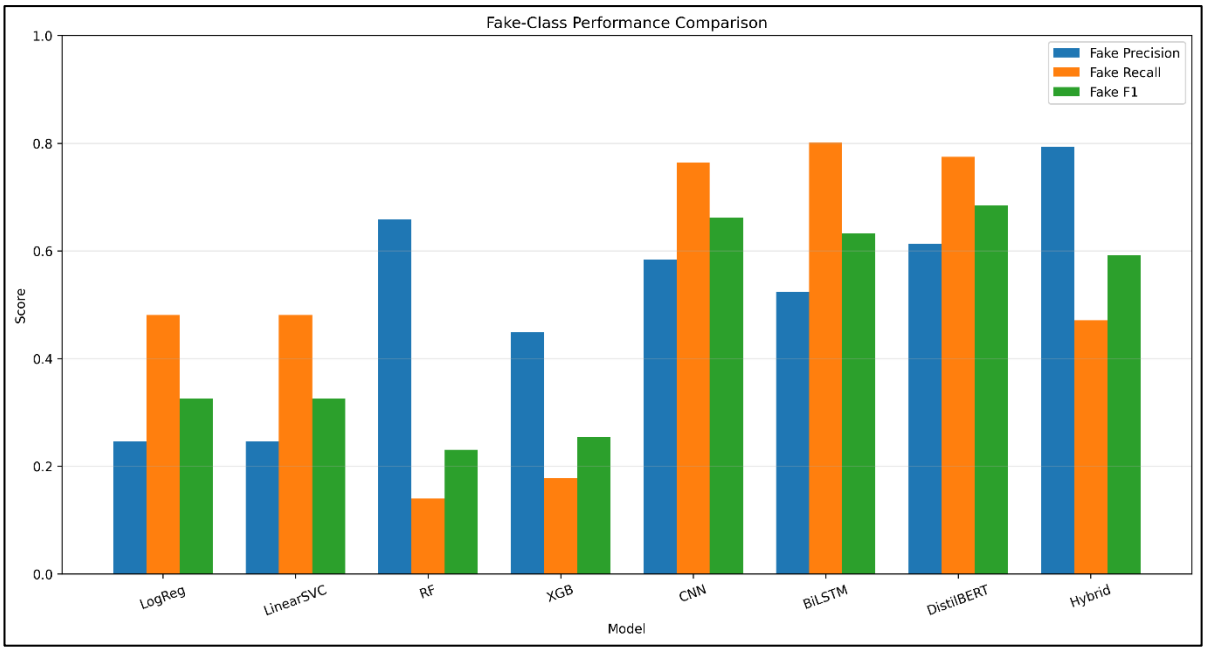

Figure 20. Fake-Class Performance Comparison

The overall comparison shows a clear separation between the Machine Learning and Deep Learning approaches evaluated in this study. The Machine Learning baselines, particularly Logistic Regression and LinearSVC, produced substantially weaker results across nearly all metrics. Their accuracy remained close to 52%, Cohen's Kappa was near zero, and ROC-AUC was only marginally above random performance, indicating very limited discriminative ability in the headline-only setting.

The two tree-based Machine Learning models, Random Forest and XGBoost, improved overall accuracy relative to the linear baselines, but their performance on the fake news class remained weak. In both cases, fake recall was particularly low, which indicates that these models favored the majority real news class and failed to recover a substantial portion of fake instances. This behavior resulted in low fake-class F1-scores despite their higher global accuracy.

In contrast, all Deep Learning models achieved markedly stronger results than the Machine Learning approaches. Their ROC-AUC values were substantially higher, their Cohen's Kappa values indicated stronger agreement beyond chance, and their fake-class F1-scores showed much more balanced performance on the minority class. This pattern suggests that the Deep Learning architectures were better suited to capturing the semantic and contextual information available in short headlines.

When the Deep Learning models are compared jointly with the Machine Learning baselines, DistilBERT emerges as the most balanced overall model. It achieved the highest ROC-AUC, the highest Cohen's Kappa, and the highest fake-class F1-score among all evaluated models, while also maintaining strong overall accuracy and weighted F1. These results indicate that DistilBERT provided the best trade-off between overall classification quality and minority-class effectiveness.

The hybrid DistilBERT + XGBoost model achieved the highest overall accuracy and the highest fake precision among all models. However, its fake recall was lower than that of the other Deep Learning models, which reduced its fake-class F1-score. This suggests that the hybrid model was more conservative when assigning the fake label, producing fewer false positives but missing more fake instances.

BiLSTM achieved the highest fake recall among all evaluated models, indicating that it was particularly effective at identifying fake news instances. CNN, on the other hand, provided a more balanced performance between overall accuracy and fake-class effectiveness. Taken together, the results suggest that the selection of the best-performing model depends on the performance objective. If the priority is the strongest balance across metrics, DistilBERT appears to be the most suitable choice. If the goal is to maximize accuracy or fake precision, the hybrid model performs best, whereas BiLSTM is preferable when fake recall is prioritized.

6.5. Comparison with Published Results

This section compares the findings of the present thesis with selected published studies on fake news detection. The purpose of this comparison is to position the reported results within the broader literature, while taking into account differences in datasets, input representations, and evaluation protocols. Since several published approaches rely on richer information sources, such as social context or user comments, the reported values should be interpreted as indicative reference points rather than directly equivalent benchmarks. The present study is therefore compared primarily with works that are either methodologically broad, based on FakeNewsNet, or relevant to transformer-based and hybrid approaches.

Study	**Dataset / Setting**	**Main Input / Approach**	**Main Reported Finding**	**Relevance to present thesis**
Khan et al. (2021)	Multiple datasets, benchmark setting	Traditional ML, Deep Learning, and pretrained transformer models	Pretrained models such as BERT and related architectures outperform traditional and deep learning baselines,	Supports the broader pattern observed in the present study, where transformer-based models

Study	Dataset / Setting	Main Input / Approach	Main Reported Finding	Relevance to present thesis
			particularly on small datasets	outperformed the ML baselines
Rai et al. (2022)	FakeNewsNet, including GossipCop	News-title based classification using BERT and BERT+LSTM	The proposed BERT+LSTM model improved accuracy over vanilla BERT by 1.10% on GossipCop	Directly relevant because it uses FakeNewsNet and title-based content classification
Albahar (2021)	PolitiFact and GossipCop	Hybrid model combining recurrent encoding and SVM, using news content and user comments	Reported accuracy of 80.2% and F1-score of 0.762 on GossipCop	Relevant as a hybrid reference, though it uses richer input than the present headline-only setup
Shu et al. (2020)	FakeNewsNet	News content, social context, and dynamic information	FakeNewsNet was designed as a multi-dimensional repository combining news content with social context and dynamic features	Useful for contextualizing the dataset choice and the limitations of comparing headline-only models with richer multi-source settings

Study	Dataset / Setting	Main Input / Approach	Main Reported Finding	Relevance to present thesis
Mouratidis et al. (2025)	Comparative analytical review / multi-model framework	Traditional ML, deep learning, BERT, and hybrid variants	The study highlights the strong performance of BERT-based approaches and discusses the growing relevance of hybrid architectures in fake news detection	Supports the broader literature trend favoring transformer-based and hybrid approaches

Table 38. Comparison with selected published studies

The comparison with published work shows that the findings of the present thesis are broadly consistent with the current literature on fake news detection. In particular, the stronger performance of the Deep Learning models over the Machine Learning baselines agrees with the broader benchmark evidence reported by Khan et al. (2021), who found that deep learning and especially pretrained transformer-based models outperform traditional methods in fake news classification. This broader trend is also aligned with the results obtained in the present study, where DistilBERT and the hybrid DistilBERT + XGBoost configuration clearly exceeded the Doc2Vec-based Machine Learning models in terms of overall discriminative performance.

A closer point of comparison is provided by Rai et al. (2022), who also worked with the FakeNewsNet dataset and followed a content-based classification approach using news titles. Their findings showed that a BERT+LSTM model improved over vanilla BERT on GossipCop, which is in line with the general effectiveness of transformer-based and sequence-aware

architectures observed in the present thesis. At the same time, direct numerical comparison should be made cautiously, since the exact preprocessing steps, model configurations, and evaluation details differ across studies.

The hybrid configuration evaluated in the present thesis can also be interpreted in relation to prior hybrid fake news detection research. Albahar (2021) reported strong results on GossipCop using a hybrid model that combined recurrent representations with an SVM classifier, but this approach also relied on user comments in addition to news content. This methodological difference is important, because the current thesis intentionally restricts the input to headline text only. Consequently, the present hybrid DistilBERT + XGBoost model is comparable in architectural spirit, but not directly equivalent in informational richness.

This distinction is further supported by the FakeNewsNet dataset paper itself, which explicitly presents FakeNewsNet as a repository combining news content, social context, and dynamic information from sources such as PolitiFact and GossipCop. Since the current thesis used only the headline component of GossipCop, the obtained results should be interpreted as reflecting a more restricted but also more controlled content-based setting.

Finally, the broader literature also supports the strong role of BERT-based and hybrid approaches in fake news detection. Mouratidis et al. (2025) describe BERT-based models as highly effective for contextual misinformation detection and note the growing relevance of hybrid architectures that combine transformer representations with complementary learning components. This overall direction is consistent with the present findings, where DistilBERT achieved the most balanced performance and the hybrid DistilBERT + XGBoost model achieved the highest accuracy.

7. Discussion and Future Work

7.1. Introduction

This chapter discusses the findings reported in Chapters 5 and 6 and interprets them in relation to the objectives of the present thesis. The aim is not only to identify which models performed better, but also to explain the observed performance patterns in light of the characteristics of the dataset, the headline-only setting, and the imbalance between fake and real news instances. Particular attention is given to the relative behavior of Machine Learning and Deep Learning approaches, the trade-offs observed across evaluation metrics, and the implications of the selected experimental design.

7.2. Interpretation of the Machine Learning Results

The Machine Learning results indicate that the headline-only setting posed a substantial difficulty for the traditional classifiers evaluated in this thesis. Logistic Regression and LinearSVC produced almost identical outcomes across all major metrics, suggesting that the linear separation of the Doc2Vec feature space was not sufficient to distinguish fake from real headlines with satisfactory accuracy. Their very low Cohen's Kappa values and ROC-AUC scores close to 0.50 further support this interpretation, indicating that these models had only limited discriminative capacity in the present task.

Although both linear models achieved moderate fake recall compared with the other Machine Learning approaches, this came at the cost of very low fake precision. In practical terms, this means that they identified a portion of the fake headlines, but also produced a large number of false positive classifications. As a result, their fake-class F1-scores remained modest, confirming that their performance on the minority class was not balanced enough to support strong overall effectiveness.

The tree-based methods, Random Forest and XGBoost, improved overall accuracy compared with the linear baselines, indicating that they were better able to capture non-linear structure in the document representations. However, this improvement was not matched by equally strong fake-class recovery. In both cases, fake recall remained low, especially for Random Forest, which suggests that these models strongly favored the majority real news class. This behavior explains why higher accuracy did not translate into stronger fake-class F1-scores.

Taken together, the Machine Learning findings suggest that Doc2Vec-based traditional classifiers were not sufficient to model the complexity of the fake news detection task when only headlines were used as input. While the tree-based models offered some gains in global accuracy, they still struggled to recover fake instances effectively. This indicates that, under the present experimental setting, traditional Machine Learning approaches were limited both by the restricted textual input and by the imbalance of the dataset.

7.3. Interpretation of the Deep Learning Results

The Deep Learning results indicate that architectures capable of learning richer semantic and contextual representations from text were substantially more effective than the Machine Learning baselines in the present headline-only setting. Although all Deep Learning models were trained using the same input source, namely headline text alone, they achieved much stronger discriminative performance, suggesting that they were better able to exploit the limited linguistic information available in short news titles.

The CNN and BiLSTM models demonstrated that sequence-based neural architectures can improve fake news detection even under constrained textual conditions. CNN achieved strong overall balance across the reported metrics, indicating that local lexical patterns and short-range textual features were informative for the classification task. BiLSTM, by contrast, produced the highest fake recall among the evaluated Deep Learning models, which suggests that sequential modeling of headline structure was particularly useful for recovering fake

instances, even though this came at the cost of lower precision and lower overall balance compared with some other approaches.

DistilBERT produced the most balanced overall Deep Learning performance. Its strong ROC-AUC, Cohen's Kappa, and fake-class F1-score suggest that contextual transformer-based representations were more effective than standard sequence models in distinguishing fake from real headlines. This result is consistent with the broader literature, where transformer-based models are often found to outperform both traditional Machine Learning methods and simpler neural architectures in misinformation detection tasks (Khan et al., 2021; Mouratidis et al., 2025).

The hybrid DistilBERT + XGBoost model achieved the highest overall accuracy, which shows that transformer-derived embeddings can also function effectively as input to a strong non-linear classifier. However, its lower fake recall compared with the other Deep Learning models indicates that this architecture was more conservative in assigning the fake label. In other words, the hybrid model achieved stronger precision at the expense of missing more fake instances. This suggests that higher overall accuracy does not necessarily correspond to the best balance for minority-class detection.

Taken together, the Deep Learning findings indicate that improved representational capacity is highly beneficial in headline-based fake news detection, but also that different architectures emphasize different performance trade-offs. DistilBERT provided the strongest balance across evaluation metrics, BiLSTM was most effective in recovering fake instances, and the hybrid model maximized accuracy while sacrificing minority-class sensitivity.

7.4. Impact of Class Imbalance and Headline-Only Input

Two factors appear to have shaped the experimental outcomes of the present study more strongly than any individual modeling choice: the imbalance between fake and real news instances and the restriction of the input to headline text only. These two characteristics jointly

define a demanding classification setting in which both minority-class detection and contextual interpretation become more difficult.

The class distribution of the GossipCop subset is notably skewed toward real news, which means that models can achieve apparently satisfactory global accuracy while still performing poorly on the fake news class. This pattern was clearly reflected in several of the evaluated models, particularly among the Machine Learning baselines, where higher accuracy did not necessarily correspond to strong fake-class recall or fake-class F1-score. For this reason, the interpretation of performance in the present study relied not only on accuracy, but also on class-sensitive metrics such as precision, recall, F1-score, ROC-AUC, and Cohen's Kappa, which are more appropriate in imbalanced classification settings (He and Garcia, 2009).

At the same time, the headline-only setting imposed a second major limitation. Headlines are short, compressed, and often intentionally attention-oriented, which means that they may contain stylistic signals without providing sufficient contextual depth for full credibility assessment. As a result, the models examined in this thesis had to make predictions under conditions of restricted semantic information. This helps explain why the Machine Learning models struggled substantially, but it also clarifies why the differences among the Deep Learning models, although meaningful, were not extremely large. When the available textual signal is inherently limited, even architectures with stronger representational capacity may converge toward relatively close performance ranges (Shu et al., 2017; Zhou and Zafarani, 2020).

The behavior of the imbalance-handling strategies should also be interpreted in this context. For the Machine Learning models, SMOTE was included as a principled method for addressing skewed class distributions during training. However, the results suggest that oversampling alone was not sufficient to overcome the representational limitations imposed by short headline text. Synthetic balancing can increase exposure to the minority class, but it cannot introduce genuinely new semantic context when the original input is already sparse and compressed. This helps explain why the use of SMOTE did not lead to consistently strong minority-class improvements across the Machine Learning models (Chawla et al., 2002; He and Garcia, 2009).

For the Deep Learning models, class weighting constituted a more suitable strategy within the holdout-based training design, yet the same structural limitation remained: even when the models were encouraged to pay greater attention to the minority class, they still operated on short and context-restricted textual inputs. Therefore, the challenge in the present study was not only imbalance in a numerical sense, but also informational imbalance, in the sense that the available text often contained insufficient evidence for fully robust discrimination.

Overall, the findings of this thesis suggest that the combination of class imbalance and headline-only input creates a constrained fake news detection setting in which model performance must be interpreted carefully. In such conditions, improvements in one metric may come at the expense of another, and stronger architectures may still be bounded by the limited informational richness of the dataset. This also explains why balanced interpretation across multiple evaluation metrics is more informative than reliance on accuracy alone.

7.5. Strengths and Limitations of the Present Study

The present study has several strengths that support the validity of its findings. First, it adopts a comparative framework that includes both traditional Machine Learning and more recent Deep Learning approaches, allowing a broad evaluation of fake news detection under a common headline-based setting. Second, the study applies a consistent experimental logic, with clearly separated preprocessing pipelines, explicit imbalance-handling strategies, and multiple evaluation metrics designed to capture both overall performance and minority-class effectiveness. Third, the inclusion of sequence-based, transformer-based, and hybrid architectures enables a more informative comparison than a narrower model selection would allow.

Another strength lies in the deliberate restriction of the input to headline text. Although this choice imposes clear limitations, it also increases methodological control by isolating the contribution of textual headline information without the influence of external

metadata, user interactions, or social propagation signals. In this sense, the study provides a focused assessment of how far fake news detection can be supported by headline content alone.

At the same time, the study is subject to important limitations. The most significant limitation is the exclusive use of headline text, which restricts the semantic and contextual depth available to the models. In real-world fake news detection settings, additional information such as full article content, publisher characteristics, user engagement patterns, and propagation dynamics may offer substantial predictive value (Shu et al., 2017; Shu et al., 2019). As a result, the findings of the present thesis should be interpreted as specific to a constrained content-based setting rather than as a complete representation of all possible fake news detection scenarios.

A second limitation concerns class imbalance within the GossipCop subset. Although imbalance-handling methods were incorporated into the experimental design, skewed class distributions still influenced model behavior and complicated the interpretation of global metrics such as accuracy. In addition, the use of different validation strategies for Machine Learning and Deep Learning models, while methodologically justified, means that comparisons across model families should be interpreted with appropriate caution. The distinction between stratified cross-validation and holdout-based evaluation does not invalidate the comparison, but it does introduce a structural difference in the estimation of performance.

A further limitation concerns the scope of hyperparameter exploration. While the evaluated models were configured in a principled and reproducible way, the study did not attempt exhaustive hyperparameter optimization across all architectures. The reported results therefore reflect a strong comparative experimental design, but not an absolute claim of globally optimal performance for every model configuration.

Overall, the strengths of the study lie in its methodological clarity, comparative breadth, and controlled headline-based design, while its limitations stem primarily from the restricted input setting, the imbalance of the dataset, and the practical constraints of model optimization and evaluation. These limitations do not undermine the value of the findings, but they help define the scope within which the conclusions should be interpreted.

7.6. Future Research Directions

Several directions for future research emerge from the findings and limitations of the present study. The most immediate extension would be the incorporation of richer textual and contextual information beyond headlines alone. Future work could examine whether the inclusion of full article content, metadata, publisher-level information, or social-context signals leads to more robust and discriminative fake news detection performance, particularly for the minority fake news class (Shu et al., 2017; Shu et al., 2019).

A second important direction concerns the systematic integration of social and propagation-based features. Since fake news dissemination on social media is shaped not only by textual content but also by engagement patterns, user interactions, and diffusion dynamics, future research could investigate multimodal or multi-source architectures that combine textual representations with social-context information. Such approaches may help overcome the limitations observed in headline-only classification settings (Zhou and Zafarani, 2020; Shu et al., 2018).

Further work could also extend the comparative modeling framework developed in this thesis. More systematic hyperparameter optimization, threshold tuning, and alternative imbalance-handling strategies could be explored in order to assess whether additional gains are possible, particularly for minority-class recall and fake-class F1-score. In addition, future studies could examine whether other transformer-based architectures, such as RoBERTa or domain-adapted language models, provide stronger results than DistilBERT under similar experimental conditions (Khan et al., 2021; Mouratidis et al., 2025).

Another relevant direction would be the evaluation of the proposed approaches under more standardized and directly comparable validation protocols. Although the use of different validation strategies for Machine Learning and Deep Learning models was methodologically justified in the present study, future work could investigate unified experimental designs that facilitate even more direct cross-family comparison.

Finally, future research could expand the present work beyond binary fake news detection and investigate more fine-grained credibility assessment tasks. These may include

distinguishing between misinformation categories, incorporating explainability mechanisms, or developing models that support human-centered verification rather than only automated classification. Such directions would broaden the practical relevance of fake news detection research and align it more closely with real-world information verification needs.

8. Conclusions

The present thesis examined the problem of fake news detection in a constrained headline-only setting using the FakeNewsNet GossipCop subset. The study compared a range of Machine Learning and Deep Learning models under clearly defined preprocessing, validation, and evaluation procedures, with particular attention to class imbalance and minority-class performance.

The findings showed that the Machine Learning baselines were generally limited in their ability to distinguish fake from real headlines. Logistic Regression and LinearSVC produced weak discriminative performance, while Random Forest and XGBoost improved overall accuracy but remained less effective in recovering fake news instances. These results indicate that traditional classifiers trained on Doc2Vec representations were not sufficient to model the complexity of the task under restricted textual conditions.

In contrast, the Deep Learning models achieved substantially stronger results. CNN and BiLSTM both improved performance on the fake news class, while DistilBERT provided the most balanced overall behavior across the reported metrics. The hybrid DistilBERT + XGBoost model achieved the highest overall accuracy, although this came with lower fake recall compared with the strongest sequence-based and transformer-based alternatives. Overall, the results suggest that richer representational capacity is particularly important in headline-based fake news detection.

At the same time, the study highlighted two major constraints shaping model performance: the imbalance of the dataset and the restricted informational content of headline text. These factors limited the extent to which performance gains could be achieved and help explain both the weakness of the Machine Learning baselines and the relatively bounded differences among the stronger Deep Learning approaches.

Taken together, the results of this thesis support the conclusion that Deep Learning models, and especially transformer-based approaches, are more suitable than traditional Machine Learning methods for fake news detection in headline-only environments. However, the findings also show that even strong architectures remain affected by the limited contextual

depth of short-text inputs. Therefore, fake news detection based solely on headlines should be understood as a relevant but inherently constrained classification problem.

References

[1] Albahar, Marwan. (2021). A hybrid model for fake news detection: Leveraging news content and user comments in fake news. IET Information Security. 15. 10.1049/ise2.12021.

[2] Chawla, Nitesh & Bowyer, Kevin & Hall, L. & Kegelmeyer, W.. (2002). SMOTE: Synthetic Minority Over-sampling Technique. Journal of Artificial Intelligence Research. 16. 321-357. 10.1613/jair.953

[3] He, Haibo & Garcia, E.A.. (2009). Learning from Imbalanced Data. Knowledge and Data Engineering, IEEE Transactions on. 21. 1263 - 1284. 10.1109/TKDE.2008.239.

[4] Khan, Junaed Younus & Khondaker, Md. Tawkat Islam & Afroz, Sadia & Uddin, Gias & Iqbal, Anindya. (2021). A benchmark study of machine learning models for online fake news detection. Machine Learning with Applications. 4. 100032. 10.1016/j.mlwa.2021.100032.

[5] Le, Quoc & Mikolov, Tomas. (2014). Distributed Representations of Sentences and Documents. 31st International Conference on Machine Learning, ICML 2014. 4.

[6] Mouratidis, D., Kanavos, A., & Kermanidis, K. (2025). From Misinformation to Insight: Machine Learning Strategies for Fake News Detection. *Information*, *16*(3), 189.

[7] Rai, Nishant & Kumar, Deepika & Kaushik, Naman & Raj, Chandan & Ali, Ahad. (2022). Fake News Classification using transformer based enhanced LSTM and BERT. International Journal of Cognitive Computing in Engineering. 3. 10.1016/j.ijcce.2022.03.003.

[8] Reis, Julio & Correia, Andre & Murai, Fabricio & Veloso, Adriano & Benevenuto, Fabrício & Cambria, Erik. (2019). Supervised Learning for Fake News Detection. IEEE Intelligent Systems. 34. 76-81. 10.1109/MIS.2019.2899143. Sanh, V., Debut, L., Chaumond, J. and Wolf, T. (2019) 'DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter', arXiv preprint arXiv:1910.01108.

[9] Shu, Kai & Sliva, Amy & Wang, Suhang & Tang, Jiliang & Liu, Huan. (2017). Fake News Detection on Social Media: A Data Mining Perspective. ACM SIGKDD Explorations Newsletter. 19. 10.1145/3137597.3137600.

[10] Shu, Kai & Mahudeswaran, Deepak & Liu, Huan. (2019). FakeNewsTracker: a tool for fake news collection, detection, and visualization. Computational and Mathematical Organization Theory. 25. 10.1007/s10588-018-09280-3.

[11] Shu, Kai & Mahudeswaran, Deepak & Wang, Suhang & Lee, Dongwon & Liu, Huan. (2018). FakeNewsNet: A Data Repository with News Content, Social Context and Dynamic Information for Studying Fake News on Social Media. 10.48550/arXiv.1809.01286.

[12] Shu, Kai & Mahudeswaran, Deepak & Wang, Suhang & Lee, Dongwon & Liu, Huan. (2020). FakeNewsNet: A Data Repository with News Content, Social Context, and Spatiotemporal Information for Studying Fake News on Social Media. Big Data. 8. 171-188. 10.1089/big.2020.0062.

[13] William Yang Wang. 2017. "Liar, Liar Pants on Fire": A New Benchmark Dataset for Fake News Detection. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 422–426, Vancouver, Canada. Association for Computational Linguistics.

[14] Zhou, X. and Zafarani, R., 2020. A survey of fake news: Fundamental theories, detection methods, and opportunities. *ACM Computing Surveys (CSUR)*, *53*(5), pp.1-40.

Figures References

[Figure 1]: Shu, Kai & Mahudeswaran, Deepak & Liu, Huan. (2019). FakeNewsTracker: a tool for fake news collection, detection, and visualization. Computational and Mathematical Organization Theory. 25. 10.1007/s10588-018-09280-3.

[Figure 3]: Papadonikolakis, Markos & Bouganis, Christos. (2012). Novel Cascade FPGA Accelerator for Support Vector Machines Classification. IEEE Transactions on Neural Networks. 23. 1040-1052. 10.1109/TNNLS.2012.2196446.

[Figure 4]: Saikia, Dimple & Dadhara, Ritam & Tanan, Cebajel & Avati, Prajwal & Verma, Tushar & Pandey, Rishikesh & Singh, Surya. (2025). Combating Antimicrobial Resistance: Spectroscopy Meets Machine Learning. Photonics. 12. 672. 10.3390/photonics12070672.

[Figure 5]: Zhang, Ye & Wallace, Byron. (2015). A Sensitivity Analysis of (and Practitioners' Guide to) Convolutional Neural Networks for Sentence Classification.

[Figure 6]: Resiandi, K., Murakami, Y., & Nasution, A. H. (2023). Neural Network-Based Bilingual Lexicon Induction for Indonesian Ethnic Languages. Applied Sciences, 13(15), 8666.

[Figure 7]: Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17). Curran Associates Inc., Red Hook, NY, USA, 6000–6010.

Appendix A. Preprocessing and Implementation Details

Appendix A.1 Dataset Files and Input Structure

The experiments of this thesis were based on the GossipCop subset of FakeNewsNet. The original data were organized into separate fake and real news files, which were subsequently merged into a unified dataset for preprocessing and modeling. Only the headline text and the corresponding binary label were retained for the purposes of this study. Full article content, user interaction metadata, and propagation-related information were excluded in order to maintain a controlled headline-only experimental setting.

Appendix A.2 Preprocessing Workflow

The preprocessing pipeline was implemented as a sequence of modular scripts. The main stages included dataset consolidation, text normalization, tokenization, lexical filtering, and preparation of the final inputs for feature extraction or neural training. For the Machine Learning models, the cleaned headlines were converted into Doc2Vec-ready documents and subsequently transformed into dense 300-dimensional document embeddings. For the Deep Learning models, the same headline data were transformed into architecture-specific inputs, including padded token sequences for CNN and BiLSTM, and transformer-tokenized inputs for DistilBERT-based models.

Appendix A.3 Feature Representations

Two broad categories of feature representation were used in this study. The Machine Learning models operated on dense Doc2Vec embeddings obtained by concatenating Distributed Memory and Distributed Bag of Words representations. This resulted in a fixed 300-dimensional numerical vector for each headline.

The Deep Learning models used text representations adapted to the architecture of each model. CNN and BiLSTM were trained on tokenized and padded headline sequences,

whereas DistilBERT was trained on transformer-tokenized inputs with attention masks. In the hybrid DistilBERT + XGBoost configuration, DistilBERT was used as a feature extractor to produce fixed-length contextual embeddings, which were then used as input to the XGBoost classifier.

Appendix A.4 Execution Environments

The preprocessing pipeline and the Machine Learning experiments based on Logistic Regression, LinearSVC, and Random Forest were implemented and executed in Visual Studio Code (VS Code). XGBoost and all Deep Learning experiments, including CNN, BiLSTM, DistilBERT, and Hybrid DistilBERT + XGBoost, were executed in Google Colab. The Colab-based experiments used NVIDIA T4 GPU resources when required by the computational demands of the models. The main software libraries employed in the study included scikit-learn, XGBoost, TensorFlow/Keras, PyTorch, and Hugging Face Transformers.

Appendix B. Representative Code Snippets

Appendix B.1 Representative Preprocessing Script

The following code excerpt presents a representative portion of the preprocessing pipeline used in this thesis. The snippet illustrates the consolidation of the raw fake and real headline files, the assignment of class labels, the normalization and tokenization of the headline text, and the lexical filtering steps applied before preparing the documents for Doc2Vec-based feature generation.

```python
import pandas as pd
import nltk
import string
from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords

nltk.download("punkt")
nltk.download("stopwords")

stop = set(stopwords.words("english"))

# 1. Load and merge raw fake and real files
fake = pd.read_csv("data/raw/gossipcop_fake.csv")
real = pd.read_csv("data/raw/gossipcop_real.csv")

fake["label"] = "fake"
real["label"] = "real"

df = pd.concat([fake, real], ignore_index=True)
df = df[["title", "label"]].rename(columns={"title": "text"})

# 2. Basic text normalization
def clean_text(text):
    text = str(text).lower()
    text = text.translate(str.maketrans("", "", string.punctuation))
    return text

df["clean_text"] = df["text"].apply(clean_text)
```

```python
# 3. Tokenization
df["tokens"] = df["clean_text"].apply(word_tokenize)

# 4. Lexical filtering for Doc2Vec preparation
def clean_tokens(token_list):
    cleaned = []
    for t in token_list:
        if t in stop:
            continue
        if t.isdigit():
            continue
        if len(t) < 2:
            continue
        cleaned.append(t)
    return cleaned

df["doc2vec_tokens"] = df["tokens"].apply(clean_tokens)

# 5. Save processed file
df.to_csv("data/processed/gossipcop_doc2vec_ready.csv", index=False)
```

Appendix B.2 Representative Machine Learning Training Script

The following code excerpt presents a representative implementation of the Machine Learning training and evaluation pipeline used in this thesis. The example corresponds to the Logistic Regression classifier and illustrates the use of Doc2Vec embeddings, feature scaling, SMOTE oversampling within the training folds, stratified 10-fold cross-validation, and metric-based evaluation.

```python
import os
import pandas as pd
import matplotlib.pyplot as plt

from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import StratifiedKFold
from sklearn.metrics import (
    confusion_matrix,
    precision_score,
    recall_score,
    f1_score,
    accuracy_score,
    roc_auc_score,
    cohen_kappa_score,
    roc_curve
)
from sklearn.preprocessing import StandardScaler
from imblearn.pipeline import Pipeline
from imblearn.over_sampling import SMOTE

# Load Doc2Vec embeddings
df = pd.read_csv("data/processed/gossipcop_doc2vec_embeddings.csv")
X = df.drop(columns=["label"])
y = df["label"].map({"fake": 0, "real": 1})
```

```python
# Model configuration
logreg = LogisticRegression(
    max_iter=500,
    solver="liblinear",
    random_state=42
)

pipeline = Pipeline([
    ("scaler", StandardScaler()),
    ("smote", SMOTE(k_neighbors=5, random_state=42)),
    ("clf", logreg)
])

skf = StratifiedKFold(n_splits=10, shuffle=True, random_state=42)

# Cross-validation
all_true, all_pred, all_prob = [], [], []

for train_idx, test_idx in skf.split(X, y):
    X_train, X_test = X.iloc[train_idx], X.iloc[test_idx]
    y_train, y_test = y.iloc[train_idx], y.iloc[test_idx]

    pipeline.fit(X_train, y_train)

    y_pred = pipeline.predict(X_test)
    y_prob = pipeline.predict_proba(X_test)[:, 1]

    all_true.extend(y_test.tolist())
    all_pred.extend(y_pred.tolist())
    all_prob.extend(y_prob.tolist())
```

```python
# Evaluation metrics
acc = accuracy_score(all_true, all_pred)
kappa = cohen_kappa_score(all_true, all_pred)
roc_auc_val = roc_auc_score(all_true, all_prob)

p_f = precision_score(all_true, all_pred, pos_label=0)
r_f = recall_score(all_true, all_pred, pos_label=0)
f_f = f1_score(all_true, all_pred, pos_label=0)

cm = confusion_matrix(all_true, all_pred)

# ROC curve
os.makedirs("outputs", exist_ok=True)

fpr, tpr, _ = roc_curve(all_true, all_prob)

plt.figure(figsize=(8, 6))
plt.plot(fpr, tpr, lw=2, label=f"ROC curve (AUC = {roc_auc_val:.4f})")
plt.plot([0, 1], [0, 1], lw=2, linestyle="--")
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel("False Positive Rate")
plt.ylabel("True Positive Rate")
plt.title("ROC Curve - Logistic Regression")
plt.legend(loc="lower right")
plt.grid(alpha=0.3)
plt.tight_layout()
plt.savefig("outputs/logistic_regression_roc_curve.png", dpi=300, bbox_inches="tight")
plt.show()
```

This representative script illustrates the core Machine Learning workflow used throughout the Doc2Vec-based experiments, with model-specific changes applied where appropriate for the remaining classifiers.

Appendix B.3 Representative Deep Learning Training Script

The following code excerpt presents a representative implementation of the Deep Learning training and evaluation pipeline used in this thesis. The example corresponds to the CNN model and illustrates the use of tokenized and padded headline sequences, class weighting, holdout validation, neural training with early stopping, and ROC-based evaluation.

```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

from sklearn.model_selection import train_test_split
from sklearn.metrics import (
    confusion_matrix,
    accuracy_score,
    cohen_kappa_score,
    precision_score,
    recall_score,
    f1_score,
    roc_auc_score,
    roc_curve,
    auc
)
from sklearn.utils.class_weight import compute_class_weight

from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Embedding, Conv1D, GlobalMaxPooling1D, Dense, Dropout
from tensorflow.keras.callbacks import EarlyStopping
```

```python
# Load dataset
df = pd.read_csv("data/processed/gossipcop_text_label.csv")
texts = df["text"].astype(str).tolist()
labels = df["label"].map({"fake": 0, "real": 1}).values

# Holdout split: 70% train, 15% validation, 15% test
X_temp, X_test, y_temp, y_test = train_test_split(
    texts, labels, test_size=0.15, stratify=labels, random_state=42
)

X_train, X_val, y_train, y_val = train_test_split(
    X_temp, y_temp, test_size=0.1765, stratify=y_temp, random_state=42
)
```

```python
# Tokenization and padding
max_words = 30000
max_len = 50

tokenizer = Tokenizer(num_words=max_words, oov_token="<UNK>")
tokenizer.fit_on_texts(X_train)

X_train_seq = tokenizer.texts_to_sequences(X_train)
X_val_seq = tokenizer.texts_to_sequences(X_val)
X_test_seq = tokenizer.texts_to_sequences(X_test)

X_train_pad = pad_sequences(X_train_seq, maxlen=max_len, padding="post", truncating="post")
X_val_pad = pad_sequences(X_val_seq, maxlen=max_len, padding="post", truncating="post")
X_test_pad = pad_sequences(X_test_seq, maxlen=max_len, padding="post", truncating="post")

# Class weights
class_weights = compute_class_weight(
    class_weight="balanced",
    classes=np.array([0, 1]),
    y=y_train
)
class_weight_dict = {0: class_weights[0], 1: class_weights[1]}
```

```python
# CNN model
vocab_size = min(max_words, len(tokenizer.word_index) + 1)
embedding_dim = 100

model = Sequential([
    Embedding(input_dim=vocab_size, output_dim=embedding_dim),
    Conv1D(filters=256, kernel_size=5, activation="relu"),
    GlobalMaxPooling1D(),
    Dropout(0.5),
    Dense(64, activation="relu"),
    Dropout(0.5),
    Dense(1, activation="sigmoid")
])

model.compile(
    loss="binary_crossentropy",
    optimizer="adam",
    metrics=["accuracy"]
)

early_stop = EarlyStopping(
    monitor="val_loss",
    patience=4,
    restore_best_weights=True
)
```

```python
# Training
history = model.fit(
    X_train_pad,
    y_train,
    epochs=30,
    batch_size=128,
    validation_data=(X_val_pad, y_val),
    class_weight=class_weight_dict,
    callbacks=[early_stop],
    verbose=1
)

# Evaluation
y_prob = model.predict(X_test_pad).ravel()
y_pred = (y_prob >= 0.5).astype(int)

acc = accuracy_score(y_test, y_pred)
kappa = cohen_kappa_score(y_test, y_pred)
roc_auc_val = roc_auc_score(y_test, y_prob)

cm = confusion_matrix(y_test, y_pred)
```

```python
# ROC curve
fpr, tpr, _ = roc_curve(y_test, y_prob)
roc_auc = auc(fpr, tpr)

plt.figure(figsize=(7, 6))
plt.plot(fpr, tpr, lw=2, label=f"ROC curve (AUC = {roc_auc:.4f})")
plt.plot([0, 1], [0, 1], lw=2, linestyle="--")
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel("False Positive Rate")
plt.ylabel("True Positive Rate")
plt.title("ROC Curve - CNN")
plt.legend(loc="lower right")
plt.grid(alpha=0.3)
plt.tight_layout()
plt.show()
```

Appendix B.4 Representative Hybrid Model Script

The following code excerpt presents a representative implementation of the hybrid DistilBERT + XGBoost pipeline used in this thesis. The hybrid approach first extracts fixed-length contextual embeddings from raw headlines using DistilBERT and then uses these embeddings as input to an XGBoost classifier for binary fake news classification.

```
import os
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import torch

from sklearn.model_selection import train_test_split
from sklearn.metrics import (
    confusion_matrix,
    accuracy_score,
    cohen_kappa_score,
    roc_auc_score,
    roc_curve,
    auc
)
from torch.utils.data import Dataset, DataLoader
from transformers import DistilBertTokenizerFast, DistilBertModel
from xgboost import XGBClassifier

# Load dataset
df = pd.read_csv("gossipcop_text_label.csv")
df = df.dropna(subset=["text"])

texts = df["text"].astype(str).tolist()
labels = df["label"].map({"fake": 0, "real": 1}).tolist()
```

```python
# Holdout split: 70% train, 15% validation, 15% test
X_temp, X_test, y_temp, y_test = train_test_split(
    texts,
    labels,
    test_size=0.15,
    stratify=labels,
    random_state=42
)

X_train, X_val, y_train, y_val = train_test_split(
    X_temp,
    y_temp,
    test_size=0.1765,
    stratify=y_temp,
    random_state=42
)
```

```python
# DistilBERT embedding extraction
pretrained_model = "distilbert-base-uncased"
max_length = 128
embedding_batch_size = 32

device = torch.device("cuda" if torch.cuda.is_available() else "cpu")

tokenizer = DistilBertTokenizerFast.from_pretrained(pretrained_model)
bert_model = DistilBertModel.from_pretrained(pretrained_model).to(device)
bert_model.eval()

class TextDataset(Dataset):
    def __init__(self, texts, tokenizer, max_length=128):
        self.encodings = tokenizer(
            texts,
            truncation=True,
            padding=True,
            max_length=max_length
        )

    def __getitem__(self, idx):
        return {k: torch.tensor(v[idx]) for k, v in self.encodings.items()}

    def __len__(self):
        return len(self.encodings["input_ids"])

def get_bert_embeddings(texts, batch_size=32, max_length=128):
    dataset = TextDataset(texts, tokenizer, max_length=max_length)
    loader = DataLoader(dataset, batch_size=batch_size, shuffle=False)
```

```
    all_embs = []

    with torch.no_grad():
        for batch in loader:
            input_ids = batch["input_ids"].to(device)
            attention_mask = batch["attention_mask"].to(device)

            outputs = bert_model(input_ids=input_ids, attention_mask=attention_mask)
            cls_emb = outputs.last_hidden_state[:, 0, :]
            all_embs.append(cls_emb.cpu().numpy())

    return np.concatenate(all_embs, axis=0)

X_train_emb = get_bert_embeddings(X_train, batch_size=embedding_batch_size, max_length=max_length)
X_val_emb = get_bert_embeddings(X_val, batch_size=embedding_batch_size, max_length=max_length)
X_test_emb = get_bert_embeddings(X_test, batch_size=embedding_batch_size, max_length=max_length)
```

```
# XGBoost classifier
xgb = XGBClassifier(
    n_estimators=300,
    max_depth=6,
    learning_rate=0.05,
    subsample=0.8,
    colsample_bytree=0.8,
    eval_metric="logloss",
    tree_method="hist"
)

xgb.fit(
    X_train_emb,
    y_train,
    eval_set=[(X_val_emb, y_val)],
    verbose=False
)

# Evaluation
y_pred = xgb.predict(X_test_emb)
y_prob = xgb.predict_proba(X_test_emb)[:, 1]

acc = accuracy_score(y_test, y_pred)
kappa = cohen_kappa_score(y_test, y_pred)
roc_auc_val = roc_auc_score(y_test, y_prob)
cm = confusion_matrix(y_test, y_pred)
```

```python
# ROC curve
os.makedirs("outputs", exist_ok=True)

fpr, tpr, _ = roc_curve(y_test, y_prob)
roc_auc = auc(fpr, tpr)

plt.figure(figsize=(8, 6))
plt.plot(fpr, tpr, lw=2, label=f"ROC curve (AUC = {roc_auc:.4f})")
plt.plot([0, 1], [0, 1], lw=2, linestyle="--")
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel("False Positive Rate")
plt.ylabel("True Positive Rate")
plt.title("ROC Curve - Hybrid DistilBERT + XGBoost")
plt.legend(loc="lower right")
plt.grid(alpha=0.3)
plt.tight_layout()
plt.savefig("outputs/hybrid_distilbert_xgboost_roc_curve.png", dpi=300, bbox_inches="tight")
plt.show()
```