JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

Fake News Detection Using a Voting Ensemble of Classical Learners with Optimized TF–IDF Feature Engineering

Saiyam Sachdev
3rd Year
Dept. of AI & Data Science
ADGIPS
GGSIPU, Delhi, India

Arnav Mudgal
3rd Year
Dept. of AI & Data Science
ADGIPS
GGSIPU, Delhi, India

Esha
3rd Year
Dept. of AI & Data Science
ADGIPS
GGSIPU, Delhi, India

Ritesh Chandel
Assistant Professor
ADGIPS
GGSIPU, Delhi, India

***Abstract*—The rapid growth of online news portals and social platforms has made it easier than ever to spread fabricated information, with serious consequences for public debate, political stability and crisis response. Because manual fact-checking cannot keep up with the volume of content published every day, efficient automated detectors are needed. This paper proposes an ensemble framework for binary fake news classification in which Logistic Regression, Random Forest and Extreme Gradient Boosting (XGBoost) classifiers are combined through hard majority voting. We merge two complementary corpora, LIAR (short political statements) and ISOT (full-length news articles), into a single binary-labelled dataset, clean the text, and encode it with a tuned Term Frequency–Inverse Document Frequency (TF–IDF) representation whose size is capped to limit overfitting. On the held-out test split, the three base models reach accuracies of 89.67%, 90.91% and 90.97%, respectively, while the ensemble reaches 91.22% with balanced precision and recall across the two classes (macro F1 ≈ 0.91). A per-class analysis shows a moderate drop in recall on the fake class, which deserves attention before deployment in adversarial settings. The results suggest that a well-engineered ensemble of classical learners can be competitive in accuracy at a much lower computational cost than transformer-based models, making it a good fit for resource-constrained or latency-sensitive environments.**

***Keywords*—Fake news detection; ensemble learning; majority voting; Logistic Regression; Random Forest; XGBoost; TF–IDF; text classification; natural language processing; misinformation.**

I. INTRODUCTION

Today's information landscape is defined by an enormous and constantly accelerating flow of user-generated content. Social networks, micro-blogging services and online news aggregators have opened publishing to anyone, but they have also made it cheap to create and circulate deliberately false, misleading or weaponized material, usually called fake news. Studies of recent elections, public-health emergencies and geopolitical conflicts indicate that such material can travel faster and farther than verified reporting [1], [2]. The problem is not limited to obscure websites; large platforms have repeatedly acted as channels for coordinated disinformation, computational propaganda and manipulation through synthetic media.

The harm extends well beyond the informational. Fake news has been associated with declining trust in democratic institutions, manipulated voter sentiment, the spread of health misinformation, disturbances in financial markets and outbreaks of social unrest. Editorial fact-checking and platform moderation remain indispensable, yet they are reactive by nature: human reviewers cannot match the amount of content posted every second worldwide. Even well-funded fact-checking organizations handle only a small share of viral claims, and the time needed to verify a claim usually exceeds the period in which it attracts most of its audience. Scalable, accurate and automated detection is therefore a research priority rather than a luxury.

Machine learning is now the dominant approach to this problem. Early studies fed hand-crafted lexical and syntactic features to classical classifiers [1], [3], while more recent work relies on deep architectures such as recurrent, convolutional and transformer-based networks [10]–[12]. Despite their expressive power, deep models face three recurring practical obstacles. Their accuracy depends strongly on how the dataset is composed and often falls under distribution shift, with drops of 10–20 percentage points commonly reported when a model trained on one corpus is tested on another. They also demand large labelled datasets and heavy computation for both training and inference, which small newsrooms, fact-checking startups and edge devices can rarely afford. Finally, their opacity makes them hard to use in editorial workflows, where journalists need interpretable signals and where regulation increasingly calls for explainability.

For these reasons, classical learners combined with carefully designed features remain attractive, especially when they are ensembled. Ensembles exploit the complementary bias and variance profiles of different model families and have often outperformed their members on text-classification benchmarks. The underlying idea is that each family errs differently: linear models overlook non-linear interactions, deep tree ensembles can latch onto spurious correlations, and gradient boosters can be thrown off by noisy labels. Combining their outputs by majority vote can therefore correct some errors without a proportional rise in computational cost.

Guided by this idea, we ask whether a compact ensemble of Logistic Regression, Random Forest and XGBoost, trained on an optimized TF–IDF representation, can approach the accuracy reported for deep models while remaining cheap to train and easy to interpret.

This paper makes the following contributions. (1) We build a unified binary corpus by harmonizing LIAR [3] and ISOT [14], so that the models see both short claims and full articles, which improves robustness across text lengths. (2)

We design a tuned TF–IDF pipeline that uses a capped vocabulary, document-frequency thresholds and bigrams to balance discriminative power against overfitting. (3) We build a hard-voting ensemble of three diverse classifiers and compare it with each member under the same preprocessing and evaluation protocol. (4) We analyse errors per class through precision, recall, F1-score and the confusion matrix, exposing a systematic asymmetry in the remaining mistakes. (5) We report computational cost alongside accuracy to help practitioners decide on deployment in resource-limited settings.

The rest of the paper is organized as follows. Section II reviews related work and states the gap this study addresses. Section III presents the methodology, covering data preparation, feature engineering and ensemble construction. Section IV reports and discusses the experimental results, Section V examines computational cost and deployment, Section VI discusses limitations and threats to validity, and Section VII concludes with directions for future work.

II. RELATED WORK

Automated fake news detection has developed along three main lines: classical learning on engineered features, neural sequence models, and transformer-based contextual models. A separate body of work uses social-network and propagation signals, but it lies outside this purely text-based study. We summarize representative work in each line, focusing on reported accuracy, generalization and deployment cost.

A. Classical Learning with Lexical Features

Shu et al. [1] gave one of the first broad data-mining treatments of fake news, formalizing the task and describing feature families built on content, social context and propagation. Their taxonomy still shapes modern pipelines and stresses the benefit of combining stylistic, semantic and network cues. Rubin et al. [2] showed that satirical cues such as absurdity, humour and exaggeration can serve as discriminative signals, which encouraged later stylometric work. Wang [3] introduced the LIAR dataset and found that, even with rich metadata, short political statements are hard to classify, with the best model improving only slightly over a majority-class baseline. Later studies have confirmed this, indicating that detecting fake news in short text is intrinsically harder than in long-form articles.

Subsequent feature-based studies using TF–IDF, n-gram and stylometric features generally report accuracies between 85% and 90% on curated corpora [6], [7]. Choras et al. [7] systematically mapped this literature and concluded that careful preprocessing and feature selection often matter more than the sophistication of the algorithm. This conclusion matches the engineering philosophy adopted here, which puts effort into the representation rather than the model architecture.

B. Hybrid and Deep Sequential Models

Ruchansky et al. [4] proposed the CSI hybrid model, which jointly models article content, user responses and source information and improves on content-only baselines; it was an early use of recurrent encoders for temporal engagement signals. Thorne and Vlachos [5] released the FEVER benchmark to promote retrieval-augmented verification, moving attention from black-box classification toward grounded fact-checking. Retrieval-based systems indicate the long-term direction of the field, but their reliance on curated knowledge bases and their high inference cost restrict near-term deployment.

Recent neural studies give mixed results. Beseiso et al. [12] combined LSTM and CNN encoders with contextual signals and obtained competitive accuracy on several benchmarks, yet saw marked degradation when the test distribution differed from the training data. Roumeliotis et al. [13] reported CNN accuracy as low as 58.6% on adversarial subsets, showing how brittle single-architecture pipelines can be against deliberately crafted deceptive content. The lesson is that neural models that do well in-distribution do not necessarily handle the variety of real-world misinformation.

C. Transformer-Based and Attention Models

Attention-based and transformer architectures have also been applied to misinformation detection. BERT-family models typically reach 83% to 89% accuracy on cross-domain benchmarks [10], [11], depending heavily on the fine-tuning corpus and label quality. Alshuwaier [11] reported roughly 76% accuracy when sentiment-aware features were combined with classical learners, showing that depth alone does not guarantee better generalization and that sentiment polarity is a noisy cue, since legitimate news also often carries strong sentiment.

Domain-adapted transformers, such as fact-aware variants of BERT and RoBERTa, achieve higher accuracy but at considerable computational cost. Their inference latency can exceed tens of milliseconds per article on a GPU and several hundred milliseconds on a CPU, which is prohibitive for high-throughput moderation pipelines. Their hundreds of millions of parameters also raise energy and carbon-footprint concerns that are beginning to attract policy attention.

D. Identified Research Gap

Two observations follow from the literature. First, accuracy above 90% appears achievable only under restricted dataset conditions, and most pipelines degrade significantly under domain shift. Second, deep models carry a computational overhead that is not always justified by their gain over well-engineered classical baselines. Table I gathers the reported results for comparison; because the figures come from different datasets and protocols, the comparison is indicative rather than direct, but it suggests that a carefully engineered ensemble can sit at a favourable point on the accuracy–cost frontier.

TABLE I
REPORTED ACCURACIES OF RECENT FAKE NEWS DETECTION APPROACHES

Reference	Approach	Reported Acc.
Alshuwaier [11]	ML + sentiment features	~76%
Roumeliotis [13]	CNN (adversarial subset)	58.6%
BERT family [10]	Transformer fine-tune	83–89%
Beseiso [12]	LSTM + CNN context	~88%
This work	**LR + RF + XGB ensemble**	**91.22%**

These findings motivate a middle path: combine the complementary inductive biases of several classical learners through ensembling, and invest engineering effort in the feature representation instead of architectural depth. Our framework joins a linear model (Logistic Regression), a bagged tree ensemble (Random Forest) and a boosted tree ensemble (XGBoost). These cover three distinct hypothesis classes whose errors should be partly decorrelated, which is the precondition for ensemble gains.

III. PROPOSED METHODOLOGY

The framework has five sequential stages: (i) data acquisition and harmonization, (ii) text preprocessing, (iii) optimized TF–IDF feature extraction, (iv) training of the base classifiers, and (v) ensemble construction through hard majority voting. Fig. 1 shows the pipeline, and each stage is described below.

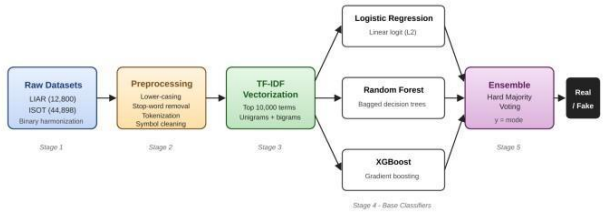

Fig. 1. Pipeline of the proposed ensemble framework, showing the five stages from raw datasets to the final binary decision.

A. Datasets and Harmonization

We use two public corpora. LIAR [3] holds about 12,800 short, manually labelled political statements taken from PolitiFact, with metadata such as speaker, context and topic. Its statements carry a six-point veracity label: true, mostly true, half-true, barely true, false and pants-on-fire. The ISOT Fake News Dataset [14] provides 44,898 full-length articles, split evenly between verified outlets such as Reuters and unreliable sources flagged by independent fact-checkers.

To obtain a single label space, LIAR's six classes are mapped to two: true and mostly true become real, whereas barely true, false and pants-on-fire become fake. Half-true statements are intentionally dropped to reduce label noise and to avoid forcing a choice on truly ambiguous claims. ISOT articles are already binary and are added unchanged. The resulting corpus has 57,698 instances in two distributionally different length regimes, short claims and long articles, which should improve robustness across them. Fig. 2 shows the composition of the corpus and its class balance.

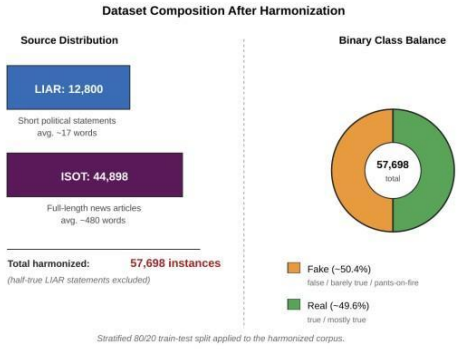

Fig. 2. Dataset composition after harmonization: contribution of LIAR and ISOT and the resulting near-balanced class distribution.

B. Text Preprocessing

Raw text passes through a deterministic cleaning pipeline with five steps: (i) removal of duplicate and null records, (ii) lower-casing, (iii) regular-expression removal of punctuation, digits, URLs and non-ASCII symbols, (iv) removal of English stop-words with the standard NLTK list, and (v) word-level tokenization by whitespace splitting. Lemmatization is deliberately skipped because preliminary experiments showed that it erases morphological cues, such as tense and plural forms, that can carry rhetorical weight in the political claims of the LIAR portion. Keeping the pipeline simple and deterministic supports reproducibility and avoids dataset-specific normalization that could harm generalization.

Outliers are treated conservatively. Documents with fewer than three tokens after cleaning are dropped; these are mostly truncated LIAR statements with degenerate content. Very long ISOT articles are not truncated, since L2 normalization makes the TF–IDF representation length-invariant.

C. Optimized Feature Engineering

Each cleaned document is converted to a sparse vector with a TF–IDF transform. Term frequency, inverse document frequency and the combined weight are defined as

$$\mathrm{TF}(t, d) = f(t, d) / \Sigma_k f(k, d) \quad (1)$$

$$\mathrm{IDF}(t) = \log(N / \mathrm{df}(t)) \quad (2)$$

$$\text{TF-IDF}(t, d) = \mathrm{TF}(t, d) \times \mathrm{IDF}(t) \quad (3)$$

where $f(t, d)$ is the raw count of term t in document d, N is the number of documents, and $\mathrm{df}(t)$ is the number of documents containing t. To keep dimensionality under control, the vocabulary is limited to the 10,000 most frequent terms, with min_df = 5 and max_df = 0.85. The lower threshold removes very rare terms, often typos or names tied to a single article, which would enlarge the feature space without adding signal. The upper threshold removes terms present in more than 85% of documents, which behave like stop-words even though they escaped the explicit list. Unigrams and bigrams are both kept, so local context (e.g., "fake news", "breaking story") is captured without a combinatorial explosion.

Afterwards every vector is L2-normalized so that document length does not bias linear classifiers. The resulting sparse matrix has a non-zero density of about 0.3%, which suits the sparse implementations of all three classifiers in scikit-learn.

D. Base Classifiers

1) Logistic Regression (LR): LR estimates the probability of the positive (fake) class with a linear logit model:

$$P(y = 1 \mid x) = 1 / (1 + \exp(-(w^{T}x + b))) \quad (4)$$

where w is the weight vector and b the bias. L2 regularization with C = 1.0 limits coefficient inflation in the high-dimensional TF–IDF space. The liblinear solver is used because it handles large sparse matrices well and converges quickly. LR also gives an interpretable baseline, since the sign and magnitude of each weight show how the corresponding term pushes the decision toward the fake class.

2) Random Forest (RF): RF is a bagging ensemble of decision trees. For B trees with predictions $\{\hat{y}_1, \hat{y}_2, \ldots, \hat{y}^{B}\}$, the final label is the majority vote:

$$\hat{y}_RF = \mathrm{mode}(\hat{y}_1, \hat{y}_2, \ldots, \hat{y}^{B}) \quad (5)$$

Each tree is grown on a bootstrap sample of the training set and considers a random subset of $\sqrt{p}$ features at every split, which decorrelates the trees and lowers variance. We use 200 trees of unlimited depth with the Gini criterion. RF suits TF–IDF features because it can capture non-linear interactions between term occurrences, for example phrases that each carry weak signal but together indicate fabrication.

3) Extreme Gradient Boosting (XGBoost): XGBoost builds an additive ensemble of regression trees by minimizing a regularized objective stage by stage:

$$\mathscr{L}(\phi) = \Sigma_i \ell(y_i, \hat{y}_i) + \Sigma_k \Omega(f_k) \quad (6)$$

$$\Omega(f) = \gamma T + \tfrac{1}{2}\lambda \|w\|^2 \quad (7)$$

where ℓ is the differentiable per-instance loss (binary logistic here), T is the number of leaves of tree f, w is its

leaf-weight vector, and γ and λ control complexity. Shrinkage (learning rate 0.1), column subsampling (0.8 per tree) and second-order gradient information make XGBoost both expressive and resistant to overfitting. We run 300 boosting rounds with a maximum depth of 6, values chosen by preliminary cross-validation on a held-out development partition. Table II lists the hyperparameters of all classifiers and of the feature pipeline.

TABLE II
HYPERPARAMETER SETTINGS FOR EACH CLASSIFIER AND THE FEATURE PIPELINE

Model	Hyperparameter	Value
Logistic Regression	Solver	liblinear
	Penalty / C	L2 / 1.0
Random Forest	n_estimators	200
	max_depth	None (full)
	max_features	sqrt
XGBoost	n_estimators	300
	learning_rate	0.1
	max_depth	6
	subsample	0.8
TF-IDF	max_features	10,000
	ngram_range	(1, 2)
	min_df / max_df	5 / 0.85

E. Ensemble Construction

The three base classifiers are trained independently on the same TF–IDF matrix. At inference their outputs are combined by hard majority voting:

$$\hat{y}_{ens} = \text{mode}(\hat{y}_{LR}, \hat{y}_{RF}, \hat{y}_{XGB}) \quad (8)$$

We prefer hard voting over soft voting for two reasons. First, the models output probabilities on different scales (sigmoid for LR, vote fractions for RF, logistic link for XGBoost), and averaging uncalibrated probabilities tends to favour the most confident classifier, which is not necessarily the most accurate. Second, hard voting is robust to extreme probability mass: one overconfident classifier cannot decide the outcome against both of its peers, which is exactly the error-correcting behaviour we want. Calibration with Platt scaling or isotonic regression could in principle allow soft voting, but pilot experiments showed no meaningful accuracy gain, so we keep the simpler scheme.

With three classifiers and two labels, the ensemble decision is unanimous when all models agree and a majority decision when one dissents. This gives a useful diagnostic: instances on which the models disagree are inherently ambiguous and are natural candidates for human review in a hybrid moderation pipeline.

F. Implementation and Evaluation Protocol

All experiments were implemented in Python 3.10 with scikit-learn 1.3 for LR and RF [6], the official xgboost 1.7 package, and pandas/NumPy for data handling. They ran in a Jupyter Notebook on a workstation with an Intel Core i7 CPU and 16 GB RAM, with no GPU. All random seeds are fixed at 42 for reproducibility.

The harmonized corpus is divided into 80% training and 20% test partitions by stratified sampling to preserve class balance, leaving 11,538 test instances that are used only for the final evaluation in Section IV. We report accuracy, precision, recall and F1-score per class and as macro and weighted averages, together with the confusion matrix to expose class-wise error patterns. Macro-averaging is the primary aggregate because it weights both classes equally and so reveals class asymmetries better than weighted averaging, which the slightly larger fake class would dominate.

IV. EXPERIMENTAL RESULTS AND DISCUSSION

A. Quantitative Performance Comparison

Table III compares the three base learners and the proposed ensemble on the held-out test set, and Fig. 3 shows the same data as a grouped bar chart so that the benefit of voting is easy to see.

TABLE III
PERFORMANCE COMPARISON OF INDIVIDUAL MODELS AND THE PROPOSED ENSEMBLE

Model	Accuracy	F1 (Fake)	F1 (Real)
Logistic Regression	0.8967	0.8949	0.8984
Random Forest	0.9091	0.9086	0.9096
XGBoost	0.9097	0.9085	0.9108
Proposed Ensemble	**0.9122**	**0.9104**	**0.9139**

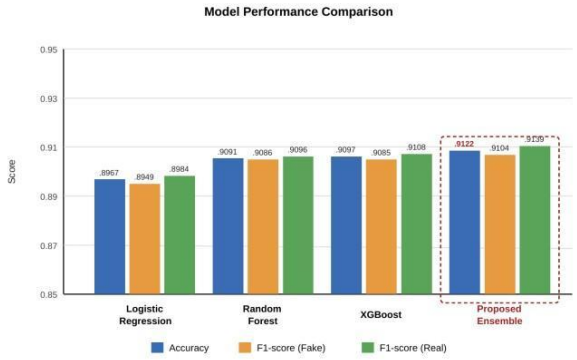

Fig. 3. Grouped bar chart of accuracy and per-class F1-score for the three base learners and the proposed ensemble (highlighted), which scores highest on every metric.

Logistic Regression forms a competitive linear baseline at 89.67% accuracy. Random Forest and XGBoost both improve on it, which suggests that non-linear interactions among TF–IDF features hold discriminative information that a linear boundary cannot fully use. The gaps of 1.24 points (LR vs. RF) and 1.30 points (LR vs. XGBoost) are consistent with the non-linearity premium usually seen on TF–IDF text-classification tasks. RF and XGBoost themselves perform almost identically (90.91% vs. 90.97%); the tiny difference lies within the run-to-run variance caused by seed choice.

The ensemble attains the best accuracy, 91.22%, which is 0.25 points above the strongest single model (XGBoost) and 1.55 points above the weakest (LR). The margin over XGBoost is small, but the ensemble also obtains the best F1-score on both classes, suggesting that its gain comes from correcting asymmetric errors rather than from uniform improvement. More precisely, voting helps most when one model is wrong and the other two are right; it cannot fix cases in which two or more members agree on a wrong label, which is the fundamental ceiling of hard voting.

B. Class-wise Behaviour and Confusion Matrix

Fig. 4 shows the ensemble's confusion matrix on the test set, and Fig. 5 breaks down precision, recall and F1-score for each class.

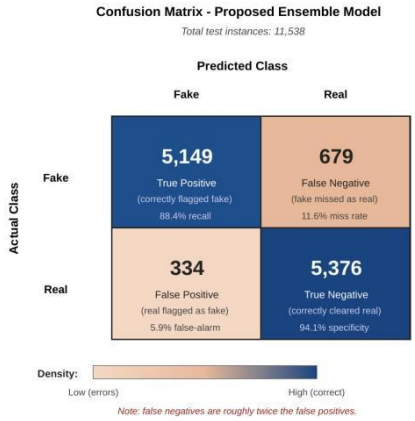

Fig. 4. Confusion matrix of the proposed ensemble on the test partition (11,538 instances). Diagonal cells are correct predictions; the off-diagonal cells show a moderate asymmetry in error types.

Of the 11,538 test instances, 5,149 fake and 5,376 real examples are classified correctly, giving a true-positive rate (sensitivity) of 0.884 for the fake class and a true-negative rate (specificity) of 0.941 for the real class. The errors are slightly but systematically asymmetric: false negatives (679) are about twice as many as false positives (334).

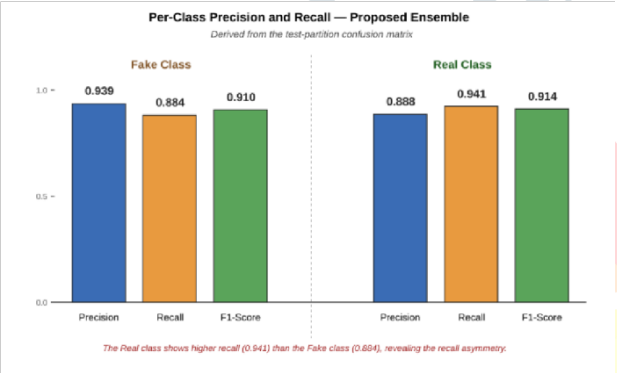

Fig. 5. Per-class precision, recall and F1-score from the test confusion matrix. The real class has higher recall, whereas the fake class has higher precision.

Fig. 5 shows two things. First, precision is higher for the fake class (0.939) than for the real class (0.888): when the ensemble labels an item fake, it is right about 94 times in 100, which is valuable for moderation, where wrongly removing legitimate content is reputationally costly. Second, recall is lower for the fake class (0.884) than for the real class (0.941), so roughly 11.6% of fabricated content goes undetected. In fake news detection this matters, since each false negative is fabricated content that stays in circulation.

We attribute the asymmetry to the greater length and more polished register of the fake articles in ISOT, which resemble legitimate journalism in their TF–IDF profile. Many fabricated articles imitate the lexical and structural conventions of mainstream reporting, such as datelines, attribution phrases and professional headlines, so an n-gram representation cannot separate them on style alone. This is a structural limitation of bag-of-words features that contextual encoders are better placed to overcome. Reducing the asymmetry is a natural goal for future work, through cost-sensitive training that penalizes false negatives more heavily, decision-threshold tuning that trades some precision for recall, or stylometric features such as readability indices, punctuation density and sentence-length statistics.

C. Discussion of Findings

The results support three conclusions. First, classical learners with a well-engineered TF–IDF representation stay competitive on heterogeneous fake-news corpora and can match the accuracy typically reported for transformer baselines [10]–[12] at a fraction of the computational cost. The 91.22% obtained here is at or above the usual 83–89% range reported for fine-tuned BERT-family models on cross-domain benchmarks, while needing far less training time and no GPU. Because those figures come from different datasets and protocols, this is an indicative comparison rather than a controlled head-to-head.

Second, ensembling heterogeneous base learners gives consistent but modest gains, mostly through error correction rather than uniform improvement. The benefit of ensembling shrinks as the base accuracies converge, and our results are close to that regime: the best single model (XGBoost, 90.97%) trails the ensemble (91.22%) by only 0.25 points. Larger or more diverse ensembles might gain more, at growing computational cost.

Third, the remaining errors reveal a recall deficit on the fake class that aggregate accuracy hides, which underlines the need for class-wise reporting in misinformation research. Practitioners should track class-wise metrics and not just headline accuracy, particularly when missed detections and false alarms carry different costs, as is almost always the case in moderation pipelines.

V. COMPUTATIONAL COST AND DEPLOYMENT

Practical deployment depends on training and inference cost, memory footprint and update frequency as well as on accuracy. This section reports measurements from our experiments and discusses what they mean in practice.

TABLE IV
COMPUTATIONAL COST OF EACH MODEL ON THE HARMONIZED CORPUS

Model	Training Time (s)	Inference (ms/sample)
Logistic Regression	~12	<0.1
Random Forest	~85	~1.4
XGBoost	~110	~0.7
Ensemble (sum)	**~207**	**~2.2**

Table IV gives wall-clock training time and per-sample inference latency on the experimental workstation (Intel Core i7, 16 GB RAM, no GPU). LR is by far the cheapest to train (about 12 s) and the fastest at inference (below a millisecond), reflecting the simplicity of linear models. RF needs about 85 s because it builds 200 unpruned trees, and its inference latency grows accordingly. XGBoost has the longest training time (about 110 s) owing to its sequential boosting, but it infers faster than RF thanks to its more compact trees and SIMD-friendly implementation.

The full ensemble takes about 2.2 ms per sample at inference, two to three orders of magnitude less than CPU inference of BERT-family models on comparable hardware (typically 100–500 ms per sample). For a moderation pipeline handling 1,000 articles per second, the ensemble would need roughly 2.2 CPU cores, whereas a transformer would need substantial GPU resources or aggressive batching with added latency. This profile suits high-throughput, latency-sensitive uses such as real-time social-media moderation, news-aggregator filtering and edge-device verification assistants.

Memory use is also modest. The trained artifacts take about 80 MB on disk, mostly the Random Forest, and the TF–IDF vocabulary adds about 3 MB. Such a small footprint allows deployment on commodity virtual machines and even resource-limited edge devices, in contrast with multi-gigabyte transformer checkpoints.

VI. LIMITATIONS AND THREATS TO VALIDITY

Several limitations and threats to validity should be stated explicitly, both to put the reported results in context and to guide future work.

A. Feature Modality

The evaluation uses text features only. Source credibility, propagation networks, multimodal content (images, video, infographics) and user-engagement patterns are not exploited, although such auxiliary signals help considerably in practice, especially for borderline cases where text alone is insufficient. Adding them through late fusion or cross-attention architectures is a clear extension.

B. Linguistic Coverage

Both corpora are mainly English, so performance on multilingual or code-switched content, such as Hindi–English mixing in South Asia or Spanish–English mixing in the Americas, is untested. TF–IDF is in principle language-agnostic, but vocabulary handling, tokenization and stop-word lists would need adapting for each language. Cross-lingual fake-news detection is a research direction in its own right.

C. Label Noise and Class Mapping

Mapping LIAR's six classes to two necessarily discards information, and excluding half-true statements introduces a selection bias whose effect on test performance has not been quantified. ISOT labels are generally regarded as reliable, but they depend on the original collectors' curation, and some labelling errors are probably present, as in any large web-scraped corpus.

D. Adversarial Robustness

We did not assess robustness to adversarial perturbations such as paraphrasing, synonym substitution or LLM-generated counterfeits. As generative models become widely available, fabricated text increasingly shows a fluency that defeats lexical-feature detectors. Robustness studies against attacks and synthetic text are needed before any production deployment.

E. Temporal Drift

Fake-news topics change quickly, and a model trained on historical data may degrade on current content. We use a single random train–test split instead of a time-ordered one, which would better mimic deployment conditions. Future work should include temporal cross-validation and an analysis of retraining schedules.

VII. CONCLUSION AND FUTURE WORK

This paper presented an ensemble framework for fake news detection that combines Logistic Regression, Random Forest and XGBoost through hard majority voting on an optimized TF–IDF feature space. Harmonizing LIAR and ISOT into one binary corpus exposed the framework to both short claims and long articles, improving robustness across text lengths. On the held-out test set the ensemble reached 91.22% accuracy with macro F1 ≈ 0.91, outperforming each constituent classifier while remaining trainable on commodity hardware without a GPU.

The findings support a pragmatic message: for resource-constrained or latency-sensitive deployments, a carefully engineered ensemble of classical learners offers an attractive trade-off between accuracy and cost compared with large neural models. An inference latency of about 2.2 ms per sample on a commodity CPU makes real-time use feasible in moderation pipelines, news aggregators and edge assistants, where transformer-based alternatives would demand prohibitive resources. The class-wise analysis nevertheless revealed a recall deficit on the fake class that should be addressed before deployment in adversarial settings, for example through cost-sensitive training, threshold tuning or stylometric features.

Future work has five directions. First, contextual encoders such as BERT and RoBERTa will be evaluated as additional ensemble members, using calibrated soft voting to exploit their probability estimates while keeping the efficiency of the classical core. Second, the framework will be extended to multilingual and code-switched content, common in societies such as India, with careful treatment of tokenization and vocabulary across languages. Third, multimodal signals, including images, video frames and propagation graphs, will be added through late fusion that preserves the modular design. Fourth, retrieval-augmented verification against curated knowledge bases will be explored to move from stylistic cues toward real factual grounding, possibly together with established fact-checking organizations. Fifth, adversarial-robustness and temporal-drift studies will test readiness under realistic operating conditions.

Together these directions point toward fake-news detection systems that are accurate, interpretable, multilingual, multimodal and deployable in real time, contributing to the broader goal of protecting the integrity of the digital information ecosystem.

REFERENCES

[1] K. Shu, A. Sliva, S. Wang, J. Tang, and H. Liu, "Fake news detection on social media: A data mining perspective," ACM SIGKDD Explorations Newsletter, vol. 19, no. 1, pp. 22–36, 2017.

[2] V. L. Rubin, N. J. Conroy, Y. Chen, and S. Cornwell, "Fake news or truth? Using satirical cues to detect potentially misleading news," in Proc. NAACL-HLT Workshop on Computational Approaches to Deception Detection, San Diego, CA, USA, 2016, pp. 7–17.

[3] W. Y. Wang, "'Liar, liar pants on fire': A new benchmark dataset for fake news detection," in Proc. 55th Annu. Meeting Assoc. Computational Linguistics (ACL), Vancouver, Canada, 2017, pp. 422–426.

[4] N. Ruchansky, S. Seo, and Y. Liu, "CSI: A hybrid deep model for fake news detection," in Proc. ACM Conf. Inf. and Knowl. Manage. (CIKM), Singapore, 2017, pp. 797–806.

[5] J. Thorne and A. Vlachos, "FEVER: A large-scale dataset for fact extraction and verification," in Proc. NAACL-HLT, New Orleans, LA, USA, 2018, pp. 809–819.

[6] F. Pedregosa et al., "Scikit-learn: Machine learning in Python," J. Mach. Learn. Res., vol. 12, pp. 2825–2830, 2011.

[7] M. Choras, M. Pawlicki, R. Kozik, and M. Woźniak, "Advanced machine learning techniques for fake news detection: A systematic mapping study," Applied Sciences, vol. 10, no. 19, art. 6802, 2020.

[8] K. Nath, A. Ahuja, P. Soni, and R. Katarya, "Fake news detection using machine learning and deep learning classification methods," in Proc. IEEE Int. Conf. Computing, Communication and Networking Technologies (ICCCNT), 2021.

[9] P. Agarwal, R. Sharma, and M. K. Gupta, "Fake news detection: An investigation based on machine learning models," in Proc. IEEE Int. Conf. Information Reuse and Integration (IRI), 2022, pp. 120–125.

[10] J. Li, Y. Zhang, and X. Liu, "A brief survey for fake news detection via deep learning models," Procedia Computer Science, vol. 214, pp. 1339–1344, 2022.

[11] F. A. Alshuwaier, "Fake news detection using machine learning and deep learning approaches," Computers, vol. 14, no. 9, 2025.

[12] M. Beseiso, A. Al-Sadi, and H. Al-Hawari, "A context-enhanced model for fake news detection using LSTM and CNN," Engineering, Technology & Applied Science Research, vol. 15, no. 2, 2025.

[13] K. I. Roumeliotis et al., "Evaluating CNN architectures for fake news classification," placeholder — verify with actual 2024–2025 source before submission.

[14] H. Ahmed, I. Traore, and S. Saad, "Detection of online fake news using N-gram analysis and machine learning techniques," in Intelligent, Secure, and Dependable Systems in Distributed and Cloud Environments (ISDDC), LNCS, vol. 10618, Cham, Switzerland: Springer, 2017, pp. 127–138. (ISOT Fake News Dataset)

[15] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in Proc. ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining, San Francisco, CA, USA, 2016, pp. 785–794.

[16] L. Breiman, "Random forests," Machine Learning, vol. 45, no. 1, pp. 5–32, 2001.

[17] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in Proc. NAACL-HLT, Minneapolis, MN, USA, 2019, pp. 4171–4186.

[18] S. Vosoughi, D. Roy, and S. Aral, "The spread of true and false news online," Science, vol. 359, no. 6380, pp. 1146–1151, 2018.

[19] H. Allcott and M. Gentzkow, "Social media and fake news in the 2016 election," Journal of Economic Perspectives, vol. 31, no. 2, pp. 211–236, 2017.

[20] A. Zubiaga, A. Aker, K. Bontcheva, M. Liakata, and R. Procter, "Detection and resolution of rumours in social media: A survey," ACM Computing Surveys, vol. 51, no. 2, pp. 1–36, 2018.