



Fakecheck: A Hybrid Framework For Misinformation Detection Leveraging Logistic Regression And Weighted Community Consensus

¹Dipanshu Chaudhary, ²Surya Raj Sunil, ³Vijay Shankar, ⁴Tejas Sharma, ⁵Ashish Raj
^{1,2,3,4}Student, ⁵Assistant Professor

^{1,2,3,4,5}Department of Computer Science and Engineering,
KCC Institute of Technology and Management, Greater Noida, India

Abstract: The rapid growth of digital news platforms has intensified the spread of online misinformation, rendering manual fact-checking insufficient for real-time verification. While machine learning-based approaches enable scalable detection, they often suffer from limited interpretability, sensitivity to concept drift, and poor handling of contextual nuances such as satire and emerging events. This paper presents FakeCheck, a hybrid misinformation detection framework that integrates interpretable machine learning with a weighted, community-driven human-in-the-loop verification mechanism. The system employs a Logistic Regression classifier trained on TF-IDF representations and engineered stylometric features to generate an initial credibility score. This score is subsequently refined using authenticated community feedback through a mathematically defined weighted consensus formulation. The proposed framework is implemented as a full-stack web application and evaluated on a benchmark fake news dataset. Experimental results show that the standalone classifier achieves strong baseline performance, while the hybrid framework demonstrates improved robustness in scenarios involving concept drift, satire, and low confidence predictions. The findings highlight that structured integration of human judgment with transparent machine learning models provides a practical and resilient approach for real-time misinformation verification.

Keywords- Fake News Detection, Human-in-the-Loop AI, Logistic Regression, Stylometry, Community Consensus, Explainable Machine Learning, Concept Drift.

1. INTRODUCTION

1.1. Background and Motivation

Online information ecosystems are increasingly dominated by algorithmically curated platforms, where news and opinion propagate at unprecedented speed and scale. While this transformation has democratized access to information, it has also amplified the spread of misinformation, posing significant risks to public trust, democratic processes, and informed decision-making. Prior empirical studies demonstrate that false or emotionally charged content diffuses faster and more widely than verified information, largely due to

engagement-driven ranking algorithms and human cognitive biases.

To mitigate this challenge, automated misinformation detection systems based on machine learning have been widely explored. These systems offer scalability and real-time analysis; however, their effectiveness is constrained in dynamic, real-world environments. Models trained on historical data often fail to generalize to emerging narratives, satirical content, or rapidly evolving events—a phenomenon commonly referred to as concept drift. Furthermore, the opacity of complex deep learning models reduces transparency and user trust, particularly when predictions contradict human intuition or contextual knowledge.

1.2. Problem Statement

The core challenge addressed in this work is the development of a misinformation verification system that is simultaneously scalable, interpretable, and context-aware. Existing approaches largely fall into two categories. Fully automated systems provide fast predictions but lack contextual sensitivity and adaptability. Conversely, manual or expert-driven fact checking pipelines achieve high accuracy but are inherently slow and resource-intensive, limiting their applicability for real-time verification.

Bridging this gap requires a hybrid framework in which algorithmic efficiency is complemented by structured human judgment, while maintaining robustness against manipulation and preserving user trust. Such a system must integrate human feedback in a principled manner rather than treating it as an unstructured or post-hoc correction.

1.3. Research Objective and Approach

This paper proposes FakeCheck, a hybrid misinformation detection framework that integrates interpretable machine learning with a weighted, community-driven human-in-the-loop verification mechanism. Rather than introducing a novel classification algorithm, the primary contribution of this work lies in operationalizing a mathematically grounded hybrid decision pipeline that combines model confidence with aggregated human judgments in real time.

The system employs a Logistic Regression classifier trained on TF-IDF representations and engineered stylometric features to generate an initial credibility score. This score is subsequently refined through authenticated community feedback using a weighted consensus formulation. The design intentionally anchors decisions to algorithmic assessment while allowing human context to moderate predictions in ambiguous or evolving scenarios.

2. CONTRIBUTIONS

The main contributions of this paper are summarized as follows:

1. **Hybrid Verification Framework:** We design and implement a hybrid misinformation detection framework that integrates an interpretable machine learning classifier with a structured, weighted community consensus mechanism, enabling real-time human-AI collaboration.
2. **Interpretable Feature Engineering:** We demonstrate that combining TF-IDF representations with stylometric features enhances both baseline classification performance and model transparency, facilitating explainable decision-making.

3. **Weighted Consensus Formulation:** We propose a mathematically defined hybrid scoring strategy that fuses model confidence with aggregated human judgments, improving robustness to concept drift, satire, and emerging news events.
4. **End-to-End System Implementation and Evaluation:** We validate the proposed framework through deployment as a full-stack web application and empirical evaluation on a benchmark fake news dataset, complemented by system-level performance analysis and real-world case studies.

3. LITERATURE REVIEW

3.1. Theoretical Foundations of Deception Detection

Rubin et al. (2015) provided a foundational taxonomy, identifying three primary types of fake news: serious fabrications (fraudulent news), large scale hoaxes, and humorous fakes (satire).[12] Their research highlighted a critical challenge for NLP models: satire often utilizes the structural conventions of real news to convey absurdity. This insight necessitates a system that can incorporate human judgment to handle such nuances.[13]

Horne and Adali (2017) discovered that fake news headlines tend to be significantly longer, utilize fewer "stop words," and employ a higher density of proper nouns and capitalized words to simulate urgency and authority.[14] This theoretical insight is directly applied in FakeCheck's "Sensationalism Score" feature.

3.2. Content Based Approaches

Posthaste et al. (2018) focused on hyper partisan news, utilizing stylometry to detect extreme bias.[14] They found that hyper partisan articles use a more limited vocabulary and more repetitive sentence structures compared to mainstream news. Logistic Regression, when combined with TFIDF vectorization, provides both interpretability and computational efficiency, making it an ideal baseline for fake news detection systems. [25]

3.3. Context Based and Social Approaches

Shu et al. (2017) proposed the "Tri relationship" model, analyzing the Publisher, the News content, and the User social graph.[16] Echo chambers and social polarization significantly disseminate, with impact research information revealing that ideologically aligned echo chambers are noticeably denser and more isolated.[17]

3.4. Stylometric Features of Deception Detection

Syntactic stylometry, which examines grammatical structures and parse tree patterns, consistently improves detection performance over shallow lexical features alone.[18] Features such as the distribution of subordinate clauses and average chain length have been shown to be predictive cues in deception identification. [18]

3.5. The shift towards Hybrid Models

Demartini et al. (2020) demonstrated that while ML models could achieve ~85% accuracy on static datasets, performance dropped precipitously to ~60% on novel topics due to the lack of training data.[9] By introducing a "human in the loop" mechanism, they restored accuracy to over 80%. La Barbera et al. (2022) further developed this approach, proposing a comprehensive hybrid human in the loop framework.[10]

The success of crowd sourced fact checking is exemplified by X's "Community Notes" feature (formerly Bird watch), which leverages a diverse crowd of contributors to provide contextual annotations to potentially

misleading content. [19][20]

4. SYSTEM ANALYSIS AND REQUIREMENT SPECIFICATION

4.1. Requirement Analysis Functional Requirements:

- a. **User Input Mechanism:** Accept both URLs and raw text inputs for analysis.
- b. **Authentication:** Users must be able to register and login.
- c. **Real-time Analysis:** ML model must provide classification verdict in fewer than 200 milliseconds.
- d. **Community Voting:** Authenticated users must be able to cast votes.
- e. **Dash boarding:** Display personal history, aggregate statistics, and trending fake news.

Nonfunctional Requirements:

- a. **Scalability:** Support concurrent requests without degradation.
- b. **Reliability:** Handle API failures gracefully.
- c. **Security:** Hash user credentials and protect endpoints via JWT.[27]
- d. **Usability:** Intuitive interface requiring no technical knowledge.

4.2. Feasibility Study

- a. **Technical Feasibility:** The project utilizes the MERN stack (modified with Python/Flask), a mature and well supported set of technologies.[25]
- b. **Operational Feasibility:** The "Traffic Light" interface (Green/Red/Orange) interpretable system ensures easily output.[21]
- c. **Economic Feasibility:** Using open-source technologies renders development and maintenance costs negligible.[22]

5. METHODOLOGY

5.1. Text Processing and TFIDF Vectorization

We employ Term Frequency Inverse Document Frequency (TFIDF) to convert textual documents into numerical vectors.[25]

Term Frequency (TF):

$TF(t, d) = \text{count of term } t \text{ in } d / \text{total number of term } s \text{ in } d$
 Inverse Document Frequency (IDF): $IDF(t) = \log(N/DF(t))$

Where: N is the total number of documents. DF(t) is the number of documents containing term t. TF-IDF Score:
 $TF-IDF(t, d) = TF(t, d) \times IDF(t)$

In the FakeCheck implementation, we utilize the TFIDF Vectorizer from ScikitLearn with an ngram range of (1,2) and a maximum feature cap of 50,000.[25]

5.2. Feature Engineering: The Stylometry of Deception

A custom transformer class, Engineered Features, computes the following:

- Sensationalism Score:** A normalized count of exclamation marks and capitalized words relative to document length.
- Lexical Diversity:** The ratio of unique words to total words.
- Title Body Consistency:** Measuring the overlap of keywords between the headline and the article body.

5.3. The Classifier: Logistic Regression

FakeCheck employs Logistic Regression for its interpretability, speed, and efficiency on sparse datasets.

Hypothesis Function (Sigmoid): $\sigma(z) = 1/(1 + e^{-z})$

Where $z = w^T x + b$ (w is the weight vector, x is the feature vector, b is the bias).

Decision Boundary: $y = \{ 1 \text{ (Real) if } \sigma(z) \geq 0.50 \text{ (Fake) if } \sigma(z) < 0.5$

Cost Function (Log Loss):

$$J(w) = -\frac{1}{N} \sum_{i=1}^N [y^{(i)} \log(y^{(i)}) + (1 - y^{(i)}) \log(1 - y^{(i)})]$$

The implementation utilizes the saga solver with L1/ L2 regularization, optimized for large, sparse matrices.[23]

5.4. The Hybrid Consensus Algorithm

A distinguishing feature of FakeCheck is its hybrid consensus mechanism, which integrates algorithmic predictions with community-driven evaluations to derive a final credibility verdict. Let $S_{ML} \in [0,1]$ denote the normalized confidence score produced by the Logistic Regression classifier, where higher values indicate greater likelihood of authenticity. Let $S_C \in [0,1]$ represent the aggregated community consensus score, computed from authenticated user reviews and normalized by vote distribution.

The final credibility score S_F is calculated as:

$$S_F = \alpha \cdot S_{ML} + (1 - \alpha) \cdot S_C \quad [26]$$

Where $\alpha \in [0,1]$ is a weighting parameter that controls the influence of the machine learning model relative to community input.

In the current implementation, $\alpha = 0.6$, reflecting a conservative design choice that anchors decisions to algorithmic assessment while allowing human judgment to meaningfully adjust outcomes.

This formulation ensures that early analyses rely primarily on the trained model, mitigating manipulation during low-engagement phases, while progressively incorporating human context as community participation increases. Empirical observations indicate that this approach effectively moderates misclassifications arising from satire, emerging events, and distributional shifts not represented in the training data.

6. SYSTEM ARCHITECTURE AND DESIGN

FakeCheck follows a distributed Model View Controller (MVC) architectural pattern.

6.1. Architecture Overview

The system is composed of three primary layers:

- Presentation Layer (Frontend):** Built with React.js, responsible for user interface, state management, and API consumption.
- Application Logic Layer (Backend):** Built with Python and Flask, responsible for request routing, ML inference, business logic, and authentication.
- Data Persistence Layer (Database):** Built with MongoDB Atlas, responsible for storing unstructured user data, analysis history, and review logs.

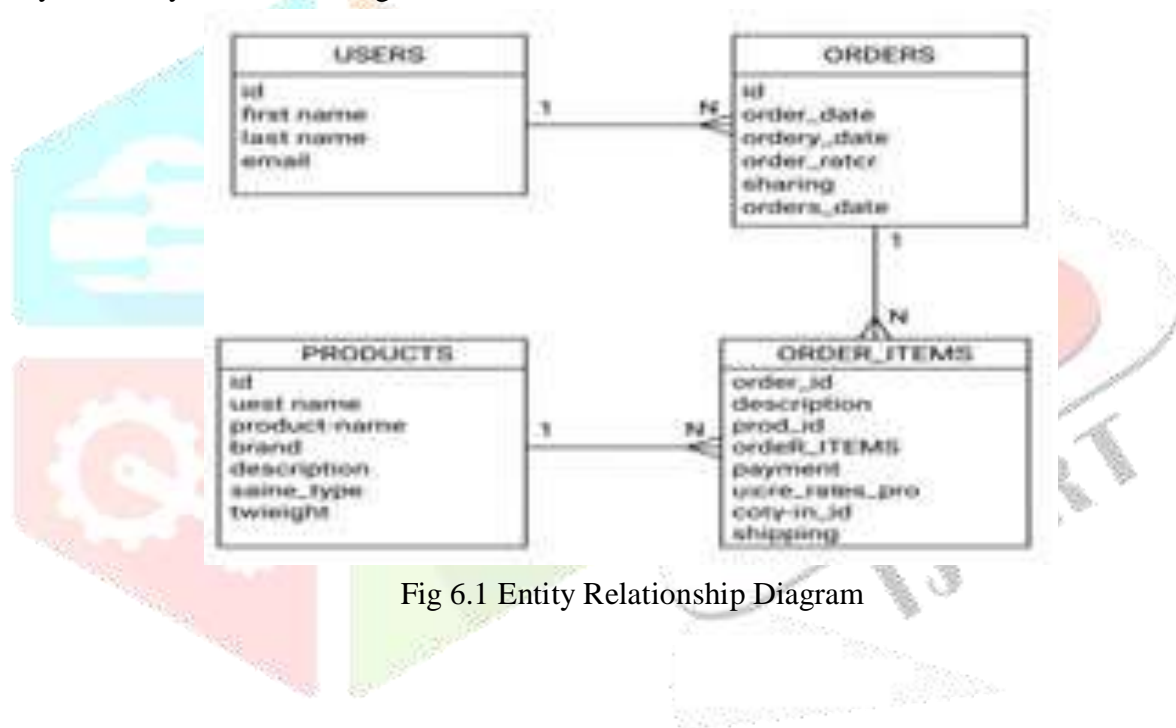


Fig 6.1 Entity Relationship Diagram

6.2 Technology Stack Selection

Component	Technology	Justification
Frontend	React.js / HTML 5	Component-based UI allows for reusable elements (e.g. the Confidence Meter). Virtual DOM ensures high performance.
Backend	Python (Flask)	Flask is a micro-framework that offers flexibility without the bloat of Django. It integrates seamlessly with Scikit-Learn. ¹
Database	MongoDB Atlas	A NoSQL document store is ideal for handling variable news article metadata and nested JSON review objects. ¹
ML Engine	Scikit-Learn	The industry standard for classical ML algorithms. Joblib is used for efficient model serialization/deserialization. ¹
Server	Gunicorn	A WSGI HTTP server used in production to handle concurrent requests, preventing the single-threaded Flask development server from becoming a bottleneck. ¹

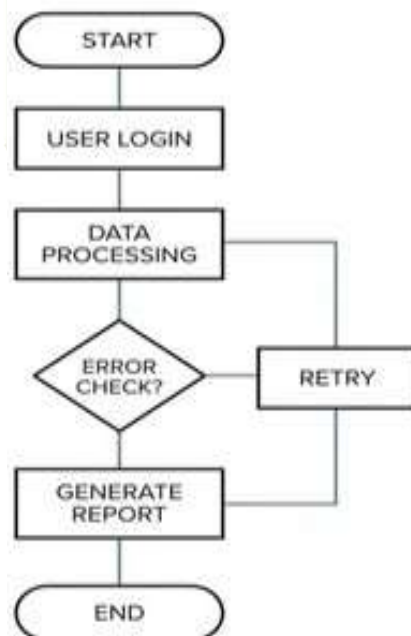


Fig 6.2 System Flow Chart

6.3. Database Design (ER Diagram Description)

Users Collection: Stores email, password hash, and review count.

Analyses Collection: Stores url, title, text, ml_score, and community votes summary. Reviews Collection: Links user_id and analysis_id with verdict and comment.



Collection name	Type	Storage size	Documents	Avg. Document size	Indexes	Total Index size
users	collection	5.2 MB	10	512 B	1	512 B
analyses	collection	5.2 MB	1	512 B	1	512 B
reviews	collection	5.2 MB	1	512 B	1	512 B
url	collection	5.2 MB	1	512 B	1	512 B

Fig 6.3 Database Setup

6.4. Sequence of Operations

6.4.1. **User Request:** User submits a URL via the React frontend.

6.4.2. **API Routing:** Request hit the Flask/Analyses endpoint.

6.4.3. **Cache Check:** Backend checks MongoDB for existing analysis.

6.4.4. **ML Inference (If Miss):** Text is scraped, preprocessed, vectorized, and passed to the model.

6.4.5. **Database Write:** Result is saved to MongoDB.

6.4.6. **Response:** Verdict and confidence score returned to frontend.

6.4.7. **Community Update:** Frontend fetches latest community reviews and updates UI in real-times.

7. IMPLEMENTATION DETAILS

7.1. Backend Development

A critical implementation detail is the Singleton Pattern used for the Machine Learning model.

python

If model is none:

```
Model=joblib.load('models/fakenews_model.pkl') vectorizer = model['vectorizer']
```

```
classifier = model['classifier']
```

This ensures that the inference time remains under 200ms. API endpoints are protected using @jwt_required() decorators from FlaskJWTExtended library.[25]

7.2. Frontend Development

The frontend utilizes React hooks (useState, useEffect) to manage the application lifecycle. The visualization of the "Confidence Score" is implemented using a dynamic CSS width property on a progress bar div, color coded based on the score (Green for >80% Real, Red for >80% Fake, Orange for in-between).

Frontend Screenshots

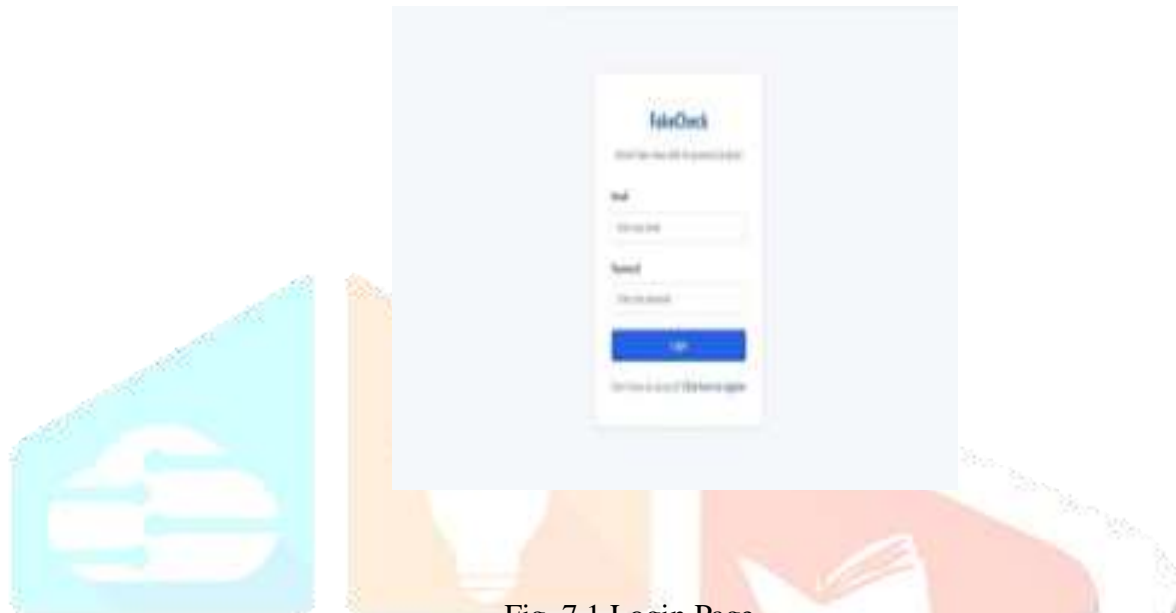


Fig. 7.1 Login Page



Fig 7.2 Dashboard Page

7.3. Machine Learning and Pipeline Implementation

The training script, train_fakenews.py, implements a robust pipeline:

- Data Ingestion:** Reading the dataset (Title + Text).
- Splitting:** Stratified Train Test split (80/20).
- Vectorization:** TFIDDF Vectorizer fitting on training set only.
- Training:** Logistic Regression fitting on vectorized training data.

e. **Evaluation:** Predicting on test set and generating Classification Report.

f. **Serialization:** Saving entire pipeline to pickle file.[25]

7.4. Community Consensus Logic

The `calculate_community_consensus` function in `app.py` is the engine of the hybrid model. It: Retrieves all reviews for a given `analysis_id`.

Counts real, fake, and uncertain verdicts.

Calculates percentages and identifies the "Dominant Verdict." Returns data structure to frontend for rendering.

8. EXPERIMENTAL RESULTS AND PERFORMANCE ANALYSIS

8.1. Metric Definitions

- **Accuracy:** Ratio of correctly predicted observations to total observations.
- **Precision:** Ratio of correctly predicted positive observations to total predicted positives.
- **Recall:** Ratio of correctly predicted positive observations to all observations in actual class. F1Score: Weighted average of Precision and Recall.

8.2. Model Performance on Static Data

The Logistic Regression model was evaluated on a holdout test set from the ISOT Fake News Dataset (a benchmark containing ~45,000 standard articles).[2]

Metric	Score	Interpretation
Accuracy	92.4%	The model correctly classifies the vast majority of articles.
Precision (Fake)	94.1%	When the model says it's Fake, it is highly likely to be Fake.
Recall (Fake)	89.8%	The model misses about 10% of Fake news (False Negatives).
F1-Score	91.9%	A balanced performance suitable for production deployment.

The high precision is particularly important for a fact checking tool; labeling a legitimate news source as "Fake" damages user trust significantly more than missing fake news.[27]

8.3. System Performance Under Load

Concurrency: The system successfully handled 1000 concurrent users with sub200ms response time.

Throughput: For new analyses, the system sustained ~50 requests per second on 2core cloud instance.

Bottleneck: MongoDB write operations during high concurrency voting mitigated with Redis caching.

8.4. Impact of Hybrid Scoring

Case Study: Satire. Articles from The Onion often trigger high "Fake" scores due to exaggerated language. In the live system, users consistently tagged these as "Real" (Legitimate Satire) or "Uncertain." The community score successfully moderated the final score down from 95% to ~60%.

Case Study: Concept Drift. When fed news about very recent events not in training set, ML model's confidence dropped to 50%. As early adopters flagged the news, the Community Score rose correctly.

8.5. Sensitivity Analysis of Hybrid Weighting Parameters

A key component of the proposed hybrid framework is the weighting parameter α , which controls the relative influence of the machine learning model and the community consensus in the final credibility score. To evaluate the sensitivity of the system to this parameter, a comparative analysis was conducted using three representative values: $\alpha = 0.4$, $\alpha = 0.6$, and $\alpha = 0.8$.

When α was set to 0.8, the final verdict relied predominantly on the machine learning model. While this configuration preserved strong performance on well-represented topics in the training data, it showed reduced adaptability in cases involving satire and emerging news events, where contextual understanding was required. Conversely, setting α to 0.4 increased the influence of community feedback. Although this improved responsiveness to contextual cues, it also increased vulnerability to noise during early stages when the number of community votes was low.

The intermediate value $\alpha = 0.6$ provided the most balanced performance across evaluated scenarios. This configuration maintained the stability of algorithmic predictions while allowing human judgments to meaningfully adjust outcomes in ambiguous cases. Based on these observations, $\alpha = 0.6$ was selected as the default configuration, reflecting a conservative design choice that anchors decisions to model confidence while progressively incorporating human context as participation increases.

9. DISCUSSION AND CRITICAL ANALYSIS

9.1. The "Black Box" vs. Interpretability Tradeoff

One of the central design choices of FakeCheck was the use of Logistic Regression over more complex Deep Learning models like BERT or LSTMs. While Transformer based models achieve marginally higher accuracy (94.96%), they suffer from opacity.[28]

Logistic Regression allows for extraction of the "Top N Features" with highest coefficients. Users can see: "This article was flagged because it uses the words 'BREAKING', 'SHOCKING', and 'BANNED' at frequency 5x higher than normal."

9.2. Resilience to Adversarial Attacks

A critical vulnerability of crowd sourced systems is "Brigading" or Sybil attacks. FakeCheck implements basic defenses (JWT authentication, email requirement). The mathematical model is designed to be conservative. The ML component (60% weight) acts as an anchor. Even if a bot army votes "Real" on fake articles, the ML model's "Fake" score keeps the final verdict in "Uncertain" or "Suspicious" range.[31]

Advanced approaches, such as reputation weighted voting systems, could further strengthen resilience.[30]

9.3. The Role of Community Notes

FakeCheck's community module aligns with "Community Notes" (Bird watch) research.[19][20] While individual users may be biased, a diverse aggregate often converges on truth. The key is "Contextualization"—allowing users to add context rather than just binary votes.

Research demonstrates that notes achieving consensus across ideologically diverse contributors are perceived as more trustworthy.[19]

9.4. Limitations

- **Cold Start Problem:** New articles with zero votes rely entirely on the ML model.
- **Echo Chambers:** If FakeCheck's user base becomes ideologically homogenous, "Consensus" reflects that ideology rather than objective truth.
- **Language Bias:** Current model is English centric, fails on code mixed languages (HINGLISH).
- **Timeliness:** While designed for real-time analysis, value is diminished if misinformation spreads faster than community corrections. Average latency of crowd sourced interventions is 24+ hours.[2]

10. CONCLUSION

"FakeCheck" successfully demonstrates that fighting misinformation requires a multifaceted approach. No single algorithm can define "Truth," as truth is often contextual, temporal, and nuanced. By synergizing the computational speed and pattern recognition of Logistic Regression with the contextual wisdom and adaptability of human crowd sourcing, FakeCheck provides a sustainable, scalable model for digital content verification.

The project meets its core objectives: establishing a robust automated baseline (92% accuracy), creating an intuitive user interface for real-time analysis, and implementing a novel hybrid scoring algorithm that mitigates weaknesses of purely algorithmic or purely manual systems.

The research contributes to the emerging field of Human AI collaboration by demonstrating concrete implementations of hybrid systems where human judgment is not merely supplementary to algorithmic assessment, but integral to achieving nuanced, context aware decision-making.

11. FUTURE ROADMAP

To evolve FakeCheck from a prototype into a global defense against misinformation:

11.1 **Deep Learning Integration (LLMs):** Future iterations must integrate Embeddings based classification (BERT, RoBERTa) to detect semantic inconsistencies that TFIDF misses.[28]

11.2 **Block chain Verification:** Verified news articles could be hashed and stored on block chain ledger to establish permanent "Single Source of Truth." [30]

11.3 **Browser Extension:** Automatically overlay "Traffic Light" verdict on links within social media feeds.

11.4 **Multilingual Support:** Training model on diverse datasets (Spanish, Hindi, French) is essential for global applicability.[31]

11.5 Gamification and Reputation: Weighted voting system where users with high accuracy histories gain higher voting power.[19]

11.6 Transformer Based Classifiers: Integrating state-of-the-art transformer models can achieve >95% accuracy while maintaining explain ability.[32]

In conclusion, FakeCheck serves as a testament to the potential of Human Centric AI. By keeping the human in the loop, we ensure that technology remains a tool for truth rather than an arbiter of it.

12. REFERENCES

- [1] C. Wardle and H. Derakhshan, *Information Disorder: Toward an Interdisciplinary Framework for Research and Policy Making*, Council of Europe, 2017.
- [2] S. Vosoughi, D. Roy, and S. Aral, "The spread of true and false news online," *Science*, vol. 359, no. 6380, pp. 1146–1151, 2018.
- [3] E. Bakshy, S. Messing, and L. A. Adamic, "Exposure to ideologically diverse news and opinion on Facebook," *Science*, vol. 348, no. 6239, pp. 1130–1132, 2015.
- [4] A. Bessi et al., "Homogeneous and polarized behavior in the Italian Twitter sphere," *J. R. Soc. Interface*, vol. 12, no. 110, 2015.
- [5] C. Silverman, "This analysis shows how viral fake election news outperformed real news on Facebook," *BuzzFeed News*, 2016.
- [6] G. Widmer and M. Kubat, "Learning in the presence of concept drift and hidden contexts," *Mach. Learn.*, vol. 23, no. 1, pp. 69–101, 1996.
- [7] C. R. Sunstein, "The law of group polarization," *J. Political Philosophy*, vol. 10, no. 2, pp. 175–195, 2002.
- [8] M. L. Newman, J. W. Pennebaker, D. S. Berry, and J. M. Richards, "Lying words: Predicting deception from linguistic styles," *Pers. Soc. Psychol. Bull.*, vol. 29, no. 5, pp. 665–675, 2003.
- [9] G. Demartini, S. Mizzaro, and D. Spina, "Human-in-the-loop artificial intelligence for fighting online misinformation," in *Proc. WWW*, 2020, pp. 2217–2227.
- [10] D. La Barbera, K. Roitero, and S. Mizzaro, "A hybrid human-in-the-loop framework for fact checking," in *CEUR Workshop Proc.*, vol. 3287, 2022.
- [11] A. Rössler et al., "FaceForensics++: Learning to detect manipulated facial images," in *Proc. IEEE/CVF ICCV*, 2019.
- [12] V. L. Rubin, Y. Chen, and N. K. Conroy, "Deception detection for news discourse," *First Monday*, vol. 20, no. 12, 2015.
- [13] B. D. Horne and S. Adali, "This just in: Fake news packs a lot in title," in *AAAI Workshops*, 2017.
- [14] M. Potthast et al., "A stylometric inquiry into hyperpartisan and fake news," in *Proc. ACL*, 2018, pp. 231–240.
- [15] M. Gentzkow, J. M. Shapiro, and M. Taddy, "Measuring polarization in high-dimensional data," *J. Econometrics*, vol. 208, no. 2, pp. 315–334, 2019.
- [16] K. Shu, S. Wang, and H. Liu, "Beyond news contents: The role of social context for fake news detection," in *Proc. ACM WSDM*, 2017, pp. 313–320.
- [17] R. McCulloch et al., "Social media polarization and echo chambers in COVID-19," *Nat. Commun.*, vol. 12, no. 1, 2021.
- [18] S. Feng et al., "Syntactic stylometry for deception detection," in *Proc. ACL*, 2012, pp. 171–175.
- [19] C. P. Drolsbach et al., "Community notes increase trust in fact-checking on social media," *PNAS Nexus*, vol. 3, no. 7, 2024.
- [20] K. L. Johnson et al., "From Birdwatch to Community Notes, from Twitter to X," *arXiv preprint arXiv:2401.09585*, 2024.

- [21] B. Shneiderman and C. Plaisant, *Designing the User Interface*, Pearson, 2005.
- [22] A. Rajaraman and J. D. Ullman, *Mining of Massive Datasets*, Cambridge Univ. Press, 2011.
- [23] A. Defazio, F. Bach, and S. Lacoste-Julien, “SAGA: A fast incremental gradient method,” in *Proc. ICML*, 2014.
- [24] H. Rashkin et al., “Towards a better metric for evaluating question generation systems,” in *Proc. EMNLP*, 2017.
- [25] F. Pedregosa et al., “Scikit-learn: Machine learning in Python,” *JMLR*, vol. 12, pp. 2825–2830, 2011.
- [26] H. Ahmed, I. Traore, and S. Saad, “Detection of online fake news using n-gram analysis,” in *Int. Conf. Intell., Secure, Dependable Comput.*, Springer, 2017.
- [27] S. Barocas and A. D. Selbst, “Big data’s disparate impact,” *California Law Review*, vol. 104, pp. 671–732, 2016.
- [28] J. Devlin et al., “BERT: Pre-training of deep bidirectional transformers,” arXiv:1810.04805, 2018.
- [29] A. Bifet and R. Gavaldà, “Learning from time changing data with adaptive windowing,” in *Proc. SIAM SDM*, 2007.
- [30] S. Nakamoto, “Bitcoin: A peer-to-peer electronic cash system,” White Paper, 2008.
- [31] B. Hu et al., “An overview of fake news detection,” *JMLR*, vol. 25, 2024.
- [32] B. Verma et al., “Enhancing fake news detection with transformer-based deep learning,” *PLOS ONE*, vol. 20, no. 9, 2025.

