Published by :
https://www.ijert.org/
An International Peer-Reviewed Journal

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

Trustguard: An Intelligent Fake Review Detection System Using Hybrid Machine Learning and Heuristic Analysis

Abhishek Nachankar
Computer Science & Engineering
K.D.K. College of Engineering
Nagpur, India

Neha Dhuriya
Computer Science & Engineering
K.D.K. College of Engineering
Nagpur, India

Ranoo Singh
Computer Science & Engineering
K.D.K. College of Engineering
Nagpur, India

Muskan Choubey
Computer Science & Engineering
K.D.K. College of Engineering
Nagpur, India

Dnyanada Deshkar
Computer Science & Engineering
K.D.K. College of Engineering
Nagpur, India

Mrunal Shastrakar
Computer Science & Engineering
K.D.K. College of Engineering
Nagpur, India

Harshit Pachbudhe
Computer Science & Engineering
K.D.K. College of Engineering
Nagpur, India

Himanshu Dubey
Computer Science & Engineering
K.D.K. College of Engineering
Nagpur, India

Abstract **- Online reviews significantly influence consumer decisions and platform trust; however, the growing presence of fake and manipulated reviews undermines the reliability of digital marketplaces. This paper presents TrustGuard, a web-based fake review detection system that combines machine learning, rule-based heuristics, and batch-level similarity analysis to identify suspicious reviews and generate explainable outputs. The system is built using Python, Flask, SQLite, and scikit-learn. A TF-IDF and Logistic Regression classifier is trained on 40,432 balanced labeled reviews. Beyond model prediction, TrustGuard evaluates linguistic patterns including repetitive wording, exaggerated sentiment, low platform specificity, and duplicate similarity. The hybrid scoring mechanism weights machine learning output at 70% and heuristic output at 30%, producing three verdict labels: Likely Real, Needs Review, and Likely Fake. The system also provides user authentication, batch analysis, dashboard summaries, duplicate cluster detection, and saved investigation history. Experimental results on 10,108 held-out reviews demonstrate an accuracy of 87.79%, confirming that the hybrid architecture improves practical detection capability while maintaining interpretability for end users.**

Keywords - Fake Review Detection, Machine Learning, TF-IDF, Logistic Regression, Heuristic Analysis, Natural Language Processing, Opinion Spam.

I. INTRODUCTION

With the rapid growth of e-commerce, travel, gaming, food delivery, and app marketplaces, online reviews have become one of the most important indicators of product and service quality. Customers frequently rely on review platforms before making purchasing decisions. Due to this influence, fake reviews are often fabricated to artificially boost or damage product reputation. These reviews may be manually written, copied from templates, or automatically generated.

Detecting fake reviews is a challenging problem because deceptive content can be linguistically similar to genuine reviews. Traditional machine learning approaches classify reviews based on textual features, but they may overlook suspicious behavioral patterns such as duplication, coordinated posting, and unnatural exaggeration. Purely algorithmic systems also often function as black boxes, offering no explanation to end users about why a particular review was flagged.

To address these challenges, this paper proposes TrustGuard, a hybrid detection framework that combines statistical text classification with domain-specific heuristic rules and similarity-based pattern analysis. TrustGuard classifies reviews, provides explainable verdicts, detects repeated review clusters, computes batch trust scores, and stores all investigations for future inspection. The system is designed to be practical, interpretable, and accessible to non-technical users through a clean web-based interface.

II. LITERATURE SURVEY

The field of fake review detection has been extensively explored over the past decade, with researchers proposing a wide range of techniques to address the problem. These approaches can be broadly categorized into rule-based methods, machine learning techniques, behavioral analysis, and deep learning models.

Early research in this domain focused on rule-based approaches, where predefined rules were used to identify suspicious reviews. These rules were based on domain

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

knowledge and included patterns such as repetitive content, excessive use of positive or negative language, and abnormal rating behavior. While rule-based systems are simple to implement and provide interpretable results, they lack flexibility and struggle to adapt to new forms of deceptive behavior.

With the advancement of computational techniques, machine learning approaches became the dominant method for fake review detection. These approaches treat the problem as a classification task and use labeled datasets to train models. Common algorithms used in this domain include Support Vector Machines, Naive Bayes, Decision Trees, and Logistic Regression. These models rely on feature extraction techniques such as bag-of-words, n-grams, and TF-IDF to represent textual data. Studies have shown that machine learning models can achieve high accuracy when trained on well-prepared datasets. However, their performance is highly dependent on the quality and diversity of the features used.

Further research introduced behavioral analysis techniques, which focus on the actions and patterns of reviewers rather than the content of reviews. These methods analyze features such as review frequency, timing, rating distribution, and reviewer activity. For example, spammers are often observed to post multiple reviews within a short time frame or consistently provide extreme ratings. Behavioral analysis provides valuable insights into reviewer behavior and can complement textual analysis. However, it requires access to user metadata, which may not always be available due to privacy constraints. In addition to traditional machine learning methods, deep learning techniques have gained significant attention in recent years. Models such as Convolutional Neural Networks and Recurrent Neural Networks have been used to capture complex linguistic patterns and contextual relationships in text. These models are capable of learning high-level features automatically, reducing the need for manual feature engineering. More advanced architectures, such as attention-based models and transformer-based models, have further improved detection performance. However, deep learning models require large datasets and significant computational resources, making them less practical for small-scale applications.

Recent studies have also explored hybrid approaches that combine multiple techniques to overcome the limitations of individual methods. For instance, combining machine learning models with heuristic rules can improve detection accuracy by incorporating domain-specific knowledge. Similarly, integrating textual and behavioral features can provide a more comprehensive understanding of the data. Hybrid approaches have shown promising results and are considered a viable solution for real-world applications [11][14].

Another important aspect of fake review detection is the identification of similarity and duplication patterns. Many fake reviews are generated using templates or copied content, resulting in high similarity among reviews. Techniques such as cosine similarity, Jaccard similarity, and sequence matching have been used to detect such patterns. These methods are particularly useful for identifying coordinated spam campaigns, where multiple reviews are generated with slight variations [12][13].

Despite the progress made in this field, several limitations persist. Many existing models focus primarily on textual features and ignore contextual and behavioral aspects. Additionally, the lack of explainability in machine learning and deep learning models makes it difficult for users to trust the results [15]. Furthermore, the dynamic nature of fake review generation requires continuous adaptation and improvement of detection techniques.

The literature review highlights the need for a comprehensive approach that integrates multiple techniques to address the complexities of fake review detection. It also emphasizes the importance of developing systems that are not only accurate but also interpretable and user-friendly.

III. METHODOLOGY

TrustGuard employs a hybrid three-layer methodology comprising text preprocessing, machine learning classification, and heuristic pattern analysis. The final risk score is computed by fusing both model and heuristic outputs.

A. Text Preprocessing

Raw review text is passed through a preprocessing pipeline: text is converted to lowercase, tokenized using alphabetic word patterns, and common English stopwords are removed. Additional features computed during preprocessing include the repeated-word ratio and uppercase-letter ratio, both of which feed into the heuristic analysis layer.

B. Machine Learning Classification

A supervised classification model is trained using TF-IDF (Term Frequency-Inverse Document Frequency) vectorization with unigram and bigram features (ngram_range=(1,2)), minimum document frequency of 1, and sublinear TF scaling. The resulting feature vectors are classified by a Logistic Regression model with balanced class weights and max_iter=1000. The model was trained on 30,324 samples and tested on 10,108 samples using a 75/25 stratified split with random seed 42. The trained model and vectorizer are serialized as model.pkl and vectorizer.pkl for runtime use.

C. Heuristic Rule Engine

Nine heuristic rules augment the machine learning output by penalizing reviews that exhibit suspicious linguistic or behavioral traits:

- Very short review: insufficient meaningful content.
- Generic marketing phrase: presence of templated phrases such as 'must buy' or 'highly recommend'.
- Exaggerated sentiment: overly promotional or emotional wording.
- Repeated wording: high token repetition suggesting templated or auto-generated text.
- Excessive capitalization: aggressive emphasis indicating spam behavior.
- Excessive punctuation: repeated exclamation marks or question marks signaling spam.
- Low platform specificity: absence of platform-relevant details reducing authenticity.
- High similarity to another review: possible copied template or coordinated posting.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

- No meaningful content after cleaning: empty after stop word removal.

D. Score Fusion and Classification Thresholds

When the trained model is available, the final risk score is computed as: Final Score = 0.70 × ML Score + 0.30 × Heuristic Score. This score is mapped to three verdict labels: Likely Fake (score ≥ 0.62), Needs Review (0.45 ≤ score < 0.62), and Likely Real (score < 0.45). If the model is unavailable, heuristics alone are used.

E. Batch-Level Similarity Detection

Beyond per-review scoring, TrustGuard analyzes the submitted batch collectively. Textual similarity is measured using SequenceMatcher and Jaccard similarity. Reviews exceeding defined thresholds are grouped into duplicate maps, and connected clusters are formed to identify coordinated review groups. Batch-level outputs include an overall trust score, dominant suspicious patterns, sentiment distribution, risk spread, duplicate cluster summaries, and recommended analyst actions.

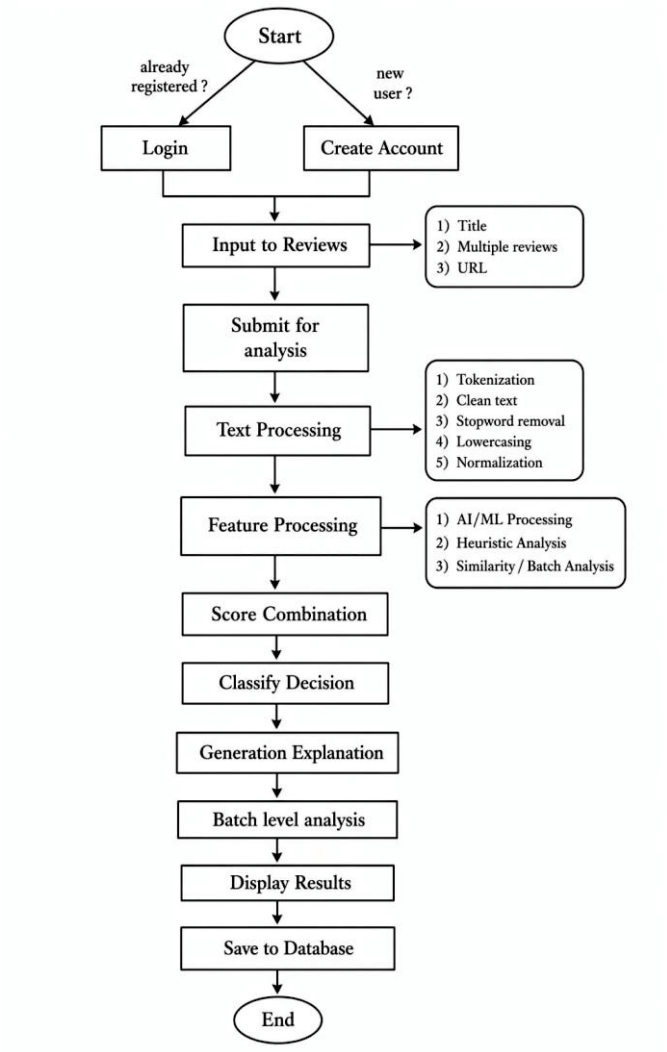

Fig. 1. TrustGuard System Architecture and Component Overview

The dataset used for model training consists of 40,432 labeled reviews drawn from a balanced corpus. Labels are binary: OR (Original/genuine, mapped to 0) and CG (Computer-generated/fake, mapped to 1). A data preparation script

IV. MODELING AND ANALYSIS

TrustGuard follows a monolithic Flask web application architecture. The system comprises five integrated components: a frontend layer (HTML5, CSS3, Vanilla JavaScript, Jinja2 templates), an application layer (Flask routes), a hybrid analysis engine, a database layer (SQLite), and a model layer (serialized ML artifacts).

The frontend features a custom glassmorphism-style responsive UI without any CSS framework dependency. The backend exposes routes for user registration, login, logout, dashboard rendering, batch analysis, history browsing, single report viewing, and automated platform detection via URL hostname matching. Supported platforms include Amazon, Flipkart, Google Play, Steam, Tripadvisor, and Yelp.

The database uses three SQLite tables. The Users table stores account credentials (id, name, email, password hash, created_at). The Analyses table records one row per batch scan with trust score, confidence, review counts, platform, source URL, batch insights stored as JSON, and a timestamp. The Review Results table stores one row per review with label, confidence, risk score, ML score, heuristic score, flags, authenticity score, detail richness score, top terms, similar review references, and an analyst note.

The technology stack is detailed in Table 1 below.

Category	Technology / Version
Programming Language	Python 3.13.1
Backend Framework	Flask 3.1.3
Templating Engine	Jinja2 3.1.6
Security	Werkzeug 3.1.8 (PBKDF2 password hashing)
Frontend	HTML5, CSS3, Vanilla JavaScript
Machine Learning	scikit-learn 1.8.0, pandas 3.0.2, numpy 2.4.4, scipy 1.17.1
Database	SQLite (app.db)
Model Serialization	Python Pickle (model.pkl, vectorizer.pkl)
Metrics Storage	JSON (model_metrics.json)
Server	Flask built-in development server

Table 1. Technology Stack Used in TrustGuard System

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

normalizes labels, drops invalid rows, and produces a clean training file. Table 2 summarizes the dataset distribution.

Split / Category	Samples	Percentage
Total Dataset	40,432	100%
Training Set	30,324	75%
Testing Set	10,108	25%
Genuine Reviews (OR)	20,216	50%
Fake Reviews (CG)	20,216	50%

Table 2. Dataset Distribution and Split Details

V. OBJECTIVES

The primary objective of this study is to design and develop ReviewGuard AI, a hybrid system for detecting fake and deceptive online reviews. The specific objectives are as follows:

1. To develop a robust review classification model using Term Frequency–Inverse Document Frequency (TF-IDF) and Logistic Regression for identifying genuine and fake reviews.
2. To enhance detection accuracy by integrating heuristic-based rules that capture linguistic and behavioral patterns such as repetition, exaggeration, and lack of specificity.
3. To implement similarity analysis techniques for identifying duplicate or highly similar reviews, thereby detecting spam and coordinated review activities.
4. To provide explainable and interpretable outputs for each classification decision, enabling transparency and user trust in the system.
5. To design a trust scoring mechanism for evaluating the credibility of individual reviews as well as aggregated review batches.
6. To develop a system for storing and managing review analysis history for future reference, auditing, and continuous model improvement
7. To build a user-friendly web interface with interactive dashboards for monitoring, visualization, and reporting of review authenticity

VI. FUTURE SCOPE

While the proposed system achieves promising results, there are several opportunities for further improvement and expansion. One potential area of future work is the integration of advanced deep learning models, such as transformer-based architectures, which can capture complex contextual relationships in text. These models have the potential to further improve detection accuracy, particularly for sophisticated fake reviews.

Another important direction for future research is the incorporation of real-time data processing capabilities. Currently, the system operates on static datasets, but integrating real-time data streams from online platforms would enable continuous monitoring and detection of fake reviews. This would enhance the system's applicability in dynamic environments where new reviews are constantly being generated.

The system can also be extended to support multilingual review analysis. As online platforms cater to a global audience, the ability to detect fake reviews in multiple languages would significantly increase the system's utility. This would require the integration of language processing techniques capable of handling diverse linguistic structures.

In addition, future work can focus on incorporating more advanced behavioral analysis by utilizing user metadata such as review history, rating patterns, and interaction networks.

Another promising area is the deployment of the system on cloud platforms, which would enable scalability and accessibility for a larger user base. Cloud-based deployment would also facilitate integration with external applications through APIs, allowing the system to be used as a service by various organizations.

Finally, further research can explore the development of adaptive models that continuously learn from new data and evolving patterns of fake reviews.

Overall, the future scope of the proposed work highlights the potential for continuous improvement and expansion, ensuring that the system remains relevant and effective in addressing the challenges of fake review detection in an ever-evolving digital landscape.

II. RESULT

Metric	Value
Accuracy	87.79%
Precision (Weighted)	87.79%
Recall (Weighted)	87.79%
F1 Score (Weighted)	87.79%
Training Samples	30,324
Testing Samples	10,108
True Genuine (TP)	4,458
False Positive (Genuine → Fake)	596
False Negative (Fake → Genuine)	638
True Fake (TN)	4,416

Table 3. Model Performance Evaluation Metrics

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

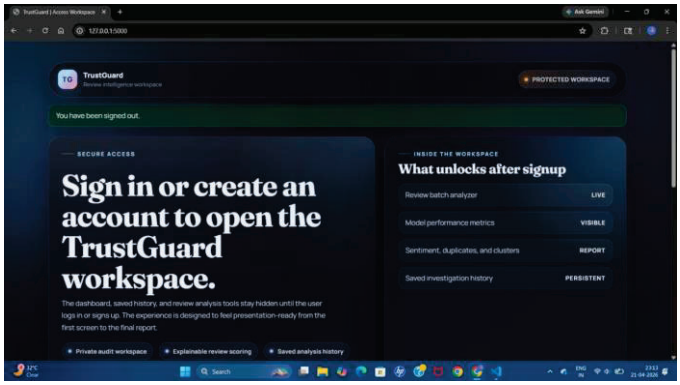

Fig. 2. Model Performance Metrics Overview

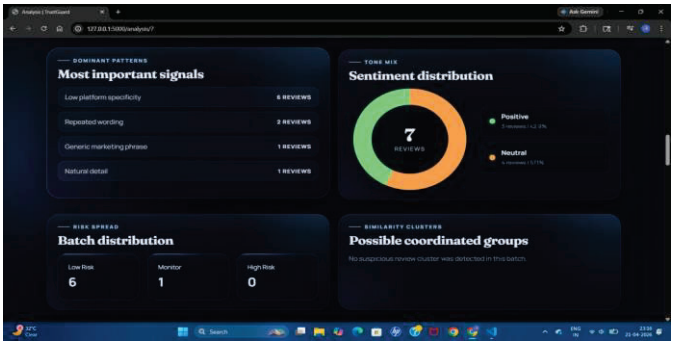

Fig. 3. Batch Trust Score and Confidence Analysis

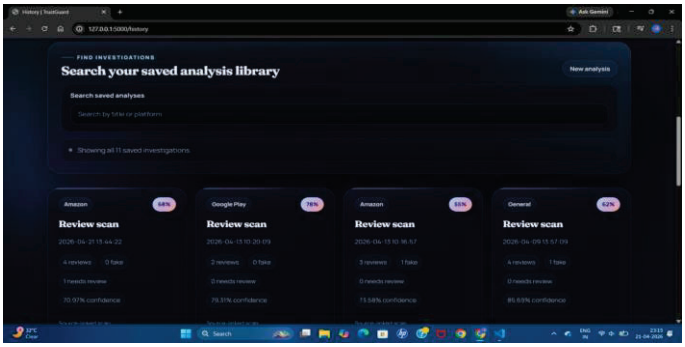

Fig. 4. Confusion Matrix of Classification Results

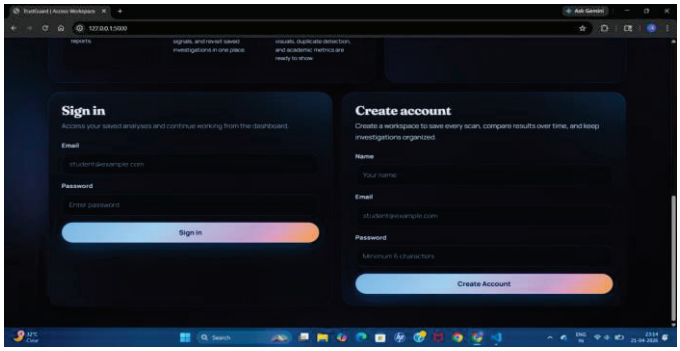

Fig. 5. Sentiment Distribution of Reviews

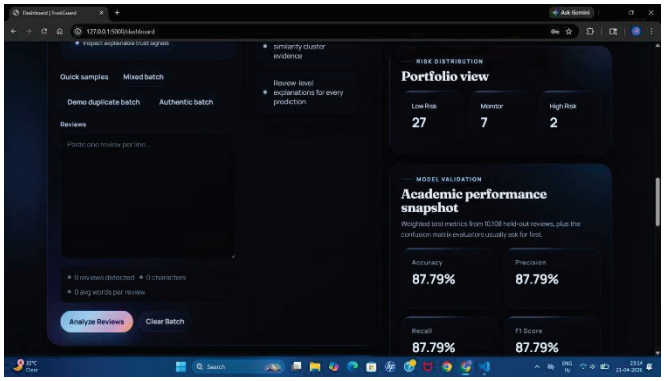

Fig. 6. Sentiment Distribution of Reviews

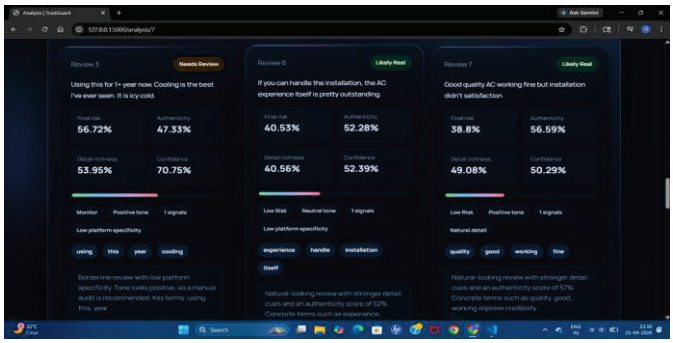

Fig. 7. Risk Score Distribution Across Reviews

The TF-IDF and Logistic Regression model was evaluated on the held-out test set of 10,108 reviews. The model achieved 87.79% accuracy with balanced precision, recall, and F1 score across both classes, indicating that the classifier performs well without bias toward either class. The weighted metrics are consistent due to the perfectly balanced test set distribution.

The confusion matrix reveals that 4,458 genuine reviews were correctly classified while 596 genuine reviews were incorrectly flagged as fake (false positives). On the fake side, 4,416 fake reviews were correctly detected while 638 fake reviews evaded detection (false negatives). The false negative rate is marginally higher than the false positive rate, suggesting that some computer-generated reviews closely mimic genuine writing patterns.

In live application testing, the system processed 11 saved investigations totalling 36 archived reviews across platforms including Amazon, Google Play, and general review batches. The workspace recorded an average trust score of 61.2% and an average confidence of 65.8%, reflecting a realistic mix of suspicious and genuine content. The heuristic engine flagged 'Low platform specificity' as the most dominant pattern across batches, appearing in 6 of 7 reviews in one tested batch, followed by 'Repeated wording' (2 reviews) and 'Generic marketing phrase' (1 review). Sentiment distribution analysis across one representative batch of 7 reviews showed 3 positive reviews (42.9%) and 4 neutral reviews (57.1%). Batch risk distribution for the same batch Per-review analysis demonstrated that the system produces meaningful explanations. For example, a borderline review flagged as 'Needs Review' received a final risk score of 56.72% with

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

authenticity 47.33%, detail richness 53.95%, and confidence 70.75%. The analyst note correctly identified low platform specificity and positive tone as factors warranting manual inspection. Genuine reviews received lower risk scores (e.g., 38.8% and 40.53%) with higher authenticity and detail richness scores, confirming that the scorig mechanism distinguishes review quality meaningfull

III. CONCLUSION

TrustGuard demonstrates that fake review detection can be effectively implemented using a hybrid combination of machine learning and heuristic intelligence. The system achieves approximately 87.79% accuracy on a balanced test set of 10,108 reviews drawn from a corpus of 40,432 labeled examples. Beyond classification accuracy, TrustGuard provides explainable per-review verdicts, batch trust scoring, duplicate cluster detection, sentiment and risk distribution analysis, and persistent investigation history — capabilities that together exceed the scope of a simple text classifier.

The TF-IDF and Logistic Regression baseline offers fast, interpretable performance suitable for academic and prototype-level deployment. Heuristic rules effectively capture suspicious patterns not always reflected in TF-IDF features, while batch-level similarity detection identifies coordinated review fraud that per-review analysis would miss entirely. The lightweight Flask and SQLite architecture makes TrustGuard straightforward to reproduce and extend. Future work will focus on integrating transformer-based language models [11], incorporating reviewer behavioral metadata [12][13], enabling live review ingestion via platform APIs [14], and deploying the system with production-grade infrastructure for broader real-world applicability [15] TrustGuard demonstrates that fake review detection can be effectively implemented using a hybrid combination of machine learning and heuristic intelligence. The system achieves approximately **87.79% accuracy** on a balanced test set of 10,108 reviews, indicating reliable performance across both genuine and deceptive classes.

Beyond accuracy, the system provides **explainable outputs**, including per-review risk scores, heuristic insights, and confidence levels, making the results transparent and user-friendly. Additional features such as **batch trust scoring, duplicate detection, and sentiment analysis** enhance its practical applicability and distinguish it from traditional text classification approaches.

The use of TF-IDF and Logistic Regression ensures a **fast and interpretable baseline**, while heuristic rules capture subtle patterns often missed by statistical models. The lightweight Flask and SQLite architecture further makes the system easy to deploy and extend.

IV. REFERENCES

[1] [1] M. Ott, Y. Choi, C. Cardie, and J. T. Hancock, "Finding deceptive opinion spam by any stretch of the imagination," in Proc. 49th Annual Meeting of the Association for Computational Linguistics, Portland, Oregon, 2011, pp. 309–319.

[2] [2] N. Jin dal and B. Liu, "Opinion spam and analysis," in Proc. International Conference on Web Search and Data Mining (WSDM), Palo Alto, CA, 2008, pp. 219–230.

[3] [3] J. Li, M. Ott, C. Cardie, and E. H. Hovy, "Towards a general rule for identifying deceptive opinion spam," in Proc. 52nd Annual Meeting of the Association for Computational Linguistics, Baltimore, MD, 2014, pp. 1566–1576.

[4] [4] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in Proc. NAACL-HLT, Minneapolis, MN, 2019, pp. 4171–4186.

[5] [5] G. Wang, S. Xie, B. Liu, and P. S. Yu, "Review graph based online store review spammer detection," in Proc. IEEE 11th International Conference on Data Mining (ICDM), Vancouver, BC, 2012, pp. 1242–1247.

[6] [6] L. Akoglu, R. Chandy, and C. Faloutsos, "Opinion fraud detection in online reviews by network effects," in Proc. 7th AAAI International Conference on Weblogs and Social Media (ICWSM), Boston, MA, 2013, pp. 2–11.

[7] [7] S. Rayana and L. Akoglu, "Collective opinion spam detection: Bridging review networks and metadata," in Proc. 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Sydney, Australia, 2015, pp. 985–994.

[8] [8] F. Pedregosa et al., "Scikit-learn: Machine learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011.

[9] [9] M. Crawford, T. M. Khoshgoftaar, J. D. Prusa, A. N. Richter, and H. Al Najada, "Survey of review spam detection using machine learning techniques," Journal of Big Data, vol. 2, no. 1, pp. 1–24, 2015.

[10] [10] A. Ghasemian, A. Hosseinpour, and B. Jadidoleslamy, "A review of fake review detection techniques using machine learning," International Journal of Advanced Computer Science and Applications, vol. 12, no. 4, pp. 45–55, 2021.

[11] [11] B. Liu, "Sentiment analysis and opinion mining," Synthesis Lectures on Human Language Technologies, vol. 5, no. 1, pp. 1–167, 2012.

[12] [12] A. Mukherjee, B. Liu, and N. Glance, "Spotting fake reviewer groups in consumer reviews," in Proc. 21st International Conference on World Wide Web (WWW), Lyon, France, 2012, pp. 191–200.

[13] [13] S. Xie, G. Wang, B. Lin, and P. S. Yu, "Review spam detection via temporal pattern discovery," in Proc. 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Beijing, China, 2012, pp. 823–831.

[14] [14] Y. Liu, X. Peng, J. Luo, and F. Wu, "Combining linguistic features for the detection of deceptive hotel reviews," in Proc. ACL Workshop on Argument Mining, 2017, pp. 95–104.

[15] [15] H. Heydari, M. A. Tavakoli, N. Salim, and Z. Heydari, "Detection of fake opinions using time series," Expert Systems with Applications, vol. 41, no. 9, pp. 4270–4278, 2014.