Detection of AI-Generated and Deceptive Fake Reviews on E-Commerce Platforms Using Lightweight Machine Learning

Arpita Singh
Student Researcher
Dr. D. Y. Patil Institute of MCA and Management
Akurdi, Pune
arpita22032004@gmail.com

Mahima Wankhade
Student Researcher
Dr. D. Y. Patil Institute of MCA and Management
Akurdi, Pune
mahimawankhade257@gmail.com

Sanyogita Dahale
Student Researcher
Dr. D. Y. Patil Institute of MCA and Management
Akurdi, Pune
sanyogitadahale@gmail.com

Snehal Patil
Research Project Guide
Dr. D. Y. Patil Institute of MCA and Management
Akurdi, Pune
snehal.patil@dypimca.ac.in

***Abstract*—Product reviews shape what people buy, yet their credibility is under a fresh threat: large language models (LLMs) can now mass-produce reviews that are fluent, varied and grammatically flawless. Current detectors cope with this only in part. The best-performing published systems lean on transformer models, paid LLM interfaces or reviewer metadata, resources that smaller platforms often lack, whereas lighter classical models were largely designed around older, human-written spam. We compare five recent studies of fake-review detection, looking at their data, labelling, methods, reported accuracy and acknowledged limitations. Two patterns emerge: accuracy tends to rise together with computational cost, and some of the highest accuracies rest on heuristic or weakly supervised labels, which casts doubt on what they actually measure. Drawing on these observations, we propose a lightweight, interpretable pipeline that pairs TF-IDF n-gram features with stylometric indicators of machine-generated text and compares four classifiers: Naïve Bayes, Logistic Regression, Support Vector Machine and Random Forest. The accompanying evaluation protocol is fully specified and includes stratified hold-out and cross-validation, cross-generator and cross-domain tests, paraphrase robustness and ablation. Experimental results for the proposed pipeline are not included in this version, and no performance figures are claimed for it. The system is intended as a risk-scoring aid for human moderators, never as proof of AI authorship.**

Keywords*—AI-generated reviews, fake review detection, opinion spam, large language models, TF-IDF, stylometry, Random Forest, e-commerce trust.

I. Introduction

A. Background Information

When shoppers cannot handle a product themselves, they turn to star ratings and written reviews, and sellers in turn rely on those same reviews to be seen and to sell. That mutual dependence is exactly what makes manipulation attractive. Work on Amazon data has uncovered an organised market in which sellers pay for invented reviews that lift their ratings and search rankings [1]. Research on opinion spam goes back some way: duplicated text and unusual reviewer behaviour were early giveaways of fake reviews [2], and deceptive reviews gathered through crowd-sourcing could be told apart from truthful ones by their word choice and psychological features [3].

Generative AI has altered the picture considerably. With nothing more than a general-purpose chatbot, a seller can generate hundreds of unique, polished, product-specific reviews within minutes. No text needs to be copied, and the typos and over-the-top language that older detectors learned to watch for are gone. These can fairly be called “deepfake reviews”: text written wholly by a machine, on behalf of someone who never used the product, and meant to be believed. Research on separating machine text from human text indicates that statistical and stylistic differences do persist [4][5][6]. The same research cautions that a classifier trained in one domain frequently loses accuracy when moved to another [5].

In response, a growing number of studies apply machine learning and deep learning to the question of review authenticity, with reported accuracies running from the high eighties to above ninety-eight percent [7][8][9][10][11]. Comparing these figures is hard, since the studies use different data, obtain their labels in different ways and assume different computing budgets. This paper begins from two questions: what do these numbers really mean, and can a practical detector be built without heavy infrastructure?

B. Research Problem, Questions and Objectives

LLMs are increasingly used to write fluent, human-sounding fake reviews. Detection systems that exist today fall into two camps. Some depend on computationally costly transformer or LLM-API pipelines; others use classical models that were never designed to pick up the linguistic traces of machine-generated text. What is missing is a lightweight, interpretable machine-learning framework that can label a review as human-written or likely AI-generated using ordinary computing resources. Four research questions guide the study:

- RQ1. Which approaches have recent studies taken to detect fake and AI-generated reviews, and what accuracy and costs do they report?
- RQ2. How far can the reported results be trusted, given differences in how datasets were built and how labels were obtained?
- RQ3. Can a lightweight pipeline based on TF-IDF n-grams and stylometric features yield useful, explainable signals for spotting AI-generated reviews?

- RQ4. How should such a pipeline be evaluated so that the results are credible, reproducible and robust to unfamiliar generators and domains?

The objectives that follow from these questions are fourfold: to synthesise the literature through a structured comparison, to pinpoint a specific research gap, to design a lightweight detection pipeline with interpretable outputs, and to define an evaluation protocol that sidesteps the validity problems seen in earlier work.

C. Significance and Scope of the Research

Smaller e-commerce platforms, consumer-protection groups and student researchers seldom have GPU clusters at hand, nor can they pay for an LLM service on every review. A detector that runs on a laptop-class machine and can explain itself would put review-integrity tools within wider reach. The work also relates to the Sustainable Development Goals for resilient digital infrastructure (SDG 9) and responsible consumption (SDG 12), because trustworthy product information is a precondition for informed buying. The scope is intentionally narrow. The paper analyses published studies, proposes a pipeline and specifies how it should be evaluated. Classifier output is treated as a risk signal requiring human review, and no claim is made that text analysis can establish who or what wrote a review.

II. Literature Review

A. Overview of Relevant Literature

a) Machine-generated text detection

Liao et al. examined ChatGPT-generated and human-written medical text. Although the generated text showed strong linguistic competence, machine-learning classifiers could still detect statistical and stylistic differences [4]. Alamleh et al. arrived at much the same conclusion using conventional classifiers with carefully designed features, but they observed weak performance across domains [5]. Katib et al. set support vector machines, decision trees, CNNs and LSTMs against one another on the same task; hybrid architectures helped, though outcomes depended on the dataset and on which features were selected [6]. Other approaches are more statistical. GLTR shows how predictable each word is under a language model [12], and DetectGPT exploits the curvature of a model's probability surface to detect text without supervision [13]. Such methods help us understand machine text, but they were built for general writing rather than short product reviews with ratings, and they require a language model to be available at detection time.

b) Fake-review detection with classical machine learning

Pandi and Siva Kumar trained KNN, Logistic Regression and SVM on a Kaggle product-review dataset, drawing on sentiment, n-gram and metadata features, and reported 95%, 94% and 92% accuracy respectively [10]. Because the paper itself says the dataset is not explicitly labelled as fake, it is difficult to judge what the reported accuracy represents. A further inconsistency deserves mention: the abstract gives an F1-score of 92.4% for the SVM, while the results table lists 0.96. Even so, the study shows that inexpensive models can reach high scores on a particular dataset.

c) Deep learning approaches

Khanna et al. compared Random Forest, Logistic Regression, CNN and LSTM on 20,000 real and 20,000 fake reviews, feeding the models TF-IDF and word embeddings. The accuracies were 85.00%, 93.94%, 91.00% and 92.19%, with LSTM performing best [8]. Since the corpus is framed in terms of computer-generated (CG) and original (OR) reviews, it comes closest to the AI-generated setting of all the studies reviewed. The authors note that deep models demand large datasets and considerable computing power, and that they generalise poorly across datasets.

d) Hybrid text-and-behaviour systems

Sajja et al. paired a fine-tuned DistilBERT model with a Logistic Regression model working on behavioural metadata, combining their outputs through soft voting. They used 40,433 Amazon Home & Kitchen reviews, of which 8,087 were held out for testing. The hybrid reached 88.71% accuracy and F1-score, compared with 89.42% for TextCNN, 88.28% for SVM, 87.86% for Logistic Regression and 84.91% for Random Forest, and the authors point out that the approach relies on behavioural metadata [9]. DistilBERT is a smaller, distilled form of BERT [14], yet fine-tuning it still takes more resources than any classical model.

Bankadamani et al. combined fine-tuned DistilBERT embeddings with a LightGBM model operating on rating deviation, review length, posting frequency and helpful votes, weighting the two components 0.60 and 0.40. On 100,000 Amazon Fashion reviews they reported 92.29% accuracy, 93.35% precision, 92.14% recall and 92.74% F1 [11]. Their comparison table also lists a DeBERTa-based model, a fuzzy RoBERTa model and a RoBERTa-LSTM model developed by other authors [15][16]. A crucial detail is that the dataset carried no ground-truth labels, so the authors produced them with a heuristic. Five-star reviews under 35 words, or reviews with many exclamation marks and promotional phrases, were marked fake; reviews with at least one helpful vote, or with more than 40 words and a rating above three stars, were marked real [11].

Rao et al. proposed a multi-factor framework that fuses AI-content analysis (through an LLM API), linguistic patterns, reviewer behaviour and temporal bursts, and outputs a credibility score with two-level flags. Their dataset mixed public reviews with LLM-generated, duplicated, template-based and time-manipulated ones. On it they reported 88% accuracy, 0.86 precision, 0.84 recall and 0.85 F1, ahead of the text-only (78%), behaviour-only (81%) and AI-only (84%) variants. Because of API latency, processing about 100 reviews took two to three minutes [7]. Earlier work on reviewer behaviour had likewise found that behaviour carries signal for detecting fake reviews [17].

e) Dataset construction for AI-generated reviews

Salminen et al. described how fake product reviews can be both created and detected, which is directly useful for anyone assembling a labelled set of generated reviews [18]. The CG/OR labelling used by Khanna et al. points to a similar construction, although the paper does not fully document where the data came from [8].

TABLE I. COMPARISON OF THE FIVE DIRECTLY RELEVANT STUDIES

Ref.	Study	Data and method	Reported accuracy	Main limitation
[7]	Rao et al. (2026)	Mixed public and synthetic data; semantic, linguistic, behavioural and temporal fusion with LLM API	88%	Needs reviewer and time data; API latency and cost; synthetic augmentation
[8]	Khanna et al. (2025)	20,000 real + 20,000 fake (CG/OR); RF, LR, CNN, LSTM with TF-IDF and embeddings	89.19% (LSTM)	Heavy compute for deep models; generalisation across domains
[9]	Sajja et al. (2026)	40,433 Amazon Home & Kitchen; DistilBERT + LR soft voting	88.71%	Depends on metadata; one product category
[10]	Pandi and Siva Kumar (2025)	Kaggle data; KNN, LR, SVM with sentiment, n-gram and metadata features	90% (SVM)	Labelling of fake reviews not explicit; inconsistent F1 reporting
[11]	Bankadamani et al. (2026)	100,000 Amazon Fashion; DistilBERT + LightGBM ensemble	92.29%	Heuristic weak labels; fashion only; resource-heavy

B. Key Concepts

Opinion spam and AI-generated reviews. Opinion spam covers any review written to mislead readers about a product. AI-generated reviews form a subset whose text was produced by a language model. Their sentiment may well be honest; they remain deceptive because no real experience stands behind them.

Lexical and stylometric features. TF-IDF gives a word more weight the more often it occurs in a review and the less common it is across the corpus. Stylometric features, by contrast, capture how a text is written instead of which words it contains: variation in sentence length, density of adjectives and reliance on generic praise are typical examples. Machine-generated text tends toward uniformity and even-handed sentiment, whereas human writing is more irregular and tied to personal experience [4][5].

Weak supervision and label validity. Where ground-truth labels are unavailable, researchers sometimes derive them from rules. If the signals used to define the labels also serve as model inputs, the model can score highly simply by learning the rule instead of the behaviour it is meant to capture. Such label leakage is central to interpreting results in this field.

C. Summary of Key Findings from the Literature

- Reported accuracy runs from 88% to 92.29%, but data, labels and resources vary so much that the figures cannot be ranked against one another [7][8][9][10][11].
- The two best-scoring text-and-behaviour systems depend on transformer models, an LLM API or reviewer metadata [7][9][11].
- Just two of the five studies deal explicitly with AI-generated text: Rao et al., through an LLM-API pipeline [7], and Khanna et al., through computer-generated and original reviews handled by deep models [8]. The remainder address fake reviews in general.
- The highest reported accuracy was obtained with heuristic labels built from features that the model can also see [11].
- Detectors of machine-generated text can lose accuracy across domains [5] and may be biased against non-native English writers [19].

III. RESEARCH GAP

Comparing the studies exposes four gaps that this work sets out to address.

1. **High accuracy comes with an infrastructure bill.** The most accurate systems call for transformer fine-tuning, a paid LLM API or behavioural metadata. In the experiments of Sajja et al., Random Forest needed 133.33 s to train, against 20.46 s for their hybrid and less than a second for Logistic Regression and SVM [9]; Rao et al. report that API calls dominate their latency [7]. Smaller platforms and student projects cannot always afford such costs.

2. **Lightweight systems rarely target AI-generated text.** Studies built on classical models were aimed at fake reviews in general [10]. Those that do tackle AI-generated text rely either on an LLM API [7] or on deep models trained on CG/OR data [8], leaving a low-cost option unexplored.

3. **Label validity is uncertain.** One study had no explicit fake labels [10]. Another built its labels from rating, length, exclamation marks and helpful votes [11], signals that overlap with its own model inputs. In such settings, reported accuracy may overstate performance in the real world.

4. **Generalisation and robustness are seldom tested.** The reviewed studies mostly report one hold-out split within a single domain. None examines an unfamiliar generator, a new product domain or paraphrased text, even though machine-text detection is known to be sensitive to these shifts [5].

Taken together, the gaps call for a detector that is inexpensive, designed for AI-generated text, trained on labels of known provenance, and tested for both generalisation and fairness.

IV. PROPOSED METHODOLOGY AND SYSTEM ARCHITECTURE

A. Research Design

The study has two parts. The first is the comparative analysis in Section II, which assesses each study against fixed criteria: data source, label provenance, features, model, reported metrics and stated limitations. The second is an experimental design for a lightweight classifier, whose pipeline appears in Fig. 1 and whose stages are described below. The experimental part is specified in enough detail to be reproduced; its results are left for the next stage of the project.

B. Data Collection

Data come from two sources. **Human reviews** are drawn from public e-commerce corpora, for example Kaggle product-review datasets and the Amazon and Yelp review collections. **AI-generated reviews** are created under a controlled protocol. For every sampled human review, an LLM receives the same product category, star rating and approximate length and is asked to write a new review, so the two classes are matched on these attributes. Several prompt styles (neutral, enthusiastic, critical) and more than one LLM are used, and each record stores the generator name, prompt and settings. This metadata makes possible the cross-generator test described in Section V. Every generated review is labelled at the moment of creation, which means no label is ever inferred from the text. The aim is

a balanced corpus, and a second, imbalanced version (about 90% human) is prepared to mirror realistic prevalence.

Genuine reviews cannot be guaranteed free of deception. They are therefore taken from verified-purchase records where these exist, and the assumption that they are human-written is declared as a limitation. Duplicate, empty and irrelevant reviews are removed before splitting, and near-duplicates are also removed across classes to prevent leakage.

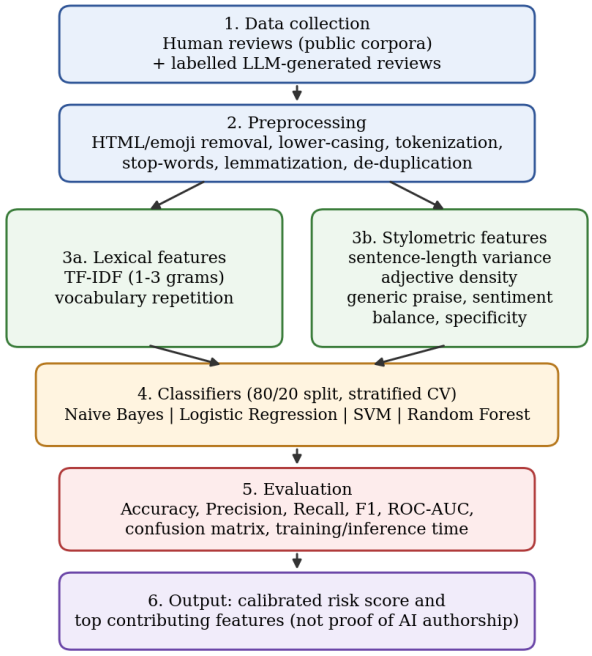


Fig. 1. Proposed lightweight pipeline for detecting AI-generated and deceptive reviews

C. Text Preprocessing

Reviews go through lower-casing, removal of HTML tags, URLs, emojis and special characters, tokenization, stop-word removal and lemmatization. Two versions of the text are kept. The fully normalised version feeds the lexical features. The other is only minimally cleaned and retains punctuation, capitalisation and stop-words, because stylometric cues such as exclamation marks and sentence boundaries would otherwise disappear. Both classes are preprocessed in exactly the same way.

D. Feature Extraction

Two families of features are extracted and then concatenated.

Lexical features. TF-IDF vectors are built over unigrams, bigrams and trigrams, with the vocabulary capped (for example at 50,000 terms) and a minimum document frequency set to discard rare tokens. The vectoriser is fitted on the training split alone. For a term t in document d, the TF-IDF weight is TF(t, d) × log(N / DF(t)), where N is the number of documents and DF(t) is the number of documents containing t.

Stylometric features. These express the signals identified in the project proposal and supported by work on machine text [4][5][6]; Table II lists them.

TABLE II. STYLOMETRIC FEATURE GROUPS

Feature group	Measures	Expected signal
Sentence regularity	Mean, standard deviation and coefficient of variation of sentence length	Uniform lengths suggest generated text
Lexical diversity	Type-token ratio, repeated n-gram rate, vocabulary richness	Repetitive phrasing
Adjective density	Share of adjectives and intensifiers per token (POS tagging)	Over-descriptive wording
Generic praise	Count of stock phrases such as “highly recommend” and “great product”	Template-like praise
Sentiment balance	Polarity score and ratio of positive to negative terms	Oddly balanced tone
Specificity	Share of product-specific nouns, numbers and named entities	Low specificity suggests no real use
Readability	Flesch reading ease, average word length, punctuation rate	Over-polished text

The main experiments leave out metadata such as rating and reviewer history, so that the detector relies on text alone and remains usable on platforms that withhold metadata. Rating serves only to match the two classes during generation. An optional extension adds metadata, as earlier hybrid systems did [9][11].

E. Classification Models

Four lightweight classifiers are compared (Table III). They were chosen because they train quickly, run on standard hardware and, in the case of the linear models and the forest, allow us to inspect what drives a decision.

TABLE III. CLASSIFIERS AND HYPER-PARAMETER SEARCH

Model	Role	Search grid (validation)
Multinomial Naïve Bayes	Fast probabilistic baseline for word counts	Smoothing $\alpha \in \{0.1, 0.5, 1.0\}$
Logistic Regression	Interpretable linear model; coefficients show influential terms	$C \in \{0.1, 1, 10\}$, L2 penalty
Linear SVM [20]	Maximum-margin boundary in high-dimensional space	$C \in \{0.1, 1, 10\}$
Random Forest [21] (primary)	Ensemble of trees; handles mixed lexical and stylometric features	Trees $\in \{100, 300\}$; depth $\in$ {None, 30}

Every model is implemented in scikit-learn [22]. To put the cost-accuracy trade-off in context, one transformer baseline (fine-tuned DistilBERT [14]) is run on identical splits wherever hardware allows. It forms no part of the proposed lightweight system; its job is to show how much accuracy, if any, is sacrificed in return for lower cost.

F. Output and Interpretability

Predicted probabilities are calibrated on the validation split (Platt scaling or isotonic regression) and converted into a risk score between 0 and 1. Each review then falls into one of three bands: likely human, needs human review, or likely AI-generated, with thresholds fixed on the validation split to reach a stated precision target. Explanations come from logistic coefficients, forest feature importance, permutation importance and, for single reviews, SHAP values [23]. The three bands deliberately reflect a limitation: text analysis cannot conclusively prove that a machine wrote a review, so uncertain cases are passed to people.

V. Experimental Setup and Implementation

A. Environment

The implementation is written in Python. It uses pandas for data handling, NLTK or spaCy for tokenization, lemmatization and part-of-speech tagging, and scikit-learn for vectorisation, models and metrics [22]. All experiments run on a standard CPU-only machine, and random seeds, library versions and generation prompts are logged so the experiments can be repeated.

B. Data Splits

Following the proposed methodology, 80% of the corpus is used for training and 20% for testing, stratified by class and by product category. Inside the training portion, hyper-parameters are tuned with five-fold stratified cross-validation (or a held-out validation fraction for the transformer baseline). Splitting takes place after de-duplication and, where several reviews concern the same product, is done by product group, so that near-identical text never lands on both sides. The test set is touched only once, after every design choice is fixed. Beyond the standard split, three shifted test sets are built: a cross-generator set of reviews from an LLM absent from training; a cross-domain set from a product category absent from training; and a paraphrase set in which generated test reviews are rewritten by another model or lightly edited.

C. Evaluation Metrics

Let TP, FP, TN and FN stand for true positives (AI-generated reviews correctly flagged), false positives (genuine reviews wrongly flagged), true negatives and false negatives. The following metrics are computed:

- Accuracy = (TP + TN) / (TP + TN + FP + FN): the overall share of correct decisions.
- Precision = TP / (TP + FP): the proportion of flagged reviews that really are AI-generated.
- Recall = TP / (TP + FN): the proportion of AI-generated reviews that are found.
- F1-score = 2 × Precision × Recall / (Precision + Recall): the balance between the two.
- ROC-AUC and PR-AUC: threshold-independent measures, with PR-AUC essential for the imbalanced version.
- Confusion matrix: the complete pattern of errors, reported for each model and each shifted test set.
- Cost: training time, inference time per 1,000 reviews and memory use.

False positives harm genuine reviewers, while false negatives let deception pass, so precision and recall are always reported side by side. Results are expressed as mean and standard deviation over cross-validation folds, with 95% bootstrap confidence intervals on the test set and McNemar's test for paired comparisons between classifiers.

D. Planned Experiments

Table IV. Planned Experiments

ID	Experiment	Purpose	Primary outcome
E1	Four classifiers on TF-IDF only, standard split	Baseline comparison (RQ3)	F1, accuracy, AUC
E2	TF-IDF + stylometric features	Added value of style features	Change in F1
E3	Ablation by stylometric group	Which cues matter	Drop in F1 per group
E4	Cross-generator and cross-domain tests	Generalisation (RQ4)	Drop in recall and F1
E5	Paraphrased and edited AI text	Robustness to evasion	Recall on paraphrased set
E6	Transformer reference and cost comparison	Cost-accuracy trade-off	F1 per unit time
E7	Error analysis by review length and writer style	Fairness and failure modes	FP rate by subgroup

VI. Results and Discussion

The only results available at this stage are the published performances of the reviewed systems, and this section reports and interprets those. The experiments in Table IV have not yet been run, so no accuracy, precision, recall or F1 value is claimed for the proposed pipeline. What follows is a critical reading of existing evidence, together with explicit expectations for the pipeline and the way each will be tested.

A. Reported Performance of Existing Systems

Fig. 2 and Table I present the accuracy reported by the five studies, which ranges from 88% [7] to 92.29% [11]. Among transformer-based systems, Sajja et al. report 88.71% [9] and Bankadamani et al. 92.29% [11]. Among those using classical or recurrent models, Pandi and Siva Kumar report up to 90% with SVM [10] and Khanna et al. 89.19% with LSTM [8]. One study compares classifiers on the same split in detail, and there the order from best to worst is the hybrid transformer model, TextCNN, SVM, Logistic Regression and Random Forest [9]. In the CG/OR study the order is LSTM, Logistic Regression, CNN and Random Forest [8]. Random Forest comes last in both, which matters here because it is this project's primary model; Section VI-D returns to the point.

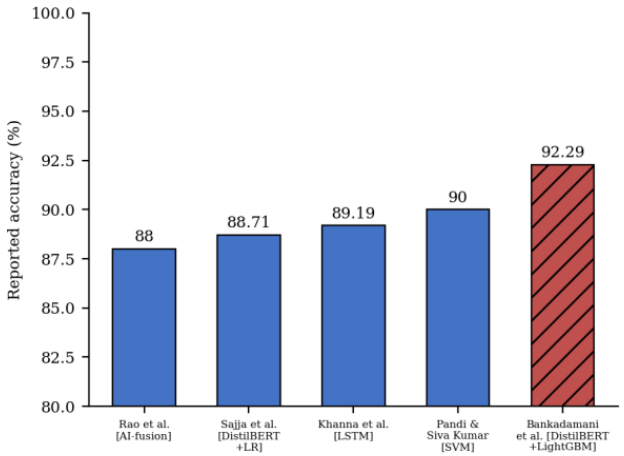


Fig. 2. Accuracy reported by the five reviewed studies (vertical axis starts at 80%). The hatched bar uses heuristic weak labels. Values are not directly comparable across studies.

B. Reading the Numbers Critically

Cost. The highest figures come with the heaviest requirements. Sajja et al. report a wide gap in training time: the hybrid (20.46 s), TextCNN (32.67 s) and Random Forest (133.33 s) sit on one side, Logistic Regression and SVM (about 0.4 s) on the other, and yet the linear models still reach 87.86%

and 88.28% accuracy [9]. Rao et al. need two to three minutes per 100 reviews because of LLM calls [7]. Accuracy thus rises with cost, though not in proportion, and the last few points are the expensive ones.

Label validity. Bankadamani et al. obtained their 92.29% figure on labels built from rating, review length, exclamation marks, promotional phrases and helpful votes, while the model inputs include rating deviation, review length and helpful votes [11]. Under these conditions a high score may simply mean that the model has reconstructed the labelling rule. The figure therefore cannot be taken as an ability to find deceptive reviews written by real fraudsters or by LLMs, and the authors themselves name human annotation in place of heuristic labels as future work [11]. In the same way, Pandi and Siva Kumar describe their dataset as not explicitly labelled as fake [10], so their 90% is best read as agreement with the dataset's class column and not as proof of fraud detection.

Unrealistic difficulty. Rao et al. augmented public reviews with synthetic duplicates, template spam and manipulated timestamps [7]. Injected fraud of this kind is easier to catch than fraud in the wild, and a single dataset cannot show how a system performs elsewhere. Their comparison with single-factor variants is nonetheless informative: combining the evidence sources lifted accuracy from 78–84% to 88%, which is consistent with the view that fusion helps.

Error structure. The reviewed papers offer little confusion-matrix evidence, but one finding is useful. Sajja et al. show that TextCNN achieved high precision (0.9362) and low recall (0.8459), meaning it seldom raised false alarms but missed many fakes. The hybrid, in contrast, reached nearly equal precision (0.9303) and recall (0.9312) [9]. Accuracy alone would conceal this difference, which is why the present protocol always reports precision, recall and the full confusion matrix, and why a risk band for uncertain reviews is built into the design.

C. Handling Sophisticated AI-Generated Text

Modern LLM output is grammatical and coherent, so surface errors no longer give it away. What remains are statistical regularities: even sentence lengths, evenly balanced sentiment, stock praise and a shortage of concrete detail from real use [4][5]. The proposed stylometric features are aimed at exactly these properties. Three weaknesses are anticipated, each tied to an experiment. A generator prompted to mimic casual, irregular writing, or text that has been paraphrased, may wipe out these cues (E5). Features learned from one LLM may also fail to transfer to another (E4), as the cross-domain losses reported for machine-text classifiers suggest [5]. And a fraudster who has actually seen the product can add concrete detail, so specificity is a weak signal on its own. For these reasons the pipeline outputs a risk band, and the plan includes subgroup analysis (E7).

Fairness is a further concern. Detectors of machine-written text have been found to misclassify the writing of non-native English speakers as machine-generated more often than native writing [19]. Because plain, uniform phrasing is also one of this project's features, false-positive rates will be reported by review length and by proxy measures of writing style, and flagged reviews must never be removed automatically.

D. Expected Behaviour of the Proposed Pipeline

The expectations below are hypotheses to be tested, not findings. (H1) On the standard split, linear models with TF-IDF should prove strong baselines, since the linear models in [9] and [8] performed well at low cost. (H2) Stylometric features should help most on short reviews, where TF-IDF has few words to work with. (H3) Random Forest ranked last in [9] and [8] on sparse text vectors, but it may improve when given dense stylometric features and a reduced set of top-ranked n-grams; this remains to be shown. (H4) Performance should fall on cross-generator and paraphrase tests by more than it does on the standard split, and the size of that fall is the most informative number for practitioners. (H5) The transformer reference will probably recover more paraphrased text, leaving open whether the gain justifies its cost. Any of H1 to H5 that the data do not support will be reported as such.

E. Implications and Limitations

Implications. Researchers should state how their labels were produced and avoid feeding the model the very signals that define them. Platforms should treat classifier output as a prompt for review rather than a verdict. Evaluation should cover shifts in generator, domain and wording, because a single hold-out split overstates what a detector can do in a changing environment.

Limitations.

- The proposed pipeline has not yet been experimentally evaluated, so its benefit remains a hypothesis.
- The comparison covers five studies, was carried out by the authors from the published texts, and does not re-analyse the original data. Some studies give limited detail, for example on the dataset and hyper-parameters in [10].
- Reported figures are taken as published, come from different data and labels, and are not ranked against each other here.
- Genuine reviews are assumed to be human-written, and the generated reviews reflect the particular LLMs and prompts used.
- Text-only detection cannot identify a human who copies or edits AI text, or an AI that imitates informal writing.

VII. Conclusion and Future Scope

A. Conclusion

Examining recent machine-learning and deep-learning studies of fake and AI-generated review detection revealed three consistent patterns. Accuracy improves along with computing cost and metadata requirements; the most striking accuracy figures depend on label construction that may favour the model; and few studies test generalisation to new generators, domains or paraphrased text. In response, the paper proposed a lightweight, interpretable pipeline that combines TF-IDF n-grams with stylometric features, compares Naïve Bayes, Logistic Regression, SVM and Random Forest, and outputs a calibrated risk score. An evaluation protocol designed to avoid the identified validity problems accompanies it. The pipeline is a screening aid for human moderators and is not proof that a review was machine-written.

B. Contributions

- A structured comparison of five recent detection studies, including corrected and consistent reporting of their published figures and stated limitations.
- A research gap defined along four dimensions: cost, AI-specific design, label validity and robustness.
- A lightweight detection pipeline using lexical and stylometric features, with interpretable outputs and a three-band risk design.
- A reproducible evaluation plan (E1 to E7) covering classifier comparison, ablation, cross-generator and cross-domain tests, paraphrase robustness, cost and fairness.

C. Future Scope

- **Run and report the experiments.** Carry out E1 to E7, publish the confusion matrices and confidence intervals, and state which of H1 to H5 hold.
- **Human-verified labels.** Add a manually annotated evaluation set so that performance is not tied to generated or heuristic labels.
- **Embeddings and transformers.** Test word embeddings, LSTM and fine-tuned DistilBERT or RoBERTa as optional higher-accuracy tiers, including distillation into the lightweight model.
- **Behavioural and temporal signals.** Where a platform shares them, combine text with reviewer behaviour and posting bursts, as earlier multi-factor work has done [7][17].
- **Adversarial and multilingual evaluation.** Study adaptive adversaries who paraphrase detected text, and extend the work to other languages and to non-text signals.
- **Deployment.** Build a moderator dashboard with explanations, and study how analysts use the risk bands.

ACKNOWLEDGMENT

The authors thank their guide, Mrs. Snehal Patil, and Dr. D. Y. Patil Institute of MCA and Management, Akurdi, Pune, for guidance and support.

REFERENCES

[1] S. He, B. Hollenbeck, and D. Proserpio, "The market for fake reviews," Marketing Sci., vol. 41, no. 5, pp. 896–921, 2022.
[2] N. Jindal and B. Liu, "Opinion spam and analysis," in Proc. Int. Conf. Web Search Web Data Mining (WSDM), 2008, pp. 219–230.
[3] M. Ott, Y. Choi, C. Cardie, and J. T. Hancock, "Finding deceptive opinion spam by any stretch of the imagination," in Proc. 49th Annu. Meeting Assoc. Comput. Linguistics (ACL-HLT), 2011, pp. 309–319.
[4] W. Liao et al., "Differentiate ChatGPT-generated and human-written medical texts," arXiv:2304.11567, 2023.
[5] H. Alamleh, A. A. S. AlQahtani, and A. ElSaid, "Distinguishing human-written and ChatGPT-generated text using machine learning," in Proc. Syst. Inf. Eng. Design Symp. (SIEDS), 2023, pp. 154–158.
[6] I. Katib, F. Y. Assiri, H. A. Abdushkour, D. Hamed, and M. Ragab, "Differentiating Chat Generative Pretrained Transformer from humans: Detecting ChatGPT-generated text and human text using machine learning," Mathematics, vol. 11, no. 15, Art. no. 3400, 2023.
[7] C. U. Rao, Y. L. Lahari, V. Nitin, K. Shashank, B. Mahesh, and B. Anil, "A multi-factor AI-driven framework for detecting fake and AI-generated online reviews using semantic, behavioral, and temporal analysis," in Proc. 6th Int. Conf. Expert Clouds Appl. (ICOECA), 2026, pp. 1563–1570, doi: 10.1109/ICOECA68095.2026.11485495.
[8] V. Khanna M A, K. Ramalakshmi, D. P. P. Latha, R. Venkatesan, and A. C. Wanika R, "An efficient approach for fake review detection with machine and deep learning techniques," in Proc. 7th Int. Conf. Innov. Data Commun. Technol. Appl. (ICIDCA), 2025, pp. 1012–1016, doi: 10.1109/ICIDCA66325.2025.11280411.
[9] S. L. Sajja, R. Pranavi, T. N. Amulya, S. S. Priya, Y. Aparna, and S. Madhumitha, "Enhanced fake review detection in e-commerce: A comparative study of deep learning and hybrid ensemble classifiers," in Proc. 4th Int. Conf. Augmented Intell. Sustain. Syst. (ICAISS), 2026, pp. 1589–1593, doi: 10.1109/ICAISS68683.2026.11525550.
[10] T. Pasupathi Pandi and N. Siva Kumar, "Fake review detection in e-commerce using machine learning and NLP technique," in Proc. 3rd Int. Conf. Inventive Comput. Informatics (ICICI), 2025, pp. 692–698, doi: 10.1109/ICICI65870.2025.11069636.
[11] L. Bankadamani, V. R. Honnagudi, B. Gudi, S. Bhagoji, C. Muttal, and G. Konnurmath, "Fake review detection in fashion e-commerce using machine learning," in Proc. IEEE Int. Conf. Convergence Comput. Technol. (I3CTCON), Lonavala, India, 2026, doi: 10.1109/I3CTCON68242.2026.11507156.
[12] S. Gehrmann, H. Strobelt, and A. M. Rush, "GLTR: Statistical detection and visualization of generated text," arXiv:1906.04043, 2019.
[13] E. Mitchell, Y. Lee, A. Khazatsky, C. D. Manning, and C. Finn, "DetectGPT: Zero-shot machine-generated text detection using probability curvature," arXiv:2301.11305, 2023.
[14] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter," arXiv:1910.01108, 2019.
[15] S. Geetha, E. Elakiya, R. Sujithra Kanmani, and M. K. Das, "High performance fake review detection using pretrained DeBERTa optimized with monarch butterfly paradigm," Sci. Rep., Mar. 2025.
[16] R. Mohawesh, H. B. Salameh, Y. Jararweh, M. Alkhalaileh, and S. Maqsood, "Fake review detection using transformer-based enhanced LSTM and RoBERTa," Array, 2024.
[17] D. Zhang, L. Zhou, J. L. Kehoe, and I. Y. Kilic, "What online reviewer behaviors really matter? Effects of verbal and nonverbal behaviors on detection of fake online reviews," MIS Quart., 2016.
[18] J. Salminen, C. Kandpal, A. M. Kamel, S. Jung, and B. J. Jansen, "Creating and detecting fake reviews of online products," Decis. Support Syst., 2022.
[19] W. Liang, M. Yuksekgonul, Y. Mao, E. Wu, and J. Zou, "GPT detectors are biased against non-native English writers," Patterns, vol. 4, no. 7, Art. no. 100779, 2023.
[20] C. Cortes and V. Vapnik, "Support-vector networks," Mach. Learn., vol. 20, no. 3, pp. 273–297, 1995.
[21] L. Breiman, "Random forests," Mach. Learn., vol. 45, no. 1, pp. 5–32, 2001.
[22] F. Pedregosa et al., "Scikit-learn: Machine learning in Python," J. Mach. Learn. Res., vol. 12, pp. 2825–2830, 2011.
[23] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in Adv. Neural Inf. Process. Syst. (NeurIPS), vol. 30, 2017.