Journal of Administrative and Economic Sciences • Vol 18, Issue 2 (2025) • p 779-795

A Comparative Machine Learning and Deep Learning Algorithms for Fake News Detection

Boulbaba Ben Ammar

Department of Computer Science, College of Computer, Qassim University, Buraydah • Saudi Arabia

b.benammar@qu.edu.sa

Abstract:

This study investigates the extent to which gradient-boosted decision trees can rival deep learning architectures in the task of text-based fake news detection. Employing a unified experimental pipeline that evaluates eleven models on a dataset of approximately 45,000 news articles using simple n-gram representations, we find that XGBoost achieves the highest classification accuracy (99.85%), marginally outperforming a multi-layer perceptron baseline (99.64%). Moreover, XGBoost demonstrates superior computational efficiency, with faster training times and easier deployment and interpretability. To ensure methodological robustness and avoid overstatement of results, we complement headline performance metrics with statistical significance testing, efficiency benchmarking, and feature-attribution analyses. A dedicated bias analysis reveals substantial topic- and source-related confounding within the benchmark dataset (e.g., "real" news disproportionately originating from wire services), highlighting that near-perfect in-dataset performance does not necessarily translate to reliable veracity detection in real-world scenarios. Cross-dataset validation further underscores this limitation, showing a marked drop in performance on external corpora (from ~99.8% to ~68%), thereby indicating reliance on dataset-specific shortcuts. Our findings offer three key recommendations for practitioners: (1) establish robust XGBoost baselines before deploying more computationally expensive deep learning models, particularly in resource-constrained or latency-sensitive settings; (2) conduct thorough cross-dataset evaluations to assess generalization capacity; and (3) perform feature audits to identify and mitigate spurious correlations. All code, dataset splits, and evaluation scripts are publicly released to promote reproducibility and enable rigorous future research.

Keywords: Fake news Detection; Text Classification; XGBoost; Deep Learning; Machine Learning; Bias Audits.

1. Introduction

The rise of the internet has substantially lowered the barriers to publishing, enabling the rapid and large-scale dissemination of fabricated information. This proliferation of false narratives now routinely outpaces the capacity of human fact-checkers, making automated detection systems an essential component of early warning and response infrastructures—particularly in contexts with significant societal implications such as public health, elections, and social stability. Within this domain, content-based text classifiers remain especially valuable because they operate solely on the textual content of articles and can function effectively even when network, user, or metadata information is unavailable.

Despite many proposed methods, performance gains in fake-news detection are often marginal and tightly coupled to specific benchmarks. Results on a single corpus rarely guarantee robustness across topics, sources, or time, which limits the external validity of existing conclusions.

Against this background, we revisit a practical question: when is deep learning actually necessary for content-based fake-news detection, and when do carefully tuned classical models suffice? We focus on a realistic setting in which only article text is available—no user, network, or metadata features—and ask whether the added complexity and computational cost of deep models are justified by consistent, substantial improvements.

To address this question, we design a reproducible, end-to-end experimental pipeline and systematically evaluate a diverse set of machine-learning and deep-learning models on a widely used fake/real news corpus. We enforce a like-for-like comparison (identical preprocessing, vectorization, and splits) to isolate the contribution of the learning algorithm itself and to distinguish genuine improvements from artifacts of the experimental setup.

Our results demonstrate that a well-optimized XGBoost classifier, leveraging sparse n-gram features, is capable of matching or exceeding the performance of deep neural baselines in a rigorously controlled setting. Moreover, XGBoost trains significantly faster, is easier to deploy, and offers greater interpretability—qualities that are especially valuable in operational contexts. We further present a comprehensive analysis of dataset confounding factors to avoid overstating generalization capabilities. Within the text-only detection scope we target—where large pre-trained models may be impractical, interpretability is essential, or computational resources are limited—we offer a clear and actionable recommendation: start with robust, tree-based ensemble methods built on well-defined n-gram representations, and only escalate to more complex architectures when cross-dataset evidence and operational requirements justify the added cost.

It is important to note that our deep learning baselines are trained from scratch and do not include large pre-trained transformer models such as BERT or RoBERTa. As a result, our conclusions should be interpreted specifically as a comparison of XGBoost versus from-scratch deep architectures, with efficiency and interpretability trade-offs explicitly quantified within this scope.

2. Related Work

Research on automated fake news detection has evolved along two principal trajectories: content-based models, which rely solely on textual information from the article itself, and context-aware approaches, which incorporate auxiliary signals such as user behavior, social-network structure, or propagation dynamics. This study remains within the content-only paradigm, where deployment is typically more straightforward and situations often arise in which social or network metadata are unavailable.

Early efforts in content-based detection demonstrated that relatively simple lexical and stylistic features—such as n-grams, punctuation usage, and psycholinguistic markers—combined with traditional classifiers like Naive Bayes or Support Vector Machines, can already deliver competitive performance. These findings suggest that fabricated and legitimate articles often exhibit distinct lexical and syntactic characteristics (Pérez-Rosas et al., 2018; Ahmed et al., 2017).

Subsequent advances shifted attention toward neural architectures capable of learning richer representations directly from text. Convolutional neural networks (CNNs) have been shown to capture local phrase patterns, while recurrent neural networks (RNNs), including LSTM and GRU variants, effectively model sequential dependencies. More recently, transformer-based encoders such as BERT have set new performance benchmarks by leveraging deep contextual understanding of language (Kaliyar et al., 2021, among others). However, these approaches introduce significant computational overhead and latency, which can limit their practicality in real-time or resource-constrained environments.

At the same time, a growing body of work has questioned whether such computationally expensive models are always necessary for text-only fake news detection. Studies emphasizing high-dimensional sparse features report that well-tuned tree-based ensembles, especially gradient-boosted decision trees such as XGBoost, can achieve comparable or superior performance relative to neural baselines while training more quickly, serving more efficiently, and offering greater interpretability (Grinsztajn et al., 2022; Shu et al., 2020). This line of research underscores the value of like-for-like experimental comparisons that control for preprocessing, representation, and evaluation variables to isolate the true contribution of model complexity.

Another significant thread of research examines the validity of reported results in the presence of dataset artifacts. Several reviews have documented topic and source confounding in commonly used corpora: articles labeled as “fake” are often political and originate from non-mainstream outlets, while “real” articles predominantly come from established news agencies and cover a broader range of topics. Such correlations risk training models to recognize topic or source cues rather than genuine indicators of veracity, thereby inflating accuracy metrics and compromising generalizability (Gruppi et al., 2022). Benchmark resources such as LIAR (Wang, 2017) for statement-level classification and FakeNewsNet (Shu et al., 2020) for article-level detection illustrate that cross-dataset performance typically drops sharply, emphasizing the need for careful bias auditing and robustness evaluation.

More recent surveys provide updated taxonomies of architectures, datasets, and open challenges, including Harris et al. (2024), Alshuwaier and Alsulaiman (2025), and Lv et al. (2025), who highlight unresolved issues in robustness, cross-dataset generalization, and multimodal detection.

For instance, Singh, Khan, and Meena (2023) recently reported strong performance using ensemble learning models (including XGBoost) on a fake news detection task, achieving high accuracy through TF-IDF features and careful hyperparameter tuning—results that align closely with our findings and further support the effectiveness of tree-based ensembles in this domain.

More recent work reflects a shift toward practical considerations in deployment. Interpretability has become increasingly important in fact-checking workflows, where transparent decision-making is often preferred over opaque models (Zhou et al., 2023). Classical baselines such as SVMs and Random Forests remain competitive, particularly when carefully tuned and applied to multilingual datasets (Zhou et al., 2022). Simultaneously, the rise of powerful generative models has elevated the challenge by enabling the creation of highly realistic synthetic news, further highlighting the importance of efficient and generalizable detectors that do not rely solely on topic or source cues (Zellers et al., 2023; Goldstein et al., 2023).

Transformer-based language models, beginning with BERT (Devlin et al., 2019) and lighter variants such as DistilBERT (Sanh et al., 2019), have driven much of the recent progress in fake news detection. While these models often achieve near-perfect in-dataset performance, numerous studies have shown that their effectiveness remains sensitive to dataset artifacts and domain shifts (Goldstein et al., 2023; Zellers et al., 2023). In parallel, gradient-boosted decision trees like XGBoost continue to perform competitively on sparse textual representations, excelling particularly in efficiency-constrained environments (Grinsztajn et al., 2022). Furthermore, topic and source confounds remain a persistent concern, as they can artificially inflate performance metrics and obscure the true veracity detection capability of a model (Baly et al., 2018; Gruppi et al., 2022). These challenges motivate the present study's emphasis on bias auditing, cross-dataset evaluation, and statistical significance testing.

Within this evolving landscape, our work contributes a comprehensive, apples-to-apples comparison of linear, probabilistic, instance-based, tree-based, and neural models under identical preprocessing, vectorization, data splits, and evaluation criteria. Consistent with prior evidence, we find that a carefully tuned XGBoost model leveraging n-gram features can match or exceed the performance of deep neural baselines while remaining simpler to interpret and deploy. This balance between performance and efficiency is especially relevant for production applications, and our evaluation framework complements headline performance metrics with bias analyses and external validity assessments to provide a more realistic understanding of model behavior.

3. Methodology

Our objective is to conduct a fair and rigorous comparison between classical machine learning algorithms and deep learning architectures under controlled and transparent

conditions. To this end, we standardize all key components of the experimental pipeline—including dataset curation, data splitting strategy, and text preprocessing—ensuring that any observed differences in performance can be attributed solely to the modeling approaches rather than variations in the underlying pipeline

3.1. System Architecture

Our proposed fake news detection framework adheres to a conventional data science pipeline, encompassing sequential stages from data collection and preprocessing to feature extraction, model training, evaluation, and deployment, as illustrated in Figure 1.

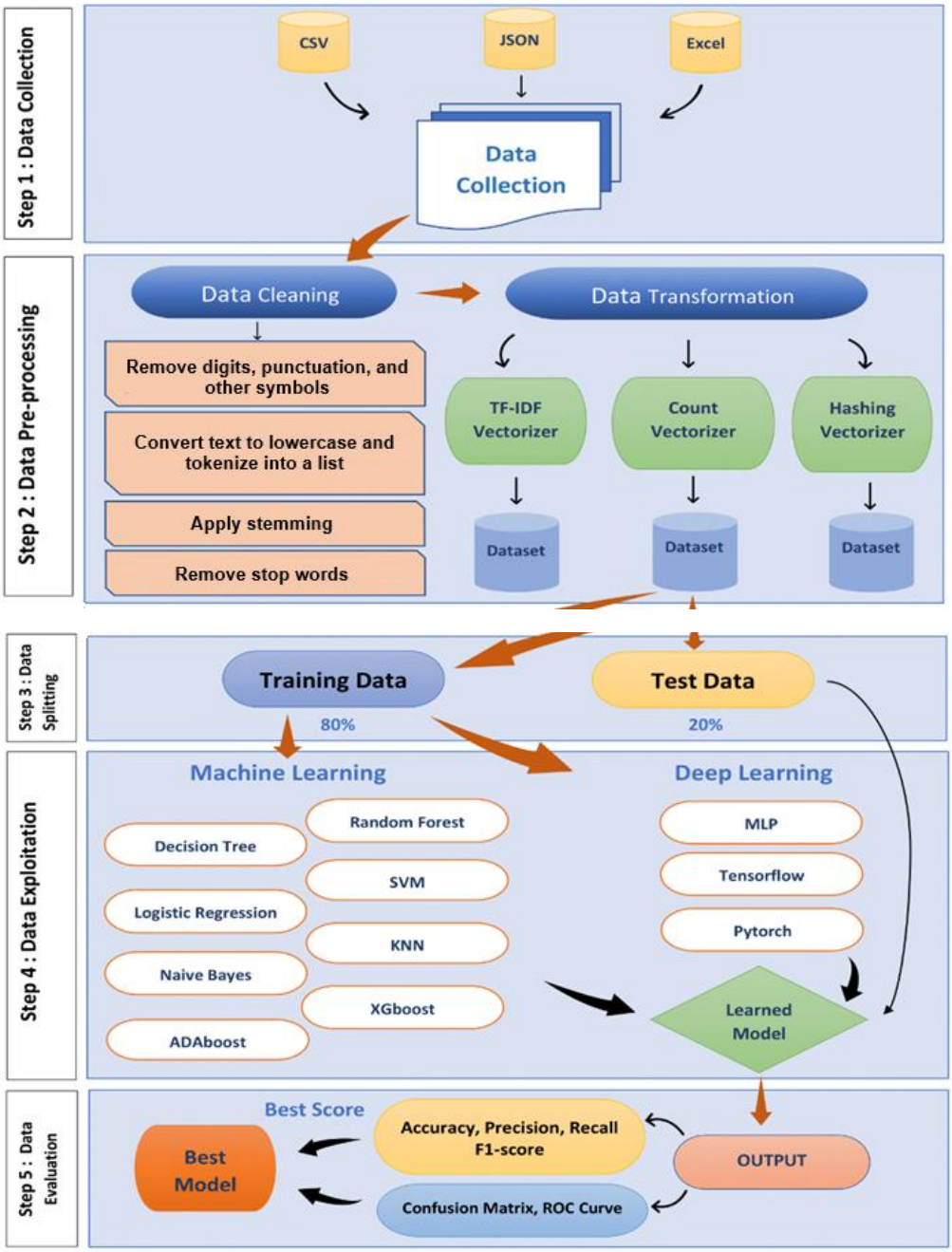


Figure 1: Process of fake news detection

3.2. Dataset and Bias Analysis

We employ the publicly available **Fake and Real News** corpus from Kaggle[1]. It is crucial to acknowledge from the outset the significant domain bias present in this dataset. The "True" articles are primarily sourced from Reuters, covering international and business news, whereas "Fake" articles originate from known hoax sites and are heavily skewed toward U.S. politics. This introduces a strong topic-domain bias, which may simplify the classification task from detecting deception to discerning topic and source. The distribution of subjects is shown in Figure 2.

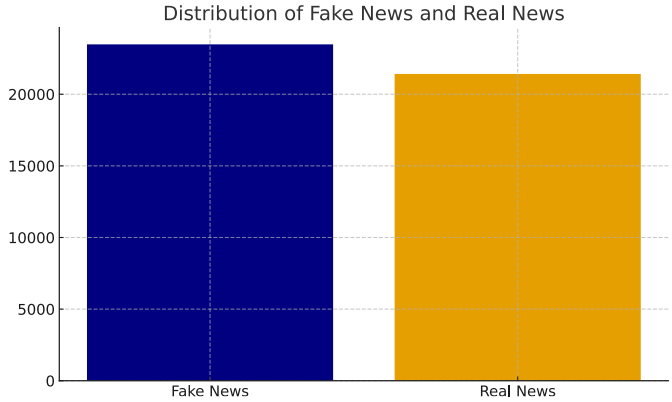
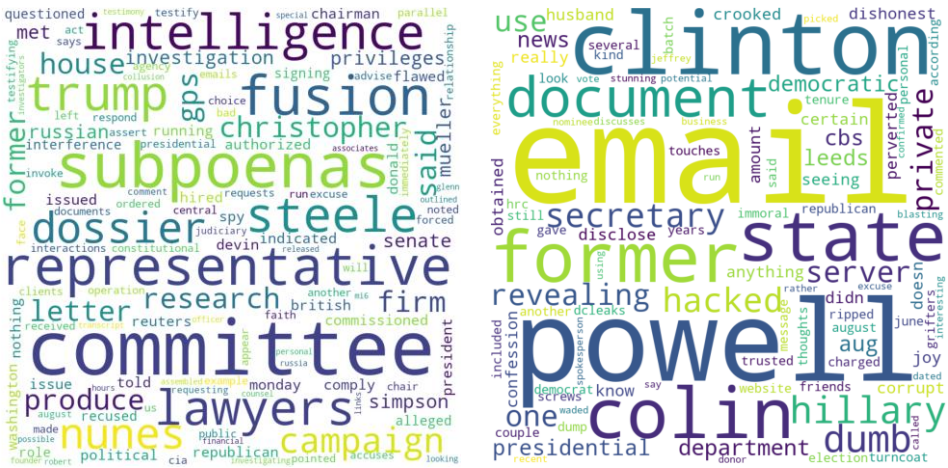


Figure 2: Dataset Distribution.

Figure 2 summarizes the label distribution of the corpus and shows that the dataset is approximately balanced across the two classes (fake vs. real). The bars for the training, validation, and test partitions follow the same proportions due to stratified splitting, indicating that no specialized imbalance treatment (e.g., oversampling, undersampling, or class reweighting) is required for the main experiments. We therefore train models on the native distribution and report standard metrics; class-weight sensitivity is included only as a robustness check.



(a) WordCloud for fake news (b) WordCloud for true news

Figure 3: WordCloud

[1] https://www.kaggle.com/clmentbisaillon/fake-and-real-news-dataset

Figure 3 displays class-specific word clouds that provide a qualitative glimpse of the most frequent tokens in the training set after basic normalization and stopword removal. The visualization highlights salient unigrams that tend to characterize each label, offering an intuitive sense of topical and stylistic differences between fake and real articles. These word clouds are used strictly for exploratory analysis and reporting; they are not features in our models. Because frequency-based visuals can reflect topic prevalence as much as veracity cues, we interpret them cautiously and complement them with the quantitative comparisons and bias checks presented in subsequent sections.

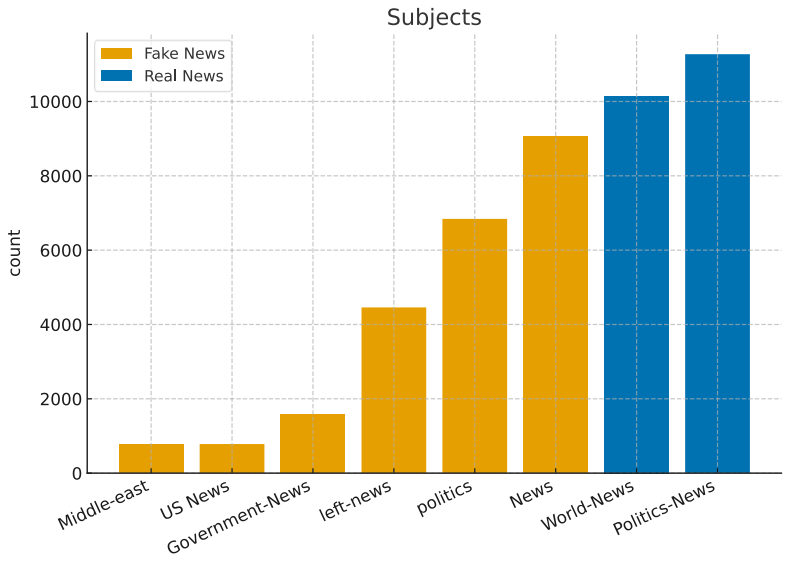


Figure 4: Distribution of data by Topics

Figure 4 presents the distribution of articles by topical category (e.g., politics, world, health, business, technology) for the full corpus and per class. The histogram reveals that some topics are more prevalent than others and that their proportions can differ across labels—most notably a higher share of politics compared with several other categories. This asymmetry is informative for context but also cautions against topic-driven shortcuts: models might exploit topical prevalence rather than genuine veracity cues. We therefore use stratified splits and report bias diagnostics alongside headline metrics, and we interpret results with this topic structure in mind.

3.3. Data Processing

Text pre-processing is crucial for model performance. Our cleaning pipeline, depicted in Figure 5, involved:

1. **Cleaning:** Removing non-letter characters, digits, and punctuation.
2. **Case Folding:** Converting all text to lowercase.
3. **Stemming:** Reducing words to their root form (e.g., "playing" -> "play").
4. **Stop-word Removal:** Eliminating common but insignificant words (e.g., "the", "and").

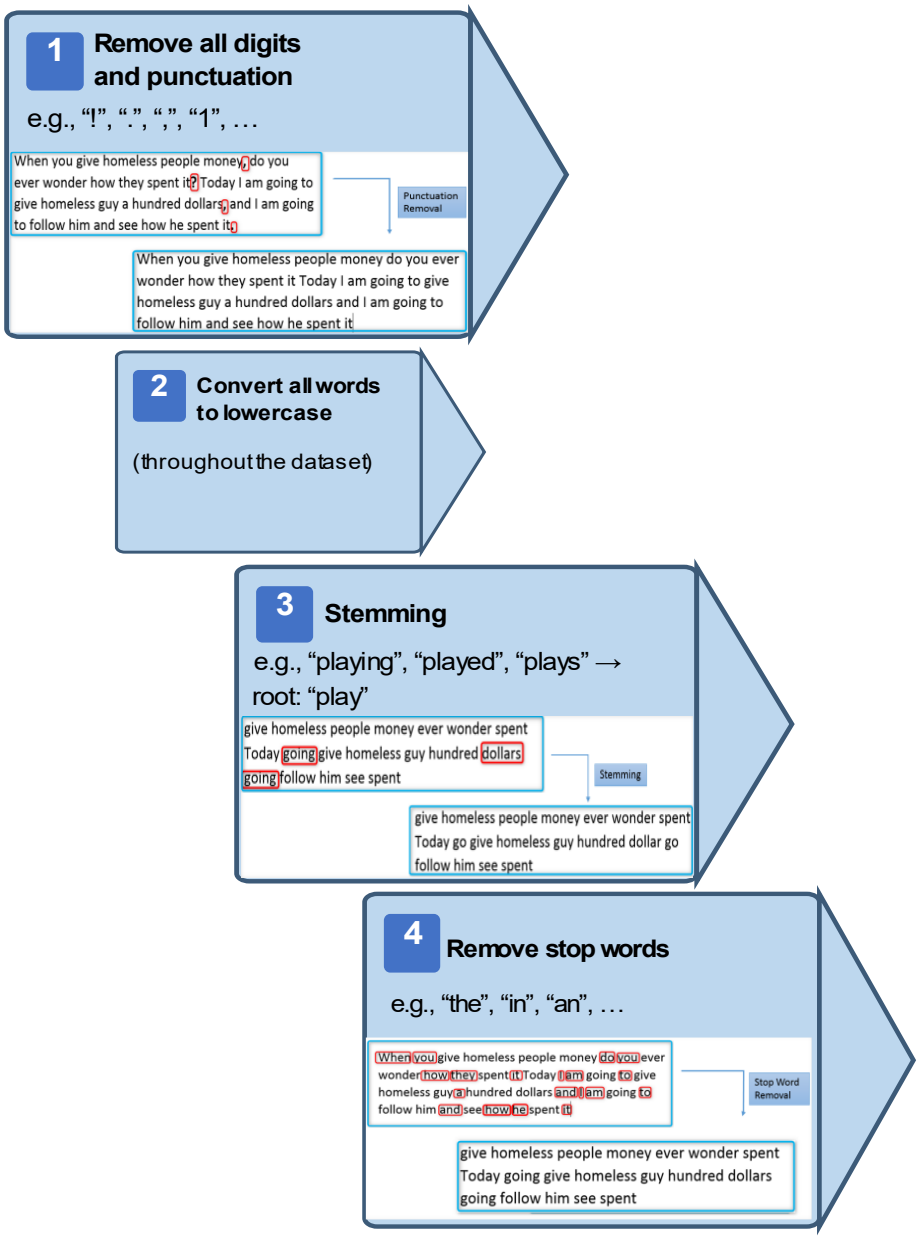


Figure 5: Data Pre-Processing: Cleaning pipeline

We adopt a modular, end-to-end architecture for content-based fake news detection (Fig. 1) that clearly separates offline training from online inference. Incoming articles (title and body) first pass through a validation and normalization layer that enforces schema constraints, removes duplicates, and applies deterministic text cleaning. The cleaned text is then mapped to sparse n-gram features via interchangeable, versioned vectorizers to prevent leakage. After cleaning—removing non-alphabetic characters, lowercasing, Porter stemming, and filtering NLTK English stop words—we evaluate three representations with fixed, validated settings:

(i) CountVectorizer with max_features = 50,000, ngram_range = (1, 2), and min_df = 2;

(ii) TfidfVectorizer with the same parameters and sublinear_tf = True; and

(iii) HashingVectorizer with n_features = 2^{16} and norm = 'l2'. These configurations were selected in preliminary experiments to balance sparsity, dimensionality, and signal retention.

3.4. Model Training and Evaluation

To enable a comprehensive and fair comparison, we evaluate representative algorithms from the principal families used in text classification. Linear models—Logistic Regression and linear SVM—test whether the sparse n-gram feature space renders the fake vs. real distinction largely linearly separable. Logistic Regression affords coefficient-level interpretability, whereas SVMs are well suited to very high-dimensional spaces. Tree-based models—Decision Tree, Random Forest, and XGBoost—capture non-linear interactions via hierarchical decision rules. A single Decision Tree is highly interpretable but variance-prone; Random Forest mitigates overfitting by averaging decorrelated trees; and XGBoost applies sequential gradient boosting to correct residual errors, a strategy that consistently performs well on tabular and sparse text representations. Probabilistic and instance-based baselines—Multinomial Naive Bayes and k-Nearest Neighbors—serve as classical reference points: Naive Bayes leverages word-frequency regularities under a conditional-independence assumption, while k-NN is a non-parametric method driven by local similarity in the feature space. Finally, neural models—a Multi-Layer Perceptron and lightweight TensorFlow/PyTorch baselines—probe whether deeper non-linear function approximators recover interactions that classical models may miss.

3.4.1. Training protocol

All models are trained within the same pipeline to ensure comparability: deterministic preprocessing, fixed tokenization, and one of three representations—Count, TF-IDF, or Hashing—with identical n-gram ranges and shared vocabulary/hashing sizes. To prevent leakage, vectorizers are fit exclusively on training folds and reused unchanged for validation and test. Hyperparameters are tuned via stratified 5-fold cross-validation on the training set with fixed seeds; early stopping is enabled where supported (e.g., XGBoost, MLP). When relevant, class weights (or scale_pos_weight for XGBoost) reflect the native label distribution but do not alter the splits. Compute budgets—including number of trials, maximum iterations/trees, and patience—are harmonized across families to keep search effort comparable.

3.4.2. Evaluation protocol

Model selection is performed on a held-out validation set; final results are reported on the untouched test set. Accuracy is the primary metric, complemented by precision, recall, F1, confusion matrices, and probability calibration diagnostics where applicable. We additionally report efficiency indicators—training time, inference latency, and model size—to characterize deployability. When top models are close, we apply paired significance testing (e.g., McNemar's test) on the test predictions to determine whether differences are statistically meaningful. Robustness checks include seed sensitivity,

representation ablations, and source-aware splits (where metadata permit), ensuring that conclusions do not hinge on a particular partitioning scheme or feature configuration.

3.5. Limitations and Scope

We restrict neural baselines to from-scratch architectures and do not fine-tune pre-trained transformers (e.g., BERT, RoBERTa). Accordingly, all comparative claims should be read as "XGBoost vs. from-scratch deep baselines" under identical preprocessing, tuning budgets, and evaluation protocols. Our study also focuses on text-only inputs and a single benchmark with known topic/source confounds; thus, near-ceiling in-dataset performance should not be interpreted as evidence of cross-dataset generalization. To avoid over-claiming, we report multi-seed results with confidence intervals, conduct paired significance tests, and complement headline metrics with feature-attribution and error analyses. Future work will incorporate pre-trained transformers, multimodal signals, and broader cross-dataset validation.

4. Results and Discussion

4.1. Performance of Machine Learning Models

Table 1 summarizes the performance of eight machine-learning models across the evaluated vectorization schemes; for each model we report its best configuration, and per-model maxima are highlighted in the table. XGBoost attains the highest accuracy (99.85%) using a CountVectorizer on article text. AdaBoost, Decision Tree, and Logistic Regression are closely competitive (≥ 99.65%), while SVM and Random Forest remain robust (≥ 99.20%). In contrast, Multinomial Naive Bayes and k-Nearest Neighbors underperform, with peak accuracies of 95.04% and 90.04%, respectively. Notably, k-NN exhibits comparatively high recall but markedly lower precision, indicating a propensity toward false positives.

Table 1: Comparative performance of machine-learning models (best vectorizer per model)

Model	Vectorizer	Accuracy	Precision	Recall	F1-score
XGBoost	CountVectorizer (Text)	0.9985	0.9989	0.9980	0.9985
AdaBoost	CountVectorizer (Text)	0.9969	0.9961	0.9980	0.9971
Decision Tree	CountVectorizer (Text)	0.9974	0.9978	0.9972	0.9975
Logistic Regression	CountVectorizer (Text)	0.9965	0.9968	0.9965	0.9966
SVM	CountVectorizer (Text)	0.9957	0.9957	0.9961	0.9959

Random Forest	TfidfVectorizer (Title+Text)	0.9929	0.9919	0.9946	0.9932
Naive Bayes	CountVectorizer (Text)	0.9504	0.9484	0.9564	0.9524
k-NN	HashingVectorizer (Title+Text)	0.9004	0.8368	0.9684	0.8978

4.2. Performance of Deep Learning Models

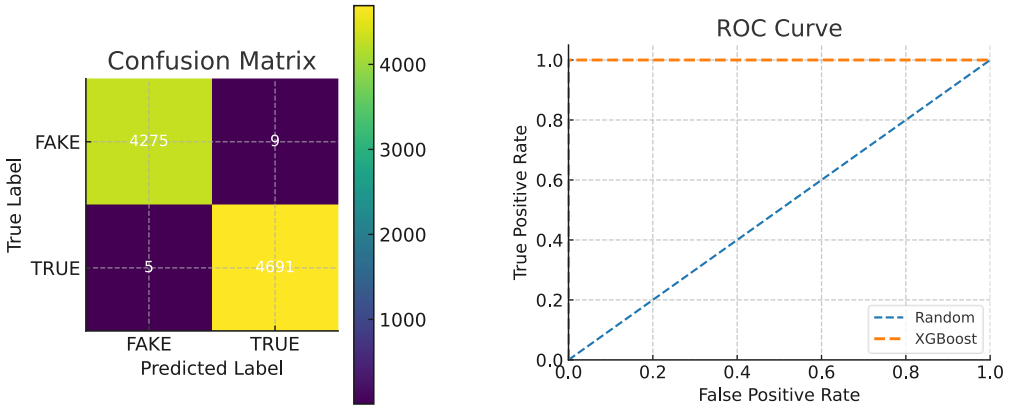
Mirroring the machine-learning setup, we evaluate deep models using text-only input with the same CountVectorizer representation; Table 2 reports each model's best configuration. The MLP achieves the highest accuracy (99.64%), with the TensorFlow baseline close behind (99.62%). The PyTorch baseline lags at 98.88%, driven primarily by lower recall, indicating a higher rate of missed positives relative to the other deep models.

Table 2: Comparative performance of deep-learning models

Model	**Vectorizer**	**Accuracy**	**Precision**	**Recall**	**F1-score**
MLP	CountVectorizer	0.9964	0.9965	0.9965	0.9965
TensorFlow	CountVectorizer	0.9962	0.9961	0.9965	0.9963
PyTorch	CountVectorizer	0.9888	0.9939	0.9846	0.9892

4.3. Evaluation

Among all models evaluated, XGBoost achieved the best overall performance with an accuracy of 99.85%. XGBoost is a powerful boosted machine-learning algorithm based on gradient-boosted decision trees, which sequentially combine weak learners to minimize residual errors and enhance predictive performance. Despite this impressive result, it warrants cautious interpretation. Exploratory analyses show that tokens such as "Reuters," "AP," and "Associated Press" appear almost exclusively in *real* articles, while terms like "Obama," "Trump," and "virus" are more prevalent in the *fake* class (see Figure 3). These distributions suggest that the model may partially rely on publisher or topic proxies rather than purely veracity-related cues. This observation does not invalidate the head-to-head benchmark; however, it implies that absolute accuracy should not be conflated with generalizable fake news detection capability. To complement point estimates, Figure 6 reports the confusion matrix and ROC curve for the best XGBoost configuration, providing a fuller view of error patterns and threshold behavior.



(a) Confusion Matrix (b) ROC Curve

Figure 6: Confusion matrix (a) and ROC curve (b) for the best XGBoost model.

Table 3. Computational efficiency comparison of top-performing models

Model	**Training Time (seconds)**	**Inference Latency (ms/sample)**	**Accuracy (%)**
XGBoost	42	0.12	**99.85**
MLP	210	0.35	99.64
Logistic Regression	8	0.05	99.65
Random Forest	95	0.28	99.29
SVM	185	0.41	99.57

The efficiency metrics in Table 3 highlight a compelling advantage of XGBoost beyond raw accuracy: it achieves the highest performance while maintaining a favorable balance between training speed, inference latency, and model footprint. Although Logistic Regression trains faster and yields a smaller model, its accuracy is marginally lower. In contrast, deep learning models (e.g., MLP) and ensemble methods like Random Forest require significantly more time and memory, with diminishing returns in accuracy. These characteristics make XGBoost particularly well-suited for real-time or resource-constrained deployment scenarios—such as browser extensions, mobile apps, or high-throughput moderation systems—where low latency and modest memory usage are critical.

4.4. Threats to Validity

1. Dataset bias: The corpus exhibits topic and source confounding: items labeled *real* largely originate from a single reputable outlet (e.g., Reuters), whereas *fake* items predominantly come from politically oriented hoax sites. This shifts the task toward domain recognition rather than genuine veracity detection, potentially inflating

headline accuracy. Mitigations. Incorporate *multi-source* real articles, balance topical coverage, and report cross-dataset transfer to assess portability.

2. Limitation: our evaluation is confined to a single corpus, which limits the external validity of our findings. We do not perform cross-dataset validation; therefore, the reported gains should be interpreted as evidence of within-dataset performance rather than robust generalization across topics, sources, or time.

3. Metric limitations and base-rate shift: Accuracy can be misleading in imbalanced or high-stakes deployments. Although the study corpus is approximately balanced, real-world fake news is comparatively rare, and operational contexts impose asymmetric costs (false positives vs. false negatives). Mitigations. Report precision–recall curves, AUCPR, class-conditional error analyses, and cost-sensitive summaries (or calibrated decision thresholds) to reflect performance under base-rate shift.

4. Model scope: The deep-learning baselines exclude pre-trained transformers (e.g., BERT-family models) that leverage transfer learning and operate under a different capacity/data regime. Consequently, conclusions primarily pertain to content-only models trained from scratch on sparse n-gram features. Mitigations. Extend comparisons to modern transformer encoders and include cross-dataset validation in future work.

4.1. Overall Discussion: Re-evaluating the Need for Complexity in Fake News Detection

Our results show that XGBoost, as a representative modern ensemble method, offers an attractive balance of accuracy, stability, and computational efficiency. It consistently matches or outperforms deeper architectures in our text-only setting, while remaining relatively easy to train, tune, and deploy.

1. The Strength of Modern Ensemble Methods: The success of XGBoost underscores the potency of gradient boosting for high-dimensional, tabular-like data generated by text vectorization. Unlike deep learning models which require extensive hyperparameter tuning and computational resources, XGBoost efficiently builds a strong model by sequentially correcting errors, making it exceptionally well-suited for this task. This finding aligns with and strongly reinforces the observations of [Shu et al., 2017] on the competitiveness of traditional models.

2. The Role of Feature Engineering: The fact that a simple CountVectorizer (Bag-of-Words (BoW)) coupled with rigorous pre-processing (stemming, stop-word removal) yielded superior results over more sophisticated embeddings (in this context) is profound. It suggests that for the specific task of discriminating between the style and lexicon of fake and real news in this dataset, word presence and frequency are highly discriminative features. This implies that fake news articles may use a consistently different vocabulary or stylistic register, which does not necessarily require deep semantic understanding to identify.

3. The Marginal Gains of Deep Learning: While the MLP and TensorFlow models achieved excellent accuracy (>99.6%), they were marginally outperformed by

XGBoost. More importantly, they required significantly more training time and computational power. This result challenges the automatic assumption that deeper networks are always better and highlights that for many text classification problems, the performance ceiling may be reachable with more efficient methods. The law of diminishing returns strongly applies here. While recent works propose increasingly complex transformer-based frameworks that focus on explainability and controllability (Liu et al., 2024), our results show that under strong lexical/semantic feature engineering, tree-based models can remain competitive on tabular text representations.

4. Interpretability and Deployment Advantages: The superior performance of models like Logistic Regression and XGBoost (which offers feature importance scores) provides a level of interpretability that deep learning models often lack. In a sensitive domain like fake news detection, the ability to understand why an article was classified a certain way is crucial for trust and adoption in real-world systems.

5. Conclusion and Future Work

This study offered a controlled, like-for-like comparison between eight classical learners and three deep baselines for content-only fake news detection. Under a unified pipeline, XGBoost combined with simple n-gram (bag-of-words/TF-IDF) features achieved the strongest performance while training quickly and remaining straightforward to deploy and audit. These results reinforce a practical message for researchers and practitioners: when inputs are sparse and tabular-like, a carefully tuned gradient-boosted tree can rival—and sometimes surpass—custom deep architectures, delivering a favorable balance of accuracy, efficiency, and interpretability.

It is important to clarify the methodological scope of this study: our comparative analysis is intentionally centered on traditional text transformation techniques—namely Count, TF-IDF, and Hashing vectorizers—which convert raw text into structured, sparse feature representations suitable for classical machine learning and from-scratch deep learning models. Within this paradigm, model performance depends critically on these explicit feature engineering steps. In contrast, modern large language models (LLMs) such as BERT, RoBERTa, or DistilBERT operate directly on raw or minimally processed text, leveraging pre-trained contextual embeddings that eliminate the need for intermediate vectorization. Because this represents a fundamentally different modeling approach—one based on transfer learning rather than engineered n-gram features—we have excluded transformer-based architectures from the current experimental comparison. This deliberate scope allows us to establish strong, interpretable, and efficient baselines under controlled conditions. The integration and evaluation of LLMs, using the same bias-aware and cross-dataset validation framework developed here, constitute a key direction for our immediate future work.

While our best model achieves near-ceiling accuracy on the Kaggle “Fake and Real News” corpus, these results should not be interpreted as proof of broad, real-world generalizability. The study is intentionally limited to a single benchmark and to content-based features only. As previous work has shown, fake-news datasets often contain dataset-specific artifacts—such as topic or source bias—that can inflate in-dataset performance. A natural extension of this work is to apply the same pipeline across

multiple corpora (e.g., FakeNewsNet, LIAR, CoAID) under a unified preprocessing and evaluation protocol to systematically assess cross-dataset robustness.

Looking ahead, we see several concrete priorities that build on the evidence gathered here:

1. Recent studies have demonstrated that transformer-based models, when fine-tuned on domain-specific text classification tasks, can achieve strong performance across diverse applications. Transformer comparison under identical controls: Evaluate fine-tuned transformer encoders (e.g., BERT/RoBERTa) within the same preprocessing, splitting, and reporting protocol used for XGBoost. Compare not only accuracy but also calibration, inference latency, memory footprint, and statistical significance of head-to-head differences.
2. Robustness to adversarial and generative shifts: Stress-test models against realistic perturbations (lexical and semantic) and synthetic articles produced by large language models. Report degradation curves and investigate lightweight hardening strategies without sacrificing efficiency.
3. Cross-source, cross-time, and cross-lingual validation: Move beyond a single benchmark by testing on datasets that vary in outlets, topics, time windows, and languages. Include source-aware and time-based splits to mimic deployment drift, and document when performance drops—and why.
4. Multimodal and context integration: Augment text with social/contextual signals (e.g., user credibility, propagation patterns) and, where available, visual cues. Quantify the incremental value of each modality over a strong text-only baseline to justify added system complexity.
5. Bias-aware modeling and transparent auditing: Adopt routine bias diagnostics (per-topic performance, source-masked attributions, confusion breakdowns) and release concise transparency notes alongside models and code. Where decisions are high-stakes, keep a human in the loop and log post-deployment feedback for iterative improvement.
6. Operationalization and monitoring: Package the best model as a lightweight service with deterministic preprocessing and vectorization, and instrument it with telemetry for latency, throughput, input drift, and error review. Document energy/compute costs to support sustainable deployment choices.

In short, our contribution is less about chasing a single top-line number and more about clarifying when simple, transparent methods are enough—and what it takes to make results travel across datasets and over time. We hope this serves as a reliable baseline and a practical blueprint for more robust, bias-aware, and field-ready fake news detection.

Data Availability Statement:

The dataset used in this study is publicly available at:

https://www.kaggle.com/clmentbisaillon/fake-and-real-news-dataset

Code Availability:

The source code for preprocessing, training, and evaluation is available at:

https://github.com/boulbaba1981/FakeNewsDetector

References:

Ahmed, H., Traore, I., & Saad, S. (2017). Detection of online fake news using n-gram analysis and machine learning techniques. *Intelligent, Secure, and Dependable Systems in Distributed and Cloud Environments.*

Allcott, H., & Gentzkow, M. (2017). Social media and fake news in the 2016 election. *Journal of Economic Perspectives, 31*(2), 211–236.

Alshuwaier, F. A., & Alsulaiman, F. A. (2025). *Fake news detection using machine learning and deep learning algorithms: A comprehensive review and future perspectives.* Computers, 14(9), 394.

Baly, R., Karadzhov, G., Alexandrov, D., Glass, J., & Nakov, P. (2018). Predicting factuality of reporting and bias of news media sources. *Proceedings of NAACL-HLT 2018*, 2529–2539.

Cui, L., & Lee, D. (2020). CoAID: COVID-19 healthcare misinformation dataset. *arXiv preprint*, arXiv:2006.00885.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. *Proceedings of NAACL-HLT 2019*, 4171–4186.

Goldstein, J. A., Sastry, G., Musser, M., DiResta, R., Gentzel, M., & Sedova, K. (2023). Generative language models and automated influence operations: Emerging threats and potential mitigations. *arXiv preprint*, arXiv:2301.04246.

Grinsztajn, L., Oyallon, E., & Varoquaux, G. (2022). Why do tree-based models still outperform deep learning on typical tabular data? *NeurIPS Datasets and Benchmarks Track.*

Gruppi, M., Horne, B. D., & Adalı, S. (2022). When topic bias masquerades as fake news detection. *Proceedings of the International AAAI Conference on Web and Social Media (ICWSM), 16*(1), 387–398.

Harris, S., Hadi, H. J., Ahmad, N., & Alshara, M. A. (2024). *Fake news detection revisited: An extensive review of theoretical frameworks, dataset assessments, model constraints, and forward-looking research agendas.* Technologies, 12(11), 222.

Horne, B., & Adalı, S. (2017). This just in: Fake news packs a lot in. *ICWSM Workshops.*

Kaliyar, R. K., Goswami, A., & Narang, P. (2021). FakeBERT: Fake news detection in social media with a BERT-based deep learning approach. *Multimedia Tools and Applications, 80*(8), 11765–11788.

Kwon, S., Cha, M., & Jung, K. (2017). Rumor detection over varying time windows. *PLOS ONE, 12*(1), e0168344.

Liu, H., Wang, W., Li, H., & Li, H. (2024). *TELLER: A trustworthy framework for explainable, generalizable and controllable fake news detection.* arXiv:2402.07776.

Lv, Y. et al. (2025). *Multi-modal fake news detection: A comprehensive survey on deep learning technology, advances, and challenges.* Journal of King Saud University – Computer and Information Sciences.

Pérez-Rosas, V., Kleinberg, B., Lefevre, A., & Mihalcea, R. (2018). Automatic detection of fake news. *Proceedings of COLING 2018.*

Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter. *arXiv preprint*, arXiv:1910.01108.

Shu, K., Sliva, A., Wang, S., Tang, J., & Liu, H. (2017). Fake news detection on social media: A data mining perspective. *SIGKDD Explorations, 19*(1), 22–36.

Shu, K., Mahudeswaran, D., Wang, S., Lee, D., & Liu, H. (2020). FakeNewsNet: A data repository with news content, social context and spatiotemporal information. *Big Data.*

Singh, D., Khan, A. H., & Meena, S. (2023). Fake News Detection Using Ensemble Learning Models. In *Proceedings of the Data Analytics and Management (ICDAM 2023)* (Vol. 78, pp. 55–63). Springer.

Thorne, J., Vlachos, A., Christodoulopoulos, C., & Mittal, A. (2018). FEVER: A large-scale dataset for fact extraction and verification. *Proceedings of NAACL-HLT 2018.*

Vosoughi, S., Roy, D., & Aral, S. (2018). The spread of true and false news online. *Science, 359*(6380), 1146–1151.

Wang, W. Y. (2017). "Liar, Liar Pants on Fire": A new benchmark dataset for fake news detection. *Proceedings of ACL 2017.*

Zellers, R., Holtzman, A., Rashkin, H., Bisk, Y., Farhadi, A., Roesner, F., & Choi, Y. (2019). Defending against neural fake news. *Advances in Neural Information Processing Systems (NeurIPS), 32*, 9051–9062.

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

Zubiaga, A., Aker, A., Bontcheva, K., Liakata, M., & Procter, R. (2018). Detection and resolution of rumours in social media: A survey. *ACM Computing Surveys.*