Fake News Detection System

Group 03: Chen Yunjian (A0297854H), Chiu Zheng Qian (A0221547J), Liu Yibo (A0297810X), Wang Haoyu (A0296165U), Zhang Shenxin (A0295793H)

GitHub Link: https://github.com/zhengqian89/group03-fake-news-detection

Abstract

Fake news poses a serious threat to society by spreading misleading or fabricated information at an alarming rate. In this project, we develop and evaluate a system for detecting fake news using a variety of machine learning and deep learning models. We experiment on a publicly available news dataset from Kaggle that contains roughly 7,800 labeled articles, equally split between "REAL" and "FAKE." The models employed range from simpler text classification approaches (Bag-of-Words with Logistic Regression) to advanced neural architectures (LSTM, BERT) and a Sentence Transformer. We assess each model's performance using accuracy, precision, recall, and F1-score. Experimental results indicate that while traditional methods (e.g., TF-IDF + Logistic Regression) are strong baselines, transformer-based embeddings (e.g., Sentence-BERT + LR) and fine-tuned BERT yield the highest accuracy, exceeding 96%. We conclude with a discussion of ethical considerations, particularly around balancing false positives and false negatives, and potential strategies for deploying a reliable fake news detection tool in real-world contexts.

1. Introduction

1.1 Problem Statement

The proliferation of false or misleading news—commonly referred to as "fake news"—has emerged as a critical global societal challenge, particularly highlighted during events such as the 2016 U.S. presidential election (Lazer et al., 2018). Research conducted by MIT in 2018 demonstrated that false news spreads on social media with remarkable efficiency, traveling "farther, faster, deeper, and more broadly" than true news (Study, 2018). Notably, false stories were found to be 70% more likely to be retweeted compared to factual ones (Study, 2018), emphasizing how misinformation can rapidly permeate networks dominated by real users rather than automated bots (Study, 2018). The real-world consequences of this phenomenon are significant. For instance, a study published in Nature revealed that belief in COVID-19 misinformation was strongly correlated with reduced compliance with public health guidelines and lower vaccination intent (van der Linden, 2022), directly impeding efforts to manage the pandemic. Beyond healthcare, the widespread dissemination of falsehoods online has eroded trust in institutions and exacerbated political polarization (Lazer et al., 2018). A 2024 report by NewsGuard further underscores the issue, noting that "fake" local news sites now outnumber legitimate local newspapers in the U.S. ("Sad Milestone," n.d.). These examples highlight not only the rapid spread of misinformation but also its tangible harms, ranging from undermining democratic discourse to jeopardizing public health.

1.2 Objectives and Scope

In response to this growing threat, there is an urgent need for robust fake news detection systems. We develop and evaluate six distinct machine learning pipelines, spanning classical ML text classifiers to advanced deep learning models, and compare their performance.

The primary objectives of this project are threefold:

1. Compare diverse classification methodologies for fake news detection, assessing trade-offs in accuracy, computational demands, and interpretability to identify optimal solutions.
2. Quantify model performance using key metrics, e.g., accuracy, precision, recall, and F1-score, to ensure the system effectively detects fake content while minimizing the risk of misclassifying legitimate articles.
3. Address practical deployment considerations, including data quality, resource limitations, and ethical concerns such as avoiding unjust censorship or introducing systemic biases.

Through rigorous experimentation and evaluation, this study seeks to establish best practices for deploying reliable fake news detection systems in real-world online environments, ensuring both effectiveness and responsible implementation.

2. Dataset

The dataset used in this study is the "News.csv" fake news dataset sourced from Kaggle (News.Csv, n.d.), which comprises 7,796 articles, each labeled as either “REAL” or “FAKE”. The data were curated from a variety of online news sources, with "real" news articles originating from credible mainstream outlets and "fake" news articles collected from unreliable or dubious websites. Each record in the dataset includes four columns: an ID, the article's title, the body text of the article, and a label indicating whether the article is real or fake. For instance, a typical entry contains the headline and full text of a news article, along with a binary label identifying it as true or false. We use only the article body (`text` column) as our input feature, while the label acts as the target variable.

The dataset is moderate in size, with 7,796 samples, and notably balanced—approximately 50% of the articles are labeled as real, and 50% as fake. Specifically, there are 3,897 articles labeled as REAL and 3,899 labeled as FAKE, ensuring a nearly equal distribution between the two classes. This balance mitigates concerns about classification bias arising from class imbalance. The topics covered in the dataset predominantly revolve around political news and general world events, reflecting the data collection period during significant political events in the mid-2010s. Overall, the dataset provides a realistic foundation for binary fake news classification, encompassing both legitimate news and misinformation. All articles are in English.

3. Pre-processing Steps

3.1 Data Cleaning

To ensure consistency and reduce noise in the dataset, several data cleaning steps were performed. These steps are designed to standardize the input and focus on meaningful content while minimizing irrelevant or redundant information.

3.1.1 Removal of Punctuation and Special Characters

All punctuation marks (e.g., periods, commas, quotes, exclamation points) and special symbols were removed from the news text. A predefined punctuation list was used to strip these characters, replacing them with whitespace or removing them entirely as appropriate. This step prevents punctuation from being treated as separate tokens and reduces sparsity in representations like Bag-of-Words. For example, the sentence “The election was rigged!!!” would be transformed into “The election was rigged.” Intra-word characters such as hyphens were retained if they formed part of a word, but standalone symbols were eliminated.

3.1.2 Lowercasing

All text was converted to lowercase to normalize case differences. This ensures that words like “Government” and “government” are treated identically. Lowercasing reduces vocabulary size for Bag-of-Words and TF-IDF representations and helps models generalize across inconsistent capitalization, particularly in titles. For instance, “Fake” and “fake” become the same token: "fake".

3.1.3 Stopword Removal

For BoW and TF-IDF pipelines, we rely on `CountVectorizer(stop_words='english')` and `TfidfVectorizer(stop_words='english')` to filter English stopwords. These frequent words, such as "the," "is," "at," "on," "and," and "a," carry little semantic value and are prevalent in both real and fake news, typically offering no distinguishing power between classes. In contrast, no explicit stopword removal is applied in the GloVe, LSTM, BERT, or SBERT pipelines, as these models either leverage contextual embeddings or inherently handle stopwords through their tokenizers and architectures.

3.1.4 Numerical Handling

All digits are removed through `re.sub(r'\d+', '', text)`. Specific numbers, such as dates and statistics, often lack generalizable significance, and treating all numbers uniformly simplifies the model’s task. Standalone numeric tokens were removed for consistency.

3.1.5 Whitespace and Formatting Normalization

After completing the above steps, extra whitespace was trimmed, and formatting inconsistencies were addressed. This included collapsing multiple spaces into one and stripping leading/trailing spaces. The result is a cleaned, lowercased text string for each article, ready for vectorization.

3.2 Label Encoding

The dataset includes categorical labels classifying articles as “REAL” or “FAKE.” To facilitate binary classification, these labels were encoded as integers: REAL as 0 and FAKE as 1. This encoding ensures compatibility with machine learning libraries such as scikit-learn and PyTorch, enabling seamless integration with classification algorithms.

3.3 Train-Validation-Test Split

To evaluate model performance effectively, the dataset was divided into three distinct subsets: training, validation, and testing. A common split ratio is 70% for training, 20% for validation, and 10% for testing. Stratified partitioning was employed to ensure that each subset maintains a balanced distribution of labels, preventing data leakage and enabling robust performance assessment. Preprocessing was applied equally to both training and test sets to avoid any subtle information bleed. Additionally, the training set was shuffled during training to eliminate potential order effects.

4. Methodology

We developed and assessed six distinct machine learning pipelines for fake news detection, ranging from straightforward linear models that utilize sparse text features to advanced neural network architectures that leverage dense embeddings and transformer-based approaches.

4.1 Traditional Supervised Classifiers

4.1.1 Bag-of-Words + Logistic Regression

The Bag-of-Words (BoW) representation is a simple yet effective approach for text classification tasks, including fake news detection. In this pipeline, each article is transformed into a high-dimensional vector of word frequencies, where the order of words is disregarded, and only the count of each word in the vocabulary is considered. This method captures distinctive word usage patterns that often differentiate fake news from real news. For instance, certain propaganda phrases or clickbait terms may appear more frequently in fake news, making them identifiable through this representation.

After preprocessing the text, we constructed a vocabulary consisting of all words appearing in the training set. Each article was then converted into a sparse vector, where each dimension corresponds to a word in the vocabulary, and the value represents the frequency (count) of that word in the article. For example, if the word “election” appears three times in an article, the corresponding feature for “election” would have a value of 3 in the article’s vector. To manage dimensionality, we limited the vocabulary to the top 10,000 most frequent words/terms in both BoW and TF-IDF vectorizers, which still accounted for the vast majority of word occurrences in the corpus. The resulting feature vectors are high-dimensional (10,000 dimensions) but very sparse, as each article typically contains only a small subset of all possible words.

For classification, we employed a Logistic Regression (LR) model, a linear model that assigns a weight to each input feature (word) to predict the probability of an article being real or fake. Logistic regression is computationally efficient, fast to train, and provides interpretable results. By inspecting the learned weights, we can identify which words contribute most to the classification decision—words with high positive weights indicate real news, while those with negative weights suggest fake news. This interpretability offers valuable insights into the model’s decision-making process.

To prevent overfitting in the large feature space, we applied L1 regularization, tuning the regularization strength on a validation split. The model was optimized using gradient descent to minimize binary cross-entropy loss. While BoW + LR serves as a baseline in our study, it has demonstrated competitive performance in previous research on fake news detection tasks (Israt Jahan et al., 2024). However, its limitations lie in its inability to capture semantic understanding or word order, which may restrict its effectiveness when deceptive writing closely mimics the style of legitimate news.

4.1.2 TF-IDF + Logistic Regression

In this pipeline, we transform the text using Term Frequency-Inverse Document Frequency (TF-IDF), replacing raw word counts with a weighted measure that reflects the importance of words within a document relative to their rarity across the corpus. TF-IDF downweights common words like “the” or “is,” which appear frequently but carry little discriminatory power, while upweighting rare but informative terms. For example, a specific term like “Pizzagate” will receive a high TF-IDF score in articles where it appears, whereas ubiquitous words like “the” will have scores close to zero. Using scikit-learn’s `TfidfVectorizer`, we limited the vocabulary to the top 5,000 terms and applied sublinear TF scaling and IDF smoothing for robustness. The resulting 5,000-dimensional feature vectors represent TF-IDF scores rather than raw counts.

For classification, we employed Logistic Regression (LR) with L1 regularization, optimized using gradient descent to minimize binary cross-entropy loss. This setup mirrors the BoW + LR pipeline but leverages TF-IDF’s ability to highlight discriminative terms, improving class separation. For instance, words like “conspiracy” that are frequent in certain articles but rare overall receive higher weights, enhancing their impact on classification. Logistic Regression remains interpretable, allowing us to analyze word weights to understand model decisions

4.2 Word Embeddings and Neural Networks

4.2.1 GloVe + Logistic Regression

In this pipeline, we transition from sparse representations like Bag-of-Words (BoW) and TF-IDF to dense semantic representations using GloVe (Global Vectors for Word Representation). Unlike BoW and TF-IDF, which treat words as independent features, GloVe embeddings capture semantic relationships between words by mapping each word to a fixed-dimensional vector learned from a large corpus. Specifically, we utilized the 100-dimensional GloVe vectors trained on 6 billion tokens from Wikipedia and Gigaword (the “glove.6B.100d” dataset). These embeddings encode meaningful relationships; for instance, “government” is closer in the vector space to “administration,” and “election” is near “vote.”

To represent an entire news article, we adopted a simple aggregation approach: averaging the GloVe vectors of all words in the article after preprocessing, stopword removal, and filtering out unknown terms. This produces a single 100-dimensional vector that serves as a rough semantic summary of the article's content. While averaging sacrifices word order and some nuanced contextual information, it effectively reduces noise and amplifies the core topic signal. For example, fake news articles might cluster in a specific region of the embedding space distinct from real news.

For classification, we used Logistic Regression (LR) with L1 regularization, optimized to minimize binary cross-entropy loss. The input features are now dense 100-dimensional vectors instead of high-dimensional sparse representations. With only 100 features, overfitting is less of a concern, and training is extremely fast (<1 second). LR attempts to find a linear decision boundary in the embedding space that separates real from fake news, effectively identifying regions where fake or real articles tend to cluster. Handling out-of-vocabulary (OOV) words was straightforward. Any word not present in the GloVe vocabulary was ignored. Fortunately, GloVe's extensive coverage ensured that the vast majority of words in our dataset had corresponding embeddings.

4.2.2 GLOVE + LSTM

In this pipeline, we employ a Long Short-Term Memory (LSTM) network to capture word order and contextual relationships, which are lost in simpler methods like averaging embeddings. LSTMs, a type of recurrent neural network (RNN), are well-suited for learning long-range dependencies in sequential data, making them ideal for tasks where context and phrasing matter.

The model architecture begins with an embedding layer initialized using 100-dimensional GloVe vectors, which provide dense semantic representations for each word. We pre-compute these GloVe embeddings for each token and feed them into an LSTM, processing up to 100 tokens per article (with padding or truncation to ensure a consistent length of 100). The final hidden state of the LSTM serves as a learned summary of the entire article, which is then passed through a dropout layer (rate 0.5) to prevent overfitting. Finally, this representation is fed into a dense layer with sigmoid activation for binary classification (real or fake).

We implemented the model in PyTorch, training it on a GPU for efficiency. Training used binary cross-entropy loss and the Adam optimizer with an initial learning rate of 0.001. Early stopping was applied to halt training if validation loss did not improve for two consecutive epochs, preventing overfitting on the relatively small dataset. Dropout was also applied to embeddings and LSTM outputs for regularization.

The LSTM excels at capturing patterns such as phrases, emphasis, and negations—e.g., distinguishing "not a hoax" from "a hoax"—which simpler models might miss. It can also identify meaningful phrases like "is a hoax" or "According to the FBI," adjusting embeddings and weights to better fit the classification task. For instance, the presence of words like "hoax" may drive predictions toward fake news.

However, LSTMs are computationally intensive, requiring significantly more training time than logistic regression models (minutes per epoch). Without attention mechanisms, capturing very long-range dependencies remains challenging. While alternative pooling strategies (e.g., averaging LSTM outputs) yielded similar results, interpretability is lower compared to simpler models, requiring advanced techniques like LIME to understand predictions.

4.3 Transformer-based Approach

4.3.1 BERT

We build on Hugging Face's pre-trained BERT-Base-Uncased model (12 layers, hidden size 768, ~110 M parameters) by fine-tuning it end-to-end on our binary classification task. Each input is the preprocessed news text (we lowercase, strip URLs/mentions/hashtags, remove punctuation and digits, and collapse whitespace), which we then feed to BERT's WordPiece tokenizer without additional stopword removal. We truncate or pad every example to a maximum of 128 tokens (adding [CLS] and [SEP] as required), since this length suffices to capture titles and the bulk of the article content under our GPU-memory constraints.

On top of BERT's pooled [CLS] representation, we add a single linear layer (768 → 2) and train using the model's built-in softmax + CrossEntropyLoss. We optimize with AdamW ($lr = 2\times10^{-5}$) and a linear learning-rate decay over 5 epochs. On top of that, we fine-tune all BERT parameters plus the classification head for 5 epochs (batch size = 16), saving the checkpoint that yields the best validation accuracy.

BERT's bidirectional transformer architecture allows it to consider context from both directions, capturing nuanced linguistic patterns that simpler models might miss. For example, it can recognize subtle cues such as skepticism in tone or inconsistencies in narratives that suggest fabrication. Recent research has demonstrated the effectiveness of transformer-based models like BERT in misinformation detection tasks (Lazer et al., 2018), often achieving state-of-the-art performance by identifying complex relationships and uncommon phrasing indicative of fake news.

Despite its strengths, BERT comes with trade-offs. Its high computational cost and slower inference times make it less practical for resource-constrained environments compared to simpler models like logistic regression. Additionally, BERT is inherently less interpretable due to its black-box nature, though techniques like attention weight analysis or explainability methods (e.g., LIME) could provide insights into its decision-making process. In contrast, logistic regression models allow direct inspection of feature importance, offering greater transparency at the expense of performance.

4.3.2 Sentence Transformer + Logistic Regression

In this pipeline, we combine the semantic capabilities of transformer-based models with the simplicity and efficiency of a linear classifier. We use Sentence-BERT (SBERT), a modified version of BERT that generates fixed-length sentence embeddings optimized for semantic similarity and text classification tasks (arxiv.org). Specifically, we employ the `all-MiniLM-L6-v2` model from the SentenceTransformers library, which outputs 384-dimensional embeddings. This distilled version of BERT is compact (6 layers) yet retains strong contextual understanding.

For each news article, we concatenated the title and body text and fed it into the SBERT model to produce a 384-dimensional vector representation. Unlike simpler methods like averaging GloVe embeddings or using BERT's [CLS] token, these embeddings are specifically designed to capture the overall meaning of the text in a high-dimensional semantic space. Importantly, we treated SBERT as a fixed feature extractor, avoiding fine-tuning to save computational resources and accelerate training. We then trained a logistic regression classifier (L1-regularized) on these embeddings to predict whether an article was real or fake. This two-stage approach involves using a pre-trained model to generate semantically rich features, followed by a simple linear classifier for predictions.

We expect the Sentence-BERT embeddings to effectively capture high-level distinctions in writing style and content between fake and real news. In this approach, logistic regression simply needs to establish a linear decision boundary within the high-dimensional semantic space provided by the embeddings. This method offers significant advantages, including rapid training—since the computationally intensive work is handled by the pre-trained SBERT model—and lower resource demands during inference compared to fine-tuning a full BERT model. While it may not match the peak performance of fine-tuned BERT, it achieves strong results by leveraging transformer-based representations. Additionally, the use of logistic regression allows for straightforward inspection of classifier weights in the embedding space, though interpreting these weights is less intuitive than analyzing word-based features.

5. Results

5.1 Overview

After training all models on the training set, we evaluated their performance on the test set (606 unseen news articles). The performance metrics used for evaluation are Accuracy, Precision, Recall, and F1 Score. In the context of this binary classification, we define these metrics with fake news being the positive class (for computing precision and recall, though we also report overall accuracy and macro-averaged F1 which is the same as the F1 for positive in a balanced dataset).

- **Accuracy**: The percentage of articles (both real and fake) correctly classified by the model out of the total number of test articles.
- **Precision (for the Fake class)**: The proportion of articles labeled as "fake" by the model that are actually fake.
- **Recall (for the Fake class)**: The percentage of actual fake news articles correctly identified by the model, calculated.
- **F1 Score** : The harmonic mean of precision and recall for the fake class, providing a balanced measure of the model's performance. In this balanced dataset, the F1 score for fake and real classes is similar, and the reported value represents the macro-average F1.

Table 1. Performance of Different Models on Fake News Detection (test set).

Model	Accuracy	Precision	Recall	F1
1. BoW + LR	0.93	0.91	0.95	0.93
2. TF-IDF + LR	0.94	0.94	0.93	0.94
3. GloVe + LR	0.84	0.84	0.86	0.85
4. GloVe + LSTM	0.88	0.88	0.87	0.88
5. BERT	0.93	0.95	0.92	0.93
6. Sentence-BERT + LR	0.85	0.85	0.87	0.86

The Bag-of-Words (BoW) + Logistic Regression (LR) model emerged as the top performer in terms of recall, achieving 95%, which underscores its effectiveness in identifying fake news and minimizing false negatives. This makes it particularly valuable for applications where failing to detect fake news could have serious consequences. While the TF-IDF + Logistic Regression and fine-tuned BERT models also demonstrated strong overall performance, with accuracies of 94% and 93%, respectively, their recall values were slightly lower, indicating a trade-off between precision and recall. Meanwhile, GloVe-based models—both the LR and LSTM variants—and Sentence-BERT showed comparatively weaker results, highlighting the limitations of generalized embeddings when it comes to capturing the nuanced features necessary for detecting misinformation.

The choice of text representation played a critical role in determining model performance. TF-IDF weighting proved to be more effective than raw counts, emphasizing the importance of factoring in word rarity. Contextual models, such as LSTM and transformer-based approaches like BERT and Sentence-BERT, significantly outperformed simpler methods like GloVe averaging. These advanced models delivered higher accuracy by leveraging contextual understanding and transfer learning, demonstrating their ability to capture subtle linguistic patterns that simpler models might miss. To better understand the behavior of each model, confusion matrices were generated, providing deeper insights into their strengths and weaknesses across different classification scenarios. This analysis reinforces the value of transformer-based architectures in achieving state-of-the-art results for fake news detection.

5.2 Confusion Matrices and Model Insights

Each confusion matrix highlights the number of correctly and incorrectly classified real and fake news articles. Key observations include:

- **BoW + LR**: Achieved a recall of 95%, indicating strong performance in identifying fake news, but exhibited slightly higher false positives (9%). This suggests that the model may misclassify some legitimate articles containing unusual or sensational terms as fake.
- **TF-IDF + LR**: Improved over BoW, with only 234 errors per class (468 total errors vs. 624 for BoW). The reduction in errors suggests that down-weighting common terms helped the model rely more on distinctive words, reducing confusion on articles with filler words.
- **GloVe + LR**: Showed larger errors (428 vs. 234 in the TF-IDF case), confirming that averaged embeddings are less discriminative. The model struggled to distinguish between topics in the semantic space, leading to misclassifications.
- **GloVe + LSTM**: Reduced errors compared to GloVe + LR (273 vs. 428 per class), demonstrating the value of sequence modeling. The LSTM captured context more effectively, resolving ambiguities present in simpler models.
- **Fine-Tuned BERT**: Achieved near-perfect performance, with only 156 errors per class. This underscores BERT’s ability to capture subtle linguistic patterns and contextual understanding.
- **SBERT + LR**: Achieved recall of 87% and accuracy of 85%, delivering near state-of-the-art performance without fine-tuning. This validates SBERT as an efficient feature extractor, retaining much of the information captured by fine-tuned BERT.

6. Discussion

6.1 Key Insights

The evaluation provided critical insights into the effectiveness of various machine learning models for fake news detection, highlighting distinct strengths and trade-offs across approaches. Notably, the Bag-of-Words (BoW) + Logistic Regression (LR) model demonstrated exceptional recall (95%), underscoring its ability to detect subtle lexical cues that distinguish fake news. Despite its simplicity, this model excelled in minimizing false negatives—a key priority in practical misinformation monitoring scenarios where failing to identify fake news can have significant consequences.

The TF-IDF + LR model achieved a well-rounded performance with high accuracy (94%) and robust recall (93%), illustrating the advantages of term-weighting techniques in emphasizing distinctive lexical patterns. By down-weighting common terms and focusing on rarer, more discriminative features, this approach effectively reduced ambiguity in classification. In contrast, the fine-tuned BERT model, while achieving slightly lower recall (92%), demonstrated superior precision (95%). This highlights BERT’s strength in capturing nuanced contextual relationships, making it particularly valuable for cases requiring deeper semantic understanding or secondary verification.

In comparison, GloVe embedding-based models exhibited weaker performance, revealing the limitations of relying solely on averaged semantic embeddings. While incorporating sequential context through an LSTM improved results relative to simple averaging, these models still fell short of the performance achieved by TF-IDF and transformer-based methods. The Sentence-BERT + LR model delivered moderate recall (87%), capturing some contextual nuances but ultimately lagging behind simpler lexical models. This discrepancy may stem from the fixed nature of pre-trained embeddings, which lack the adaptability of task-specific fine-tuning and may struggle to fully capture domain-specific characteristics of fake news.

These findings underscore the importance of selecting the right approach based on the specific priorities of the application—whether minimizing false negatives, achieving high precision, or balancing both. Transformer-based models like BERT offer state-of-the-art performance, but simpler methods such as BoW + LR and TF-IDF + LR remain highly competitive, particularly in resource-constrained or interpretability-focused scenarios.

6.2 Trade-offs and Recommendations

Given the project's emphasis on minimizing false negatives, ensuring that fake news is detected as effectively as possible, the Bag-of-Words (BoW) + Logistic Regression (LR) model emerges as the most suitable choice for primary deployment. This model achieved an impressive recall of 95%, making it highly effective at identifying fake news articles and reducing the risk of overlooking potentially harmful misinformation. While its precision is slightly lower at 91%, this trade-off results in a manageable increase in false positives, which can be addressed through subsequent human review or by integrating a secondary verification step using more precision-focused models like fine-tuned BERT. This layered approach ensures that the system remains robust while maintaining a strong focus on recall, which is critical in high-stakes applications where failing to detect fake news could have significant consequences.

For scenarios where computational resources or scalability are a concern, the TF-IDF + LR model offers a compelling alternative. With a balanced performance across accuracy (94%), recall (93%), and precision (94%), this model strikes an optimal compromise between effectiveness and efficiency. Additionally, its interpretability makes it particularly valuable in contexts where transparency and explainability are important, such as when justifying decisions to stakeholders or users. The use of term frequency-inverse document frequency (TF-IDF) weighting enhances the model's ability to focus on distinctive terms, reducing confusion caused by common filler words. As a result, TF-IDF + LR is well-suited for large-scale deployments where computational cost and ease of interpretation are prioritized without significantly compromising performance.

On the other hand, while fine-tuned BERT demonstrates state-of-the-art capabilities with an accuracy and recall of 93% and precision of 95%, its computational demands make it less practical for frontline classification in resource-constrained environments. BERT’s deep contextual understanding allows it to capture nuanced linguistic patterns that simpler models might miss, making it ideal for handling complex or borderline cases. However, its reliance on GPU acceleration and longer inference times render it better suited as a secondary verifier rather than a primary classifier. By deploying BERT in this capacity, organizations can leverage its strengths for challenging instances while relying on lighter models like BoW + LR or TF-IDF + LR for bulk processing.

In short, the choice of model should align with the specific priorities and constraints of the deployment scenario. For applications where recall is paramount, BoW + LR serves as the top choice due to its exceptional ability to minimize false negatives. When balancing performance with computational efficiency and interpretability, TF-IDF + LR provides a versatile solution. Meanwhile, fine-tuned BERT can be reserved for high-precision secondary verification or specialized tasks requiring deeper contextual analysis. This tiered strategy not only maximizes detection efficacy but also ensures adaptability across diverse operational settings.

6.3 Generalization and Robustness

While Bag-of-Words (BoW) and TF-IDF models demonstrated strong performance in detecting fake news, their reliance on static lexical features may limit their adaptability to rapidly evolving misinformation tactics. These models are particularly vulnerable to shifts in language use, such as the emergence of new slang, idioms, or trending phrases that were not present in the training data. For instance, if fake news creators begin using novel terms or framing techniques to evade detection, BoW and TF-IDF models might fail to recognize these changes due to their inability to generalize beyond the specific patterns they were trained on. This brittleness underscores the importance of regularly updating and retraining such models to maintain their effectiveness in dynamic environments.

In contrast, transformer-based models like BERT exhibit a deeper contextual understanding, enabling them to better handle linguistic nuances and adapt to evolving language patterns. By leveraging subword tokenization and contextual embeddings, BERT can infer the meaning of previously unseen terms and capture subtle semantic relationships that simpler models might miss. This inherent flexibility makes BERT particularly well-suited for environments where misinformation tactics are constantly changing. However, while BERT’s architecture provides a theoretical advantage in robustness, its real-world performance in highly dynamic contexts remains an area for further empirical validation. Testing the model against datasets featuring emerging linguistic trends or adversarial examples would be critical to fully assess its generalization capabilities.

Another consideration is the trade-off between complexity and adaptability. While BERT offers superior robustness, its computational demands may pose challenges for real-time applications or large-scale deployments. In scenarios where computational resources are constrained, hybrid approaches—such as combining lightweight models with periodic updates from more advanced models—could strike a balance between efficiency and adaptability. Overall, ensuring robustness in fake news detection requires not only selecting the right model but also implementing strategies to address the ever-changing nature of misinformation.

6.4 Ethical and Practical Considerations

6.4.1 Minimizing False Negatives

Disinformation campaigns inevitably evolve to outsmart detection systems. For instance, if a classifier depends heavily on certain buzzwords or stylistic signals, bad actors will learn to avoid those markers and slip past filters. This ongoing “cat-and-mouse” interplay means our models must be continually retrained and fine-tuned. Transformer-based architectures such as BERT offer greater resilience, since they leverage deep contextual cues rather than surface-level patterns alone. Still, attackers may craft ostensibly factual content that subtly distorts the truth—posing a serious challenge even for state-of-the-art models. Maintaining robustness against these sophisticated manipulations demands investment in adversarial training strategies and a commitment to proactive model updates.

6.4.2 Avoiding False Positives

False positives, i.e., misidentifying legitimate news as fake, carry their own risks by undermining trusted sources and eroding confidence in the detection system. Even a seemingly low false positive rate of 4–5%, as seen in our strongest models, can translate into thousands of wrongly flagged articles at scale. Imagine a respected news outlet's coverage or time-sensitive briefs being mislabeled, such errors could have serious reputational and societal repercussions. To guard against this, we should introduce secondary review workflows or human-in-the-loop checks for any content the model flags. In addition, we can fine-tune classification thresholds to the context, prioritizing precision over recall in situations where avoiding false positives is critical, to strike the right balance between safety and coverage.

6.4.3 Adversarial Adaptation

Fake news creators continually tweak their tactics once they discover how our detectors work. If a system leans on specific keywords or writing patterns, attackers will simply avoid or disguise those signals to slip by. This ongoing "cat-and-mouse" cycle means our detection models must be in a state of constant evolution of being regularly retrained and fine-tuned. Transformer architectures like BERT help, since they draw on deep contextual understanding rather than just surface features, but even they can be fooled by material that reads like legitimate reporting yet subtly twists the facts. Building true resilience requires a dedicated investment in adversarial training techniques and a disciplined regimen of proactive model updates.

6.4.4 Bias in Training Data

A model is only as good as the data it's trained on. When certain segments—whether political viewpoints, geographic regions, or subject areas—dominate the training set, the model can pick up unwanted biases. In practice, this might mean that content from particular outlets or in specific writing styles is flagged more often, even when it's entirely accurate. These imbalances not only compromise fairness but also chip away at the credibility of affected publishers and shake reader confidence. To prevent this, we must assemble training data that truly spans the full spectrum of news perspectives and topics. On top of that, periodic audits and bias reviews will keep our models honest and help maintain trust in their judgments.

6.4.5 Broader Ethical Considerations

Beyond technical challenges, the ethical implications of deploying a fake news detection system must be carefully considered:

- **Transparency**: Everyone from end-users to stakeholders needs clarity on how the system arrives at its judgments. Simple approaches like bag-of-words paired with logistic regression offer straightforward explainability, but deep models such as BERT can feel like "black boxes." We'll need interpretability aids, such as SHAP value breakdowns or attention-weight visualizations, to open the hood on these complex predictions.
- **Misuse Risks**: Left unchecked, an automated filter can become a tool for censorship or for silencing inconvenient viewpoints. Rather than positioning the detector as a gatekeeper, we should embed it as an assistive layer—flagging questionable content for human review and keeping final decision-making squarely in human hands.
- **Satire and Humor**: Satirical pieces and parodies play fast and loose with facts for comedic effect, not to deceive. Treating them as outright disinformation risks unjust takedowns and erodes the credibility of genuine content. Handling this nuance effectively may require a multi-class classification framework or dedicated model component trained specifically to detect comedic intent.

7. Conclusion

In this project, we built and benchmarked six machine-learning pipelines for fake-news detection on a Kaggle dataset, achieving strong performance across the board—from 93 % accuracy with a simple bag-of-words plus logistic regression model to 93 % with a fine-tuned BERT transformer. Our findings highlight the pivotal role of text representation: TF-IDF vectors and transformer-based embeddings consistently outshine raw token counts and averaged embeddings by capturing both lexical and contextual subtleties. While BERT delivers top accuracy, hybrid approaches such as SBERT combined with logistic regression offer an attractive trade-off between predictive power and computational cost.

We see three clear avenues for next steps. First, we'll broaden the classifier to handle multiple categories—such as separating satire from harmful falsehoods—and ensure it generalizes across different news domains by weaving in metadata cues. Second, we plan to layer in explainability tools and ensemble strategies to boost both transparency and resilience. Ultimately, our results highlight the real power of NLP-driven approaches to combat misinformation—so long as we keep refining our models to stay ahead of new tactics.

References

Israt Jahan, Md Nazmul Hasan, Syed Nurul Islam, Lima Akter, Md Khaledur Rahman Onik, Ashraful Islam, & Sm Mahamudul Hasan. (2024). Advanced machine learning techniques for fake news detection: A comprehensive analysis. Magna Scientia Advanced Research and Reviews, 12(2), 203–212. https://doi.org/10.30574/msarr.2024.12.2.0198

Lazer, D. M. J., Baum, M. A., Benkler, Y., Berinsky, A. J., Greenhill, K. M., Menczer, F., Metzger, M. J., Nyhan, B., Pennycook, G., Rothschild, D., Schudson, M., Sloman, S. A., Sunstein, C. R., Thorson, E. A., Watts, D. J., & Zittrain, J. L. (2018). The science of fake news. Science, 359(6380), 1094–1096. https://doi.org/10.1126/science.aao2998

News.csv. (n.d.). Retrieved April 17, 2025, from https://www.kaggle.com/datasets/antonioskokiantonis/newscsv

Sad milestone: Fake local news sites now outnumber real local newspaper sites in u. S. (n.d.). NewsGuard. Retrieved April 17, 2025, from https://www.newsguardtech.com/press/sad-milestone-fake-local-news-sites-now-outnumber-real-local-newspaper-sites-in-u-s/

Study: On Twitter, false news travels faster than true stories. (2018, March 8). MIT News | Massachusetts Institute of Technology. https://news.mit.edu/2018/study-twitter-false-news-travels-faster-true-stories-0308

van der Linden, S. (2022). Misinformation: Susceptibility, spread, and interventions to immunize the public. Nature Medicine, 28(3), 460–467. https://doi.org/10.1038/s41591-022-01713-6