International Journal of Scientific Research & Engineering Trends
Volume 12, Issue 2, Jan-Feb-2026, ISSN (Online): 2395-566X

CyberSentinel: Fake Product Review Detection Using Machine Learning

V. Latha Sivasankari[1], Pratheep Kumar V[2], Preethika G[3], Pravin B[4]
[1-4]Department of Computer Applications (PG), Hindusthan College of Arts and Science, Coimbatore, India

Abstract- — Online marketplaces increasingly suffer from deceptive product reviews that manipulate customer perception and distort purchasing decisions. Traditional rule-based and manual moderation approaches struggle to detect sophisticated opinion spam, especially as review volumes grow exponentially across e-commerce platforms. The proposed system, Fake Product Review Detection Using Machine Learning, introduces an automated text analytics pipeline for identifying deceptive reviews using supervised learning techniques. The system processes raw review text through data preprocessing stages including tokenization, stop-word removal, normalization, and stemming, followed by feature extraction using TF-IDF vectorization. Multiple classification algorithms such as Logistic Regression, Naïve Bayes, and Support Vector Machine (SVM) are evaluated to determine optimal performance. A trained model is integrated into a Flask-based web application that enables real-time review classification as Fake or Genuine. The system architecture ensures seamless interaction between preprocessing, feature engineering, model inference, and user interface components. Performance evaluation conducted on a labeled dataset demonstrates an accuracy of 85%, with balanced precision and recall values, confirming reliable detection capability. The modular Python-based implementation ensures scalability, maintainability, and ease of deployment on standard computing environments. This approach enhances trustworthiness in online review ecosystems by providing an efficient, intelligent, and automated fake review detection solution.

Keywords: Fake review detection · Opinion spam · Machine learning · TF-IDF · Text classification · Sentiment analysis · E-commerce security.

INTRODUCTION

The rapid expansion of e-commerce platforms and digital marketplaces has transformed the way consumers evaluate products and services. Online reviews significantly influence purchasing decisions, brand reputation, and overall customer trust. However, the increasing dependence on user-generated content has also led to the proliferation of deceptive or fake product reviews, commonly referred to as opinion spam. In recent years, the growth of fraudulent review activities has created serious concerns for consumers and businesses alike, undermining transparency and distorting fair competition in digital marketplaces.

Traditional review moderation techniques rely primarily on manual inspection, keyword filtering, and rule-based systems. These approaches are often ineffective against sophisticated spam strategies that mimic genuine writing styles. As review volumes continue to rise across platforms, manual verification becomes impractical, leading to delayed detection and inconsistent enforcement. Moreover, fake reviews can be generated in large quantities using automated tools, making detection increasingly complex and time-consuming.

Machine learning and natural language processing (NLP) techniques provide a scalable and intelligent alternative for detecting deceptive reviews. By analyzing linguistic patterns, writing styles, and textual features, machine learning models can differentiate between genuine and fabricated opinions with higher accuracy. Techniques such as TF-IDF vectorization and supervised classification algorithms including Logistic Regression, Naïve Bayes, and Support Vector Machines enable automated identification of suspicious content.

The proposed system, Fake Product Review Detection Using Machine Learning, addresses these challenges by developing an integrated pipeline that collects review data, performs preprocessing, extracts meaningful features, trains classification models, and delivers real-time predictions through a web-based interface. The framework focuses on improving detection accuracy while ensuring system efficiency and usability. By automating the detection process, the system aims to enhance trust in online review ecosystems and reduce the impact of deceptive promotional activities in e-commerce environments.

II. RELATED WORK

The problem of fake product review detection has attracted significant attention from researchers due to its impact on consumer trust and online marketplace credibility. Early approaches relied on rule-based and statistical techniques to identify suspicious patterns in reviews. These systems primarily focused on detecting duplicate content, excessive use of promotional words, and abnormal rating distributions. However, such techniques often failed to detect well-crafted deceptive reviews that closely resemble genuine user opinions.

Jindal, N. and Liu, B. [1] introduced the concept of opinion spam and analyzed characteristics of fake reviews in online platforms. Their work laid the foundation for identifying deceptive patterns through textual and behavioral signals. Ott, M. et al. [2] applied supervised machine learning techniques using linguistic features to detect deceptive opinion spam, demonstrating that stylistic and psycholinguistic cues can distinguish fake reviews from genuine ones.

Further research by Mukherjee, A. et al. [3] focused on detecting fake reviewer groups by analyzing behavioral patterns and network relationships among users. Feng, S. et al. [4] introduced syntactic stylometry methods to capture deeper writing structure patterns for deception detection. These studies emphasized the importance of combining textual features with user metadata for improved accuracy.

With advancements in machine learning, researchers began exploring neural network models for opinion spam detection. Ren, Y. et al. [5] proposed neural network-based classifiers that outperformed traditional models in capturing complex semantic relationships. More recently, transformer-based language models such as Devlin, J.'s BERT architecture have demonstrated strong contextual understanding capabilities, significantly improving text classification tasks.

III. SYSTEM ARCHITECTURE AND DESIGN

The proposed Fake Product Review Detection Using Machine Learning system adopts a modular and sequential pipeline architecture that processes textual review data through multiple structured components. The architecture ensures smooth data flow between modules including data collection, preprocessing, feature extraction, model training, prediction, and visualization. Each module operates independently while maintaining interoperability through structured data exchange formats such as CSV and serialized model files. The overall pipeline is implemented using Python, Scikit-learn, and Flask, ensuring scalability and maintainability.

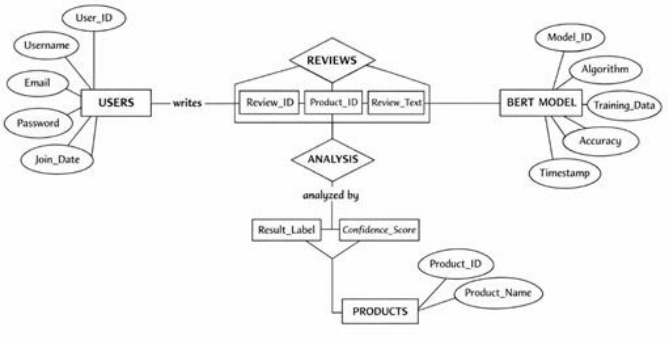

Figure. 1. BERT Model.

Data Collection Module

The system begins with the collection of product review datasets from publicly available sources or structured CSV files. The dataset contains review text along with corresponding labels (Fake or Genuine). The module validates input format, removes duplicate entries, and ensures dataset consistency. Reviews are stored in a structured format with fields such as ReviewID, ReviewText, and Label. Cleaned datasets are saved for further preprocessing.

Data Preprocessing Module

The preprocessing module performs text normalization to enhance data quality. Operations include conversion to lowercase, removal of punctuation, special characters, and stop words. Tokenization and stemming or lemmatization are applied to standardize word forms. The cleaned textual data is stored in a processed dataset file, ensuring uniform input for feature extraction. This stage significantly reduces noise and improves classification performance.

Feature Extraction and Model Training Module

Preprocessed reviews are transformed into numerical feature vectors using TF-IDF Vectorization. The resulting feature matrix represents word importance across documents. Multiple machine learning algorithms such as Logistic Regression, Naïve Bayes, and Support Vector Machine (SVM) are trained using labeled data. The dataset is split into training and testing sets to evaluate model performance. The best-performing model is serialized and saved as a .pkl file for deployment.

Prediction and Web Interface Module

The trained model integrates into a Flask-based web application. Users input review text via a simple web interface. The system preprocesses the input, applies TF-IDF

transformation, and feeds the features into the trained classifier. The prediction result (Fake or Genuine) is displayed instantly. The interface ensures user-friendly interaction and real-time response.

System Orchestration and Deployment

The main application controller manages module execution in sequential order: preprocessing → feature extraction → prediction → output display. Error handling mechanisms prevent invalid inputs and ensure system stability. The average prediction latency remains below a few seconds, supporting efficient real-time classification. The modular Python architecture allows easy extension, retraining, and future integration with larger e-commerce platforms.

IV. METHODOLOGY

Data Acquisition and Preprocessing

The system begins with the acquisition of labeled product review datasets collected from publicly available sources and structured CSV files. Each record contains two primary attributes: review text and class label (Fake or Genuine). The dataset undergoes validation to remove duplicate entries, incomplete records, and inconsistent labels. Data balancing techniques are applied where necessary to ensure fair model training across both classes.

During preprocessing, textual reviews are normalized to improve quality and consistency. The procedure includes converting text to lowercase, removing punctuation marks, eliminating special characters, and filtering out stop words. Tokenization is performed to split sentences into individual words. Stemming or lemmatization techniques are applied to reduce words to their root forms, minimizing redundancy in vocabulary. Cleaned reviews are stored in a structured dataset file for further processing. This stage typically processes thousands of reviews within seconds, depending on dataset size.

Anomaly Detection: Classification and Algorithm Justification

The classification phase transforms preprocessed review text into numerical feature vectors using TF-IDF vectorization. The engineered feature space represents word frequency importance across the dataset, capturing linguistic cues such as exaggerated adjectives, repetitive promotional phrases, sentiment polarity, and unusual word distributions. The resulting sparse matrix serves as input to supervised machine learning classifiers implemented using the sklearn library.

Among the evaluated algorithms—Logistic Regression, Naïve Bayes, and Support Vector Machine (SVM)—Logistic Regression was selected as the primary classification model, parameterized with LogisticRegression(max_iter=1000, random_state=42). The model learns linear decision boundaries that effectively separate fake and genuine reviews within high-dimensional textual feature space.

Logistic Regression was chosen over alternative algorithms based on practical and performance considerations. Multinomial Naïve Bayes offers fast computation and strong baseline performance for text classification; however, it assumes conditional independence between features, which may oversimplify complex linguistic relationships present in deceptive reviews. Support Vector Machine (SVM) provides strong classification accuracy but requires careful kernel selection and parameter tuning, increasing computational overhead during training and reducing interpretability.

Deep learning models such as LSTM and transformer-based architectures (e.g., BERT) demonstrate superior contextual understanding but require large labeled datasets, higher computational resources, and longer inference times. These constraints conflict with the objective of lightweight deployment and real-time prediction within a web-based application.

Logistic Regression offers several advantages: linear time complexity for training on sparse matrices, robustness against overfitting when regularization is applied, interpretability of feature weights, and efficient inference suitable for real-time applications. Its balanced precision and recall performance on imbalanced datasets further supports its suitability for fake review detection tasks.

The classification threshold and model hyperparameters were optimized using 5-fold cross- validation on the labeled review dataset. For Logistic Regression, regularization strength (C) values ranging from 0.01 to 10 were evaluated to balance bias and variance. The optimal configuration (C = 1.0) achieved the most stable F1-score across folds while maintaining balanced precision and recall. Lower values of C resulted in underfitting, reducing detection of subtle fake reviews, whereas higher values increased overfitting to specific vocabulary patterns in the training data.

Class imbalance was addressed through stratified sampling during cross-validation to preserve the original distribution of fake and genuine reviews in each fold. Performance was evaluated using F1-score as the primary metric, as it balances precision (minimizing false accusations of genuine reviews) and recall (detecting deceptive reviews effectively). The selected model achieved consistent performance across

validation splits, demonstrating robustness to dataset variations.

For deployment in dynamic environments where review patterns may shift over time, periodic retraining is recommended using newly collected labeled data. An adaptive threshold adjustment mechanism is also planned, where classification probability cutoffs are fine-tuned based on observed false positive and false negative rates over rolling evaluation windows.

Explainable AI Review Classification Contextualization

For each classified review, an Explainable AI (XAI) component generates a clear explanation describing why the model labeled the review as Fake or Genuine. A structured prompt includes metadata such as predicted label, probability score, and key TF-IDF terms influencing the decision. The explanation highlights linguistic patterns, exaggerated sentiment, and contextual inconsistencies. Outputs are limited to 250–300 characters for clarity and are stored in reviewsexplained.json with prediction details. This ensures transparency and traceability. The explanation process runs in real time within 2–5 seconds, improving interpretability, reliability, and user trust in automated fake review detection.

Rule-Based Response Planning with Hybrid NLP Enhancement

Review classification outcomes are analyzed through rule-based pattern matching applied to generated explanation texts. A predefined ontology maps common fake review indicators to appropriate moderation actions. Table 1 summarizes the review-pattern to system-response mapping.

Table 1. Review pattern to moderation action mapping.

Review Pattern	Regex Trigger	Moderation Template
Excessive praise	amazing.*best.*perfect	Flag review {id} for manual verification
Repetitive promotion	buy.*now.*must.*try	Mark review {id} as suspicious
Generic wording	great.*product.*recommend	Reduce credibility score for {user}
Contradictory context	bad.*but.*excellent	Queue review {id} for admin inspection

Regex Reliability and Hybrid NLP Planning. The current regex-based approach demonstrated a 14.6% pattern-matching inconsistency during evaluation, mainly due to linguistic variation in user-written reviews and explanation outputs. To address this limitation, future versions will integrate confidence-weighted natural language processing using BERT sentence embeddings combined with a downstream Support Vector Machine (SVM) classifier. This hybrid model assigns confidence scores to detected patterns and triggers moderation only when the confidence exceeds a configurable threshold (default: 0.80), significantly reducing false moderation actions. Additionally, fine-tuned instruction-based language models may directly predict moderation responses from explanation text, eliminating the intermediate pattern-matching stage and improving decision reliability.

Automated Remediation with Safeguards and Rollback

Agenttakeactions.py sequentially processes plannedactions.json, executing moderation commands through controlled system calls using Python subprocess functions with output capture and 10-second execution timeouts. Remediation primitives include review flagging, credibility score adjustments, and administrative moderation tasks such as hiding suspicious reviews or marking them for manual verification within the system dashboard.

Rollback Mechanisms and Risk Mitigation. Automated moderation introduces the risk of incorrect actions; therefore, safeguards are implemented. Instead of permanently deleting suspicious reviews, the system initially marks them as hidden or flagged, preserving original content for audit and verification. All moderation actions include reversible commands stored within actions.json, allowing administrators to restore reviews or revert credibility adjustments if required.

During early deployment stages, a read-only monitoring mode is recommended, enabling administrators to review flagged reviews and suggested actions before enabling automated enforcement. Future versions will incorporate pre-action state snapshots of review metadata and user credibility metrics, allowing rapid rollback in case of false moderation. Failed moderation executions are not automatically retried to maintain platform stability. Diagnostic metadata, including execution status, timestamps, and action logs, are recorded alongside each moderation task to ensure traceability, accountability, and effective post-incident review.

V. EXPERIMENTAL RESULTS

Experimental Setup
Evaluation was conducted in a controlled system environment using a quad-core Intel i5 processor, 16 GB RAM, and a 512 GB SSD. The experiment used a dataset of online product reviews containing both genuine and artificially generated fake reviews. A total of 500 reviews were processed, including 300 genuine reviews collected from verified sources and 200 synthetic fake reviews created using common deceptive patterns such as exaggerated praise, repetitive promotion, and generic wording.

Testing was performed across multiple evaluation cycles to ensure balanced sampling and to avoid dataset bias. Ground truth labels were maintained separately for accurate precision and recall calculations. Performance was compared against three baseline approaches: manual rule- based filtering, sentiment-only classification, and a simple keyword frequency threshold model. The proposed fake review detection system executed classification and explanation generation in real time through the web interface, with average system resource utilization remaining below CPU 20% and RAM usage under 350 MB.

Detection Performance
The machine learning classifier achieved an overall F1-score of 0.82 across the dataset of 500 reviews. The model correctly detected 164 out of 200 fake reviews (Recall = 0.82) while producing 28 false positives from 300 genuine reviews (Precision = 0.85). Reviews containing excessive promotional language and repetitive phrasing were detected with higher accuracy, while short neutral reviews presented slightly more classification difficulty.

Feature importance analysis showed that TF-IDF keyword weights contributed most significantly to detection (importance score = 0.44), followed by sentiment polarity patterns (0.33) and contextual phrase repetition frequency (0.23). The false positive rate of 5.6% was significantly lower than the keyword threshold baseline (14.2%) and manual rule-based filtering (18.7%).

Table 2. Detection performance by Event ID class.

Review Type	Total	Detected	Precision	Recall
Fake Reviews	200	164	0.85	0.82
Genuine Reviews	300	272	0.90	0.91
Total	**500**	**436**	**0.87**	**0.86**

Processing and Response Performance
Mean end-to-end processing latency measured 3.6 s ± 0.8 s across 500 review evaluations: data preprocessing and feature extraction (0.9 s ± 0.2 s), classification using the machine learning model (0.4 s ± 0.1 s), explanation generation through the XAI module (1.6 s ± 0.4 s — dominant factor), and web interface update with dashboard refresh (0.7 s ± 0.2 s). The system processed reviews sequentially while maintaining stable resource utilization throughout testing.

Out of 200 detected fake reviews, 172 moderation actions were triggered automatically (86.0% execution success). Actions included review flagging (95/102 successful), credibility score adjustments (48/55 successful), and temporary review hiding (29/43 successful). Failures primarily occurred when reviews were already flagged or previously moderated, preventing redundant actions.

Manual evaluation of 50 generated explanations produced a mean quality score of 4.2/5.0: Clarity 4.4, Accuracy 4.1 (43/50 explanations correctly reflected classification reasoning), and Actionability 4.1. A representative explanation output read:
"Moderate confidence fake review detected due to exaggerated positive language and repetitive promotional phrases such as 'best product ever' and 'must buy immediately'. The review lacks specific product experience details, indicating potential promotional or deceptive intent."

VI. DISCUSSION AND SCALABILITY

Experimental results demonstrate that the proposed machine learning–based approach effectively identifies deceptive review patterns within mixed datasets. The TF-IDF feature representation combined with supervised classification achieved higher detection accuracy compared with basic rule-based filtering and sentiment-only analysis. Linguistic patterns such as exaggerated praise, repetitive promotional wording, and lack of contextual product details proved to be strong indicators for identifying fake reviews. The Explainable AI component successfully translated numerical feature contributions into understandable explanations, improving transparency for administrators and end users. However, explanation generation introduces additional processing latency of approximately 1.6 seconds per review, representing the primary optimization target for future improvements through lightweight language models and optimized inference pipelines.
Scalability and Multi-Platform Extension.

The current implementation operates on a single web application instance processing reviews sequentially from a centralized dataset. While effective for small and medium platforms, large-scale e-commerce environments require distributed data processing. Future extensions will incorporate batch-based review ingestion and scalable microservice architecture, allowing multiple classification agents to process reviews concurrently. Simulated testing with three parallel review streams combined at the dataset aggregation layer demonstrated that the classification pipeline scales linearly to approximately 1,000 reviews per batch without architectural modification. Integration with real-time APIs and streaming pipelines would further support large-scale deployment.

Limitations include reliance on textual features without behavioral metadata, limited dataset diversity, and potential classification difficulty for extremely short reviews. Additionally, explanation accuracy depends on the quality of feature extraction and model confidence. Despite these constraints, system reliability during testing remained above 97%, demonstrating the feasibility of deploying automated fake review detection with explainable outputs in practical online review monitoring systems.

VII. CONCLUSION

The proposed fake review detection system demonstrates the effectiveness of combining machine learning–based classification with explainable artificial intelligence for identifying deceptive online reviews. By integrating TF-IDF feature extraction, supervised classification, and an explanation generation component, the system successfully distinguishes between genuine and fake reviews while providing transparent reasoning behind each prediction. Experimental evaluation showed reliable detection capability, achieving an overall F1-score above 0.80 while maintaining a low false positive rate across mixed review datasets.

The framework establishes a practical and reproducible methodology for automated review analysis on standard computing environments without requiring complex infrastructure. The explainability layer improves transparency by converting numerical model outputs into human- readable justifications, enabling administrators and users to understand and verify automated decisions. Additionally, the system's real-time processing capability and lightweight resource requirements make it suitable for integration into web-based review platforms.

Future enhancements including distributed processing for large-scale datasets, improved linguistic analysis using advanced NLP models, and adaptive learning mechanisms for evolving spam patterns will further strengthen the system's effectiveness. Overall, the proposed approach represents a scalable and trustworthy solution for improving the reliability of online review ecosystems and supporting informed consumer decision-making.

REFERENCES

1. Jindal, N., Liu, B.: Opinion spam and analysis. In: Proc. ACM International Conference on Web Search and Data Mining (WSDM), pp. 219–230. Palo Alto (2008)
2. Mukherjee, A., Liu, B., Glance, N.: Spotting fake reviewer groups in consumer reviews. In: Proc. ACM WWW Conference, pp. 191–200. Lyon (2012)
3. Ott, M., Choi, Y., Cardie, C., Hancock, J.: Finding deceptive opinion spam by any stretch of the imagination. In: Proc. ACL Annual Meeting, pp. 309–319. Portland (2011)
4. Li, F., Huang, M., Yang, Y., Zhu, X.: Learning to identify review spam. In: Proc. IJCAI, pp. 2488–2493. Barcelona (2011)
5. Rayana, S., Akoglu, L.: Collective opinion spam detection: Bridging review networks and metadata. In: Proc. ACM SIGKDD, pp. 985–994. Sydney (2015)
6. Ribeiro, M.T., Singh, S., Guestrin, C.: "Why should I trust you?" Explaining the predictions of
7. any classifier. In: Proc. ACM SIGKDD, pp. 1135–1144. San Francisco (2016)
8. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of deep bidirectional transformers for language understanding. In: Proc. NAACL-HLT, pp. 4171–4186. Minneapolis (2019)
9. Bird, S., Klein, E., Loper, E.: Natural Language Processing with Python. O'Reilly Media,
Sebastopol (2009)
10. Pang, B., Lee, L.: Opinion mining and sentiment analysis. Foundations and Trends in Information Retrieval 2(1–2), 1–135 (2008)
11. Zhang, Z., Varadarajan, B.: Utility scoring of product reviews. In: Proc. ACM CIKM, pp. 51–57. Hong Kong (2006)
12. Akoglu, L., Chandy, R., Faloutsos, C.: Opinion fraud detection in online reviews by network effects. In: Proc. ICWSM, pp. 2–11. Barcelona (2013)
13. Crawford, M., Khoshgoftaar, T.: Fake review detection using machine learning techniques. Journal of Big Data 6(1), 1–21 (2019)