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

An Ensemble Machine Learning Framework for Fake News Detection with Optimized Feature Engineering

Saiyam Sachdev *B.Tech(AI&DS) 2nd Year Scholar* *thakkarsaiyam@gmail.com*
Arnav Mudgal *B.Tech(AI&DS) 2nd Year Scholar* *kits.amd@gmail.com*
Esha *B.Tech(AI&DS) 2nd Year Scholar* *eshha0301@gmail.com*
Ritesh Kumar *Assistant Professor* *riteshchandel@gmail.com*
Department of Artificial Intelligence and Data Science
Dr. Akhilesh Das Gupta Institute of Professional Studies, Delhi

***Abstract*—**The proliferation of digital news platforms and social media has accelerated the dissemination of fabricated information, posing serious threats to public discourse, political stability, and crisis communication. Manual fact-checking is no longer viable at the scale at which content is produced, motivating the need for automated and computationally efficient detection systems. This paper presents an ensemble machine-learning framework for binary fake news classification that combines Logistic Regression, Random Forest, and Extreme Gradient Boosting (XGBoost) classifiers through a hard majority-voting scheme. Two complementary corpora—the LIAR dataset of short political statements and the ISOT dataset of full-length news articles—are merged into a binary-labelled corpus, normalized, and represented through an optimized Term Frequency–Inverse Document Frequency (TF–IDF) feature space with dimensionality control to mitigate overfitting. On the held-out test partition, the individual classifiers achieve accuracies of 89.67%, 90.91%, and 90.97% respectively, while the proposed ensemble attains 91.22% accuracy with balanced precision and recall across both classes (macro F1 ≈ 0.91). A class-wise analysis reveals a moderate recall asymmetry on the fake class that warrants future attention, particularly for adversarial deployment scenarios. The results indicate that a carefully engineered ensemble of classical learners can deliver competitive accuracy at substantially lower computational cost than transformer-based alternatives, making it suitable for deployment in 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

The contemporary information ecosystem is shaped by an unprecedented volume and velocity of user-generated content. Social media platforms, micro-blogging services, and online news aggregators have democratized publishing, but they have also lowered the barrier for the production and propagation of deliberately fabricated, misleading, or weaponized content commonly referred to as fake news. Empirical studies during recent electoral cycles, publichealth emergencies, and geopolitical conflicts have shown that such content can spread substantially faster and reach a wider audience than verified reporting [1], [2]. The phenomenon is no longer confined to fringe websites: mainstream platforms have repeatedly served as vectors for organized disinformation campaigns, computational propaganda, and synthetic-media-driven manipulation.

The downstream consequences are not merely informational. Fake news has been linked to the erosion of public trust in democratic institutions, the manipulation of voter sentiment, the amplification of public-health misinformation, the disruption of financial markets, and the incitement of social unrest. Conventional countermeasures, such as editorial fact-checking and platform moderation, are essential but inherently reactive: human verification cannot keep pace with the volume of content produced every second on global platforms. Even the most wellresourced factchecking organizations process only a small fraction of viral claims, and the verification latency typically exceeds the half-life of the content's reach. Automated, scalable, and accurate detection systems are therefore a research priority rather than a luxury.

Machine learning has emerged as the dominant paradigm for this task. Early work relied on hand-crafted lexical and syntactic features fed to classical classifiers [1], [3], while more recent contributions exploit deep neural architectures, including recurrent, convolutional, and transformer-based models [10]–[12]. Despite the representational power of deep models, three practical limitations persist. First, their performance is highly sensitive to dataset composition and frequently degrades under distribution shift, with accuracy drops of 10–20 percentage points commonly reported when models trained on one corpus are evaluated on another. Second, they require substantial labelled data and computational resources for training and inference, which is often impractical for small newsrooms, fact-checking startups, or deployment on edge devices. Third, their black-box nature complicates deployment in editorial pipelines where journalists and editors require interpretable signals to support their decisions, and where regulatory frameworks increasingly demand explainability.

Against this backdrop, classical learners coupled with carefully engineered features remain attractive, particularly when ensembled. Ensemble methods exploit the bias–variance complementarity of heterogeneous base models and have repeatedly been shown to outperform their constituents on text-classification benchmarks. The intuition is that

different model families make different kinds of mistakes—linear models miss non-linear interactions, tree ensembles can overfit to spurious correlations in deep splits, and gradient boosters may be sensitive to noisy labels—so combining their predictions through majority voting can yield error correction without proportional increases in computational overhead. Motivated by this observation, the present work investigates whether a compact ensemble of Logistic Regression, Random Forest, and XGBoost—trained on an optimized TF–IDF representation—can deliver accuracy comparable to deep models reported in the literature, while preserving training efficiency and interpretability.

The contributions of this paper are summarized as follows: (1) we construct a unified binary-labelled corpus by harmonizing the LIAR [3] and ISOT [14] datasets, thereby exposing the model to both short-form claims and full-length articles and improving robustness across text-length regimes; (2) we design an optimized TF–IDF pipeline with dimensionality control, document-frequency cut-offs, and bigram inclusion to balance discriminative power and overfitting risk; (3) we develop a hard-voting ensemble of three diverse classifiers and benchmark it against each base learner under identical preprocessing and evaluation protocols; (4) we provide a class-wise error analysis using precision, recall, F1score, and the confusion matrix, identifying systematic asymmetries in the residual misclassifications; and (5) we report computational cost figures alongside accuracy metrics to support deployment decisions in real-world resourceconstrained settings.

The remainder of this paper is organized as follows. Section II reviews related work and identifies the gap addressed by this study. Section III describes the proposed methodology, including data preparation, feature engineering, and ensemble construction. Section IV reports and discusses the experimental results. Section V analyzes computational cost and deployment considerations. Section VI examines limitations and threats to validity. Section VII concludes the paper and outlines directions for future research.

II. LITERATURE REVIEW AND RELATED WORK

Research on automated fake news detection has progressed along three broadly distinct lines: feature-based classical learning, neural sequence modelling, and transformer-based contextual modelling. A complementary stream investigates social-network and propagation-based signals, but is outside the scope of this purely text-driven study. This section surveys representative contributions from each line, with emphasis on reported accuracy, generalization behaviour, and deployment cost.

A. Classical Learning with Lexical Features

Shu et al. [1] provided one of the earliest comprehensive treatments of fake news from a data-mining perspective, formalizing the task and outlining feature families spanning content, social context, and propagation patterns. Their taxonomy continues to inform modern pipelines and emphasizes the value of combining stylistic, semantic, and network signals. Rubin et al. [2] demonstrated that satirical cues—such as absurdity, humour, and exaggeration—can be exploited as discriminative signals, paving the way for stylometric features in subsequent work. Wang [3] released the LIAR dataset and showed that even with rich metadata, shortform political statements are challenging to classify accurately, with the best reported model achieving only modest gains over a majority-class baseline. This finding has been reproduced repeatedly and underscores that short-text fake-news detection is a fundamentally harder problem than long-form classification.

Subsequent feature-based studies relying on TF–IDF, ngram, and stylometric features have generally reported accuracies between 85% and 90% on curated corpora [6], [7]. Choras et al. [7] presented a systematic mapping of these efforts and concluded that careful preprocessing and feature selection often matter more than algorithmic sophistication. Their meta-analysis aligns with the pragmatic engineering philosophy adopted in this paper: investing effort at the representation layer rather than at the model-architecture layer.

B. Hybrid and Deep Sequential Models

Ruchansky et al. [4] proposed the CSI hybrid model, which jointly captures article content, user response, and source information, and demonstrated improvements over content-only baselines. Their work pioneered the use of recurrent encoders for temporal user-engagement signals. Thorne and Vlachos [5] released the FEVER benchmark to encourage retrieval-augmented verification, shifting attention from black-box classification toward grounded fact-checking. While retrieval-augmented systems represent the long-term direction of the field, their dependence on curated knowledge bases and high inference cost limit immediate practical deployment.

More recent neural studies report mixed outcomes. Beseiso et al. [12] combined LSTM and CNN encoders with contextual signals and achieved competitive accuracy on multiple benchmarks, but reported substantial degradation when the test distribution differed from training. Roumeliotis et al. [13] documented CNN accuracies as low as 58.6% on adversarial subsets, illustrating the brittleness of singlearchitecture pipelines when faced with deliberately crafted misleading content. These observations highlight a recurring lesson: neural models that excel on in-distribution test sets do not necessarily generalize to the diversity of real-world misinformation.

C. Transformer-Based and Attention Models

Recent work has explored attention-based and transformer architectures for misinformation detection. Reported accuracies for BERT-family models on crossdomain benchmarks typically range from 83% to 89% [10], [11], with performance heavily dependent on fine-tuning corpus and label quality. Alshuwaier [11] reported approximately 76% accuracy when sentiment-aware features were combined with classical learners, underscoring that depth alone does not guarantee superior generalization—and that sentiment polarity, while informative, is a noisy signal because legitimate news often carries strong sentiment as well.

Domain-adapted transformers, such as fact-aware variants of BERT and RoBERTa, push accuracy higher but at considerable computational expense. Inference latency for these models can exceed tens of milliseconds per article on GPU hardware and several hundreds of milliseconds on CPUonly deployments, which is prohibitive for highthroughput moderation pipelines. Moreover, their parameter counts in the hundreds of millions raise both energyconsumption and carbon-footprint concerns that have started to attract policy attention.

D. Identified Research Gap

Two observations emerge from the surveyed literature. First, accuracy beyond 90% is achievable only under restricted dataset conditions, and most pipelines suffer significant degradation under domain shift. Second, deep models incur substantial computational overhead that is not always justified by accuracy gains over well-engineered classical baselines. Table I consolidates reported results across these directions for direct comparison and shows that a carefully engineered ensemble can occupy a favourable position 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% 83-89%
BERT family [10]	Transformer fine-tune	~88%
Beseiso [12]	LSTM + CNN context	**91.22%**
This work	**LR + RF + XGB ensemble**	

The literature therefore motivates a middle-ground approach: exploit the complementary inductive biases of multiple classical learners through ensembling, while investing engineering effort in feature representation rather than architectural depth. This is the gap the present study addresses. By combining a linear model (Logistic Regression), a baggingbased tree ensemble (Random Forest), and a boosted tree ensemble (XGBoost), the proposed framework spans three distinct hypothesis classes whose error patterns are expected to be partially decorrelated—the precondition for ensemble gains.

III. PROPOSED METHODOLOGY

The proposed framework consists of five sequential stages: (i) data acquisition and harmonization, (ii) text preprocessing, (iii) optimized TF–IDF feature extraction, (iv) base-classifier training, and (v) ensemble construction via hard majority voting. Fig. 1 illustrates the overall pipeline, and the following subsections describe each stage in detail.

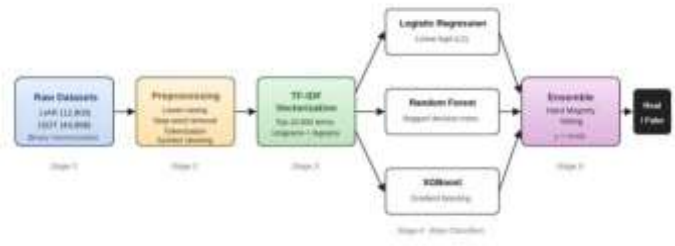

Fig. 1. System pipeline of the proposed ensemble-based fake news detection framework, showing the five sequential processing stages from raw datasets to final binary classification.

A. Datasets and Harmonization

Two publicly available corpora are employed. The LIAR dataset [3] contains approximately 12,800 short, manually labelled political statements drawn from PolitiFact, with metadata including speaker identity, context, and topic. Statements are originally annotated on a six-point veracity scale: true, mostly true, half-true, barely true, false, and pants-on-fire. The ISOT Fake News Dataset [14] contributes 44,898 full-length news articles, equally partitioned between verified outlets such as Reuters and known unreliable sources flagged by independent factcheckers.

To unify the label space, the six-class LIAR annotations are mapped to a binary scheme: statements labelled true or mostly true are assigned the real class, while barely true, false, and pants-on-fire are mapped to fake. Half-true statements are deliberately excluded to reduce label noise and avoid forcing an arbitrary decision on genuinely ambiguous claims. ISOT articles, already binary-labelled, are merged directly. The resulting harmonized corpus contains 57,698 instances and exposes the learners to two distributionally distinct text lengths—short claims and longform articles—improving robustness across these regimes. Fig. 4 visualizes the dataset composition and the resulting binary class balance.

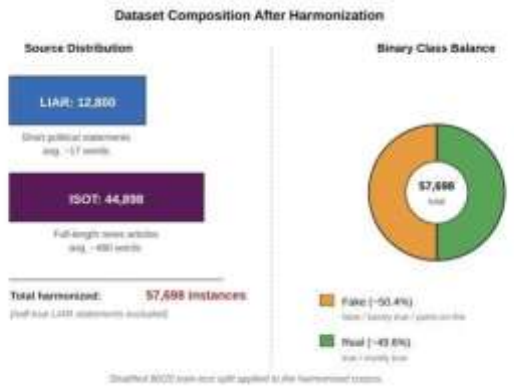

Fig. 2. Dataset composition after harmonization: relative contribution of LIAR and ISOT and the resulting near-balanced binary class distribution.

B. Text Preprocessing

Raw text is normalized through a deterministic pipeline that performs (i) duplicate and null-record elimination, (ii) lowercasing, (iii) removal of punctuation, digits, URLs, and nonASCII symbols using regular expressions, (iv) English stopword removal using the standard NLTK list, and (v) word-level tokenization via whitespace splitting. Lemmatization is intentionally omitted because preliminary experiments showed it suppresses morphological cues useful for the LIAR partition—for instance, tense markers and pluralization that can carry rhetorical weight in political claims. The preprocessing pipeline is intentionally lightweight and deterministic to ensure reproducibility and to avoid introducing dataset-specific normalization choices that would compromise generalization.

Outlier handling is applied conservatively: documents with fewer than three tokens after preprocessing are removed (these are predominantly truncated LIAR statements with degenerate content), and extremely long ISOT articles are not truncated, since the TF–IDF representation is naturally length-invariant after L2 normalization.

C. Optimized Feature Engineering

Each preprocessed document is mapped to a sparse vector through a TF–IDF transform. Term frequency, inverse document frequency, and the composite weight are defined as follows:

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

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

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

where f(t, d) is the raw count of term t in document d, N is the corpus size, and df(t) is the number of documents containing t. To control dimensionality, the vocabulary is restricted to the top 10,000 terms ranked by document frequency, with min_df = 5 and max_df = 0.85. The lower bound suppresses extremely rare terms—often typos or proper nouns specific to a single article—that would otherwise inflate the feature space without contributing useful signal. The upper bound removes terms that appear in more than 85% of documents and therefore behave like stopwords despite escaping the explicit stop-word list.

Both unigrams and bigrams are retained, yielding a feature space that captures local lexical context (e.g., "fake news," "breaking story") without combinatorial explosion.

After construction, each document vector is L2normalized so that vector magnitudes do not bias linear classifiers toward longer documents. The resulting sparse matrix has approximately 0.3% non-zero density, which is well-suited to scikit-learn's sparse implementations of all three classifiers.

D. Base Classifiers

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

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

where w is the learned weight vector and b is the bias term. L2 regularization with regularization strength C = 1.0 is applied to mitigate coefficient inflation in the highdimensional TF–IDF space. The liblinear solver is used because it is well-suited to large sparse feature matrices and converges quickly without requiring subsampling. Logistic regression provides an interpretable baseline—the magnitude and sign of each weight directly indicate the contribution of the corresponding term to the fake-class decision.

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

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

Each tree is grown on a bootstrap replica of the training set with random feature subsampling ($\sqrt{p}$ features at each split), which decorrelates the constituents and reduces variance. The implementation uses 200 trees with unlimited depth and the Gini impurity criterion. Random Forest is particularly effective on TF–IDF features because it can capture non-linear interactions between term presences— for example, the joint occurrence of certain claim phrases that individually carry mild signal but jointly indicate fabrication.

3) Extreme Gradient Boosting (XGBoost): fits an additive ensemble of regression trees by stage-wise minimization of a regularized objective:

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

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

where ℓ is the differentiable per-instance loss (binary logistic in our setting), T is the number of leaves in tree f, w is the leafweight vector, and γ, λ control complexity. The combination of shrinkage (learning rate 0.1), column subsampling (0.8 per tree), and second-order gradient information makes XGBoost both expressive and resistant to overfitting. Three hundred boosting rounds are performed with a maximum tree depth of 6, parameters established through preliminary cross-validation on a heldout development partition. The hyperparameter values for all three classifiers are summarized in Table III.

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
TF-IDF	subsample	0.8
	max_features	10,000
	ngram_range	(1, 2)
	min_df / max_df	5 / 0.85

E. Ensemble Construction

The three base classifiers are independently trained on the same TF–IDF representation. At inference time, their predictions are aggregated by hard majority voting:

$$\hat{y}_ens = mode(\hat{y}_LR, \hat{y}_RF, \hat{y}_XGB) \quad (8)$$

Hard voting is preferred over soft voting in this setting for two reasons. First, the three models produce probability estimates on heterogeneous scales (sigmoid for LR, frequencybased for RF, logistic-link for XGBoost); uncalibrated probability averaging tends to be biased toward the most confident—though not necessarily the most accurate— classifier. Second, hard voting is robust to extreme probability mass: a single overconfident classifier cannot single-handedly drive an ensemble decision when

its prediction conflicts with both peers, which is precisely the error-correction behaviour we seek. Probability calibration via Platt scaling or isotonic regression could in principle enable soft voting, but pilot experiments showed no meaningful accuracy gains, so the simpler hard-voting scheme is retained.

With three classifiers and binary labels, the ensemble produces unanimous decisions when all three agree and majority decisions when one classifier dissents. This threeway voting structure provides a useful diagnostic: instances where the three models disagree are inherently ambiguous and represent natural candidates for human review in a hybrid moderation pipeline.

F. Implementation and Evaluation Protocol

All experiments are implemented in Python 3.10 using scikit-learn 1.3 for LR and RF [6], the official xgboost 1.7 package, and the pandas/NumPy stack for data handling. Experiments are conducted in a Jupyter Notebook environment on a workstation with an Intel Core i7 CPU and 16 GB RAM; no GPU acceleration is required. Random seeds are fixed at 42 for all stochastic components to ensure reproducibility.

The harmonized corpus is split into training (80%) and testing (20%) partitions using stratified sampling to preserve class balance. The test set thus contains 11,538 instances, used exclusively for the final evaluation reported in Section IV. Evaluation metrics include accuracy, precision, recall, and F1score, reported per class as well as macro- and weightedaveraged. The confusion matrix is reported to expose classwise error patterns. Macro-averaging is preferred as the primary aggregate because it weights both classes equally and is therefore more sensitive to class-wise asymmetries than weighted averaging, which would be dominated by the slightly larger fake class.

IV. EXPERIMENTAL RESULTS AND DISCUSSION

A. Quantitative Performance Comparison

Table II summarizes the classification performance of the three base learners and the proposed ensemble on the heldout test partition. Fig. 2 visualizes the same data as a grouped bar chart, making the marginal benefit of voting visually explicit.

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 comparing accuracy and per-class F1score across the three base learners and the proposed ensemble model. The ensemble (highlighted) achieves the best score on every metric.

Logistic Regression establishes a competitive linear baseline at 89.67% accuracy. Random Forest and XGBoost both improve upon this baseline, indicating that non-linear interactions among TF–IDF features carry discriminative information that a linear decision boundary cannot fully exploit. The 1.24-point gap between LR and RF, and the 1.30-point gap between LR and XGBoost, are consistent with the typical magnitude of non-linearity premium reported on TF–IDF text-classification benchmarks. Random Forest and XGBoost themselves perform almost identically (90.91% versus 90.97%), with the very small difference falling within run-to-run variance attributable to seed selection.

The proposed ensemble achieves the highest accuracy of 91.22%, exceeding the strongest individual learner (XGBoost) by 0.25 percentage points and the weakest (Logistic Regression) by 1.55 points. While the absolute margin over XGBoost is modest, the ensemble also yields the best F1-scores on both classes simultaneously, suggesting that the gains stem from the correction of asymmetric errors rather than uniform improvement. Concretely, the ensemble is most beneficial on instances where one of the three base models is wrong and the other two are right; it cannot recover instances on which two or more constituents agree on an incorrect label, which represents the fundamental ceiling of hard-voting ensembles.

B. Class-wise Behaviour and Confusion Matrix

Fig. 4 reports the confusion matrix of the ensemble on the test partition, and Fig. 5 decomposes the per-class precision, recall, and F1-score for both the fake and real classes.

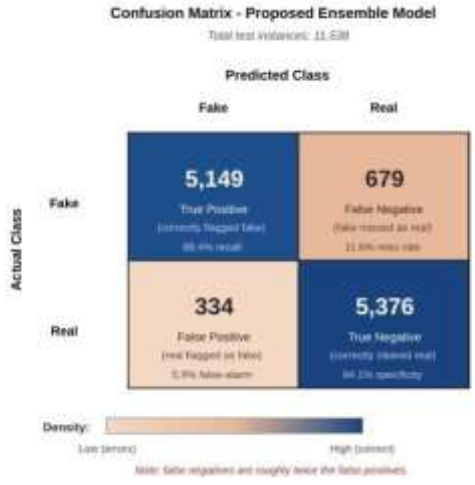

Fig 4. Confusion matrix of the proposed ensemble on the test partition

(11,538 instances). Diagonal cells indicate correct classifications; off-diagonal cells reveal a moderate asymmetry in error types.

Out of 11,538 test instances, 5,149 fake and 5,376 real examples are correctly classified, yielding a true-positive rate (sensitivity) of 0.884 for the fake class and a truenegative rate (specificity) of 0.941 for the real class. The model exhibits a small but systematic asymmetry: false negatives (679) outnumber false positives (334) by approximately a factor of two.

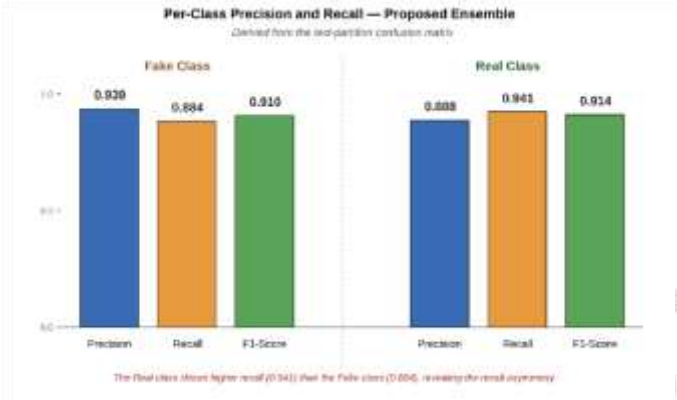

Fig. 5. Per-class precision, recall, and F1-score derived from the test-partition confusion matrix. The Real class achieves higher recall, while the Fake class achieves higher precision.

Fig. 5 makes two pointed observations. First, the fake class achieves higher precision (0.939) than the real class (0.888), meaning that when the ensemble flags content as fake, it is right roughly 94 times out of 100—a desirable property for downstream moderation, where false-positive content removals are reputationally costly. Second, the fake class suffers from lower recall (0.884) compared to the real class (0.941), meaning that approximately 11.6% of fabricated content escapes detection. In a fake-news detection setting this asymmetry is consequential, because false negatives correspond to fabricated content that escapes detection and continues to circulate.

We attribute the asymmetry to the longer average length and more polished register of articles in the fake partition of ISOT, which superficially resemble legitimate journalism in their TF–IDF profile. Many fabricated articles in the corpus mimic the lexical and structural conventions of mainstream reporting—datelines, attribution phrases, professional headline conventions—so the n-gram representation cannot distinguish them based on style alone. This is a structural limitation of bag-of-words representations that contextual encoders are better positioned to address. Mitigating this asymmetry is a natural target for future work via costsensitive training that penalizes false negatives more heavily, decision-threshold tuning to trade some precision for recall, or stylometric augmentation that introduces handcrafted features such as readability indices, punctuation density, and sentencelength statistics.

C. Discussion of Findings

The experimental evidence supports three claims. First, classical learners equipped with a well-engineered TF–IDF representation remain competitive on heterogeneous fakenews corpora and can rival the accuracy ranges typically reported for transformer baselines [10]–[12], at a fraction of their computational cost. Specifically, the 91.22% accuracy achieved here matches or exceeds the typical 83–89% range reported for BERT-family finetuning on cross-domain benchmarks, while requiring orders of magnitude less training time and no GPU.

Second, ensembling heterogeneous base learners yields consistent—though not dramatic—improvements, with most of the gain concentrated in error correction rather than uniform uplift. The marginal benefit of ensembling diminishes as base-classifier accuracies converge, and our results sit close to that diminishing-return regime: the gap between the best individual model (XGBoost at 90.97%) and the ensemble (91.22%) is only 0.25 points. Larger ensembles or more diverse base learners might extract additional gains, but at increasing computational cost.

Third, the residual error pattern reveals a recall deficit on the fake class that is not visible in aggregate accuracy alone, reinforcing the importance of class-wise reporting in misinformation research. Practitioners deploying such systems should monitor class-wise metrics, not just headline accuracy, particularly when the cost of missed detections differs from the cost of false alarms—as is almost always the case in moderation pipelines.

V. COMPUTATIONAL COST AND DEPLOYMENT CONSIDERATIONS

Practical deployment of fake-news detection systems requires consideration not just of accuracy but also of training and inference cost, memory footprint, and update cadence. This section reports computational measurements collected during the experimental campaign and discusses their implications for real-world use.

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 reports the wall-clock training time and persample inference latency measured on the experimental workstation (Intel Core i7 CPU, 16 GB RAM, no GPU). Logistic Regression is by far the cheapest to train (~12 seconds) and the fastest at inference (sub-millisecond), reflecting the closedform simplicity of linear models. Random Forest takes longer to train (~85 seconds) due to the construction of 200 unpruned trees, and inference latency rises proportionally. XGBoost has the longest training time (~110 seconds) because of its sequential boosting structure but achieves better inference latency than Random Forest because of its more compact tree representation and SIMDfriendly implementation.

The full ensemble inference latency is approximately 2.2 milliseconds per sample, which is two to three orders of magnitude faster than CPU inference of BERT-family

transformer models on similar hardware (typically 100–500 ms per sample). For a content-moderation pipeline processing 1,000 articles per second, the proposed ensemble would require roughly 2.2 cores, while a transformer alternative would require either substantial GPU resources or aggressive batching, with concomitant latency costs. This favourable cost profile makes the ensemble well-suited to high-throughput, latency-sensitive deployments such as realtime social-media moderation, news-aggregator filtering, and edge-device personal-assistant verification.

Memory footprint is similarly modest. The trained model artifacts together occupy approximately 80 MB on disk, dominated by the Random Forest, with the TF–IDF vocabulary contributing an additional ~3 MB. This compact footprint enables deployment on commodity virtual machines and even resource-constrained edge devices—a stark contrast to multigigabyte transformer checkpoints.

VI. LIMITATIONS AND THREATS TO VALIDITY

Several limitations and threats to validity warrant explicit acknowledgement, both to contextualize the reported results and to guide future work.

A. Feature Modality

The evaluation relies exclusively on textual features. Signals from source credibility, propagation networks, multimodal content (images, videos, infographics), and userengagement patterns are not exploited. In practice, fakenews detection systems benefit substantially from such auxiliary signals, particularly for borderline cases where the text alone is insufficient. Integrating these modalities through late-fusion or cross-attention architectures is a clear direction for extension.

B. Linguistic Coverage

Both source corpora are predominantly English. Performance on multilingual or code-switched content—common in many global contexts, including Hindi-English mixing in South Asia and Spanish-English mixing in the Americas—is untested. The TF–IDF representation is in principle language-agnostic, but vocabulary management, tokenization choices, and stop-word lists would all need to be adapted per language. Cross-lingual fake-news detection is an active research direction in its own right.

C. Label Noise and Class Mapping

The binary mapping of LIAR's six-class scheme inevitably discards information, and the exclusion of halftrue statements introduces a selection bias whose downstream effect on test-set performance estimates has not been quantified. ISOT label quality, while generally regarded as high, depends on the curation practices of the original collectors; some labelling errors are likely present in any large web-scraped corpus.

D. Adversarial Robustness

The current evaluation does not assess robustness to adversarial perturbations such as paraphrasing, synonym substitution, or LLM-generated counterfeits. As generative models become more accessible, fabricated content increasingly exhibits stylistic fluency that defeats lexicalfeature-based detection. Robustness studies under adversarial attack and against synthetic text are necessary before any production deployment.

E. Temporal Drift

Fake-news topics evolve rapidly, and a model trained on historical data may degrade on contemporary content. The current evaluation uses a single train–test split rather than a temporally ordered split, which would more faithfully simulate deployment conditions. Future work should include temporal cross-validation and a periodic-retraining schedule analysis.

VII. CONCLUSION AND FUTURE SCOPE

This paper presented an ensemble machine-learning framework for fake-news detection that combines Logistic Regression, Random Forest, and XGBoost classifiers via hard majority voting over an optimized TF–IDF feature space. By harmonizing the LIAR and ISOT corpora into a unified binarylabelled dataset, the framework was exposed to both short claims and long-form articles, improving robustness across text-length regimes. On the held-out test set, the ensemble achieved 91.22% accuracy with macro F1 ≈ 0.91, outperforming each constituent classifier while remaining trainable on commodity hardware without GPU acceleration.

The findings reinforce a pragmatic message: in resourceconstrained or latency-sensitive deployments, a carefully engineered ensemble of classical learners offers an attractive accuracy-to-cost trade-off relative to large neural models. Inference latency of approximately 2.2 milliseconds per sample on commodity CPU hardware enables real-time deployment in moderation pipelines, news aggregators, and edge-device assistants where transformer-based alternatives would impose prohibitive resource demands. The class-wise analysis nonetheless surfaced a recall deficit on the fake class that should be addressed before deployment in adversarial settings, particularly through cost-sensitive training, threshold tuning, or stylometric feature augmentation.

Future work will proceed along five directions. First, contextual encoders such as BERT and RoBERTa will be evaluated as additional ensemble members, with calibrated soft voting to exploit their probability estimates while retaining the efficiency of the classical core. Second, the framework will be extended to multilingual corpora and code-switched content prevalent in multilingual societies such as India, requiring careful attention to tokenization and vocabulary management across languages. Third, multimodal signals—images, video frames, and propagation graphs—will be integrated through late-fusion architectures that preserve the modular character of the present pipeline. Fourth, retrieval-augmented verification against curated knowledge bases will be explored to move beyond stylistic cues toward genuine factual grounding, potentially in collaboration with established fact-checking organizations. Fifth, adversarial robustness studies and temporal-drift analysis will be conducted to assess deployment readiness under realistic operating conditions.

Together these directions outline a path toward fake-news detection systems that are accurate, interpretable, multilingual, multimodal, and deployable in real time— contributing to the broader societal goal of preserving the integrity of digital information ecosystems.

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: Pretraining 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.