IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 10, October 2026

DOI: 10.17148/IJARCCE.2026.151085

Beyond Contextual Embeddings: A Comparative Analysis of Feature-Engineered Classical Machine Learning and Transformer-Based Approaches for Deceptive Opinion Spam Detection

Shaliny P[1], Dr.S.Niveditha[2]

M.Tech 2nd Year, Department of Computer Science and Engineering, SRMIST, Vadapalani[1]

Assistant Professor, Department of Computer Science and Engineering, SRMIST, Vadapalani[2]

Abstract: We compare classical machine learning with a published fine-tuned BERT result on the Deceptive Opinion Spam Corpus (OpSpam), a public benchmark of 1,600 hotel reviews. Naive Bayes, logistic regression and linear SVM classifiers were trained on TF-IDF features under stratified 5-fold cross-validation, the protocol used by the BERT baseline we compare against. The strongest configuration joins word n-grams (1 to 3) with character n-grams (3 to 5). It reaches 91.12% accuracy on a fixed seed and 90.78% averaged over ten repeated runs, against 89.1% reported for BERT, and it trains in about two seconds per fold on a CPU. The margin needs careful reading. A plain unigram and bigram SVM already scores 89.38%, the fusion improves on the better of its two feature views by under one point, and its edge over the plain SVM is borderline (corrected resampled t-test, $p = 0.053$). When folds are grouped by hotel, so that test hotels never appear in training, every n-gram variant falls to about 89%, level with the BERT figure. We also audit the highest accuracy we found for this corpus, 96.03% for a RoBERTa and LSTM hybrid. It comes from an 80/20 split without cross-validation or variance estimates, and its precision, recall and accuracy are hard to reconcile with a single balanced test set. Under the BERT paper's own protocol the n-gram fusion finishes 1.7 to 2.0 points ahead, but the lead disappears when hotels are held out.

Index Terms: BERT, character n-grams, cross-validation, deceptive opinion spam, fake review detection, OpSpam, support vector machine, TF-IDF.

I. INTRODUCTION

A. Background

Review sites run on text written by strangers, which gives sellers a reason to pay for flattering reviews of their own products and hostile ones about a rival's. Automatic detection of such reviews has been studied since at least Jindal and Liu [1], and a survey of the field is available [2]. Comparing results across papers is hard because datasets, labels and splits differ. One corpus is the exception. The Deceptive Opinion Spam Corpus (OpSpam) holds 1,600 reviews of twenty Chicago hotels: 800 deceptive reviews written by paid crowd workers and 800 truthful ones collected from review sites [3], [4]. It is public, balanced, and small enough that anyone can rerun a published method in minutes.

B. Evolution of Methods

The first wave of work on OpSpam used hand-built features. Ott et al. [3] fed n-grams and psycholinguistic categories from LIWC to an SVM. Feng et al. combined bag-of-words with constituency tree fragments and report 91.2% [5], [6]. Li et al. brought in a topic model [7]. Neural models followed: recurrent convolutional networks [8], GRU and CNN hybrids [9], and the DRI-RCNN [10]. Recent papers fine-tune transformers such as BERT [11]. Examples include a compact distilled BERT [12], a BERT trained with a sentiment-aware multi-label loss [13], a social-media variant called DeceptiveBERT [14], and a RoBERTa and LSTM hybrid [15]. Character n-grams appear in this history too, usually as one input among several [16], [17].

C. Problem Statement

A tacit assumption runs through this history: more pretraining should mean better detection. Kennedy et al. [6] tested it on OpSpam under a stated protocol, with stratified 5-fold cross-validation and bootstrap validation, and their fine-tuned BERT beat their own logistic regression and SVM by 2.7 to 3.5 points. We ask whether that gap survives better feature engineering on the classical side. To keep the comparison fair we reuse their protocol, stratified 5-fold cross-validation

 This work is licensed under a Creative Commons Attribution 4.0 International License

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 10, October 2026

DOI: 10.17148/IJARCCE.2026.151085

on the full corpus, and we state how every number was obtained, because accuracy from a single split, from K-fold and from the bootstrap do not measure quite the same thing [18].

D. Contributions

This paper contributes four things. (i) A reproduction of classical baselines on OpSpam under the 5-fold protocol of the BERT paper. (ii) A word plus character TF-IDF fusion with a linear SVM, which scores 91.12% on a fixed seed and 90.78% over ten repeated runs, against 89.1% for BERT [6]. (iii) Two checks we did not find in the OpSpam papers we read: a paired significance test against our own SVM baseline, and a split that holds out whole hotels. (iv) An audit of the 96.03% RoBERTa and LSTM result [15], which was obtained without cross-validation.

II. RELATED WORK

TABLE I. SELECTED LITERATURE ON DECEPTIVE REVIEW DETECTION

Paper / year and core technique	Key contribution / limitation
Cagnina & Rosso [16] (2015) — SVM using character n-grams, emotion and LIWC features	Shows stylistic cues can help detect deception; feature combination and older evaluation limit direct comparison with transformers.
Fusilier et al. [17] (2015) — Character n-grams with PU-learning and Naive Bayes	Studies positive and negative deceptive opinions; not a direct transformer-era benchmark.
Catelli et al. [13] (2022) — BERT jointly modeling deception and sentiment polarity	Uses sentiment as an auxiliary signal; requires additional annotation and preprocessing.
Paper / year and core technique	**Key contribution / limitation**
Hyder et al. [14] (2024) — Domain-adapted DeceptiveBERT for social-media text	Targets noisy informal text; transfer to curated hotel reviews remains unverified.
Mohawesh et al. [15] (2024) — RoBERTa + LSTM with SHAP explanations	Reports 96.03% accuracy, but uses an 80/20 split without cross-validation, limiting comparability.

Table II lists accuracy figures reported on OpSpam in the papers we examined. Eleven of its rows were compiled by Mohawesh et al. [15, Table 4], who set the reference methods, from SVMs with LIWC features to SOM-CNN, against their own model. We reproduce their numbers as given and do not vouch for how each was obtained. Their three SVM rows follow [3], [5], [16], though the paper does not say which row belongs to which source.

TABLE II
REPORTED ACCURACY AND F1 ON OPSPAM, AS COMPILED IN [15, TABLE 4]

Method	Accuracy (%)	F1 (%)
SVM, bigram + LIWC	82.89	82.09
SVM, unigram	86.09	83.12
SVM, LIWC + 4-gram	84.34	83.21
SAGE [7]	83.10	82.23
RCNN [8]	83.21	81.23
GRNN-CNN [9]	84.15	84.17
DRI-RCNN [10]	87.24	85.36
BERT base [12]	90.31	89.56

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 10, October 2026

DOI: 10.17148/IJARCCE.2026.151085

LDA + TextCNN [19]	90.39	90.90
MFNN [20]	88.72	89.10
SOM-CNN [21]	89.23	90.01
RoBERTa + LSTM [15] †	96.03	95.36

† 80/20 split, no cross-validation.

Two features of this record matter for what follows. Eleven of the twelve rows lie between 82.9% and 90.4%, whatever the architecture, which suggests a corpus that is easy to learn but too small to let any model show much more. And the numbers come from different protocols. The paper behind the 96.03% row describes an 80/20 split [15], the BERT paper reports both 5-fold and bootstrap results [6], and the 91.2% of Feng et al. [5] predates the negative-polarity half of the corpus [4], so it cannot have been measured on all 1,600 reviews.

Character n-grams are an old tool in stylometry [22] and have been tried on opinion spam. Cagnina and Rosso combined them with emotion features and LIWC in an SVM, and Fusilier et al. used character n-gram features with PU-learning and Naive Bayes, as summarized in [15]. Both predate fine-tuned transformers, so neither could compare against one. That head-to-head comparison is what we run.

III. DATASET

We used a public CSV export of OpSpam v1.4 with five fields: deceptive label, hotel, polarity, source and review text. The labels split evenly, 800 deceptive and 800 truthful, and so does polarity, 800 positive and 800 negative. All 800 deceptive reviews come from Amazon Mechanical Turk. The truthful side has 400 positive reviews labelled TripAdvisor and 400 negative reviews labelled Web in the export. There are twenty hotels and no missing values. Length barely separates the classes (mean 146.6 words for deceptive reviews, 150.9 for truthful) but separates polarities sharply: negative reviews average about 178 words and positive ones 116 to 123. The truthful reviews were mined from the web and are not gold standard; they are only believed to have a low deception rate [6].

IV. METHODS

A. Baselines

Three classical baselines mirror the non-neural models of [6]: multinomial Naive Bayes on unigram TF-IDF, logistic regression on unigram and bigram TF-IDF, and a linear SVM on unigram and bigram TF-IDF. The baselines use raw term frequency. Kennedy et al. say only that words were represented in TF-IDF form, so their exact settings are unknown to us.

B. Word and Character Fusion

The proposed model builds two TF-IDF views of each review. The word view uses 1 to 3 grams with minimum document frequency 2. The character view uses 3 to 5 grams cut at word boundaries, so no gram spans two words, with minimum document frequency 3. Both views use sublinear term frequency, so the weight of term t in document d is

$$w(t, d) = [1 + \ln tf(t, d)] \cdot idf(t) \quad (1) \qquad idf(t) = \ln[(1 + n) / (1 + df(t))] + 1 \quad (2)$$

and each document vector is L2-normalized, as implemented in scikit-learn [23]. The word and character vectors of a review are concatenated,

$$x = [x_w ; x_c] \quad (3)$$

and a linear SVM [24] with squared hinge loss is fit on the result by minimizing

$$\min_{w,b} \tfrac{1}{2}\|w\|^2 + C \sum_i \max(0, 1 - y_i(w \cdot x_i + b))^2 \quad (4)$$

with C = 1. Fig. 1 shows the pipeline. Vectorizers are fit on the training fold only.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 10, October 2026

DOI: 10.17148/IJARCCE.2026.151085

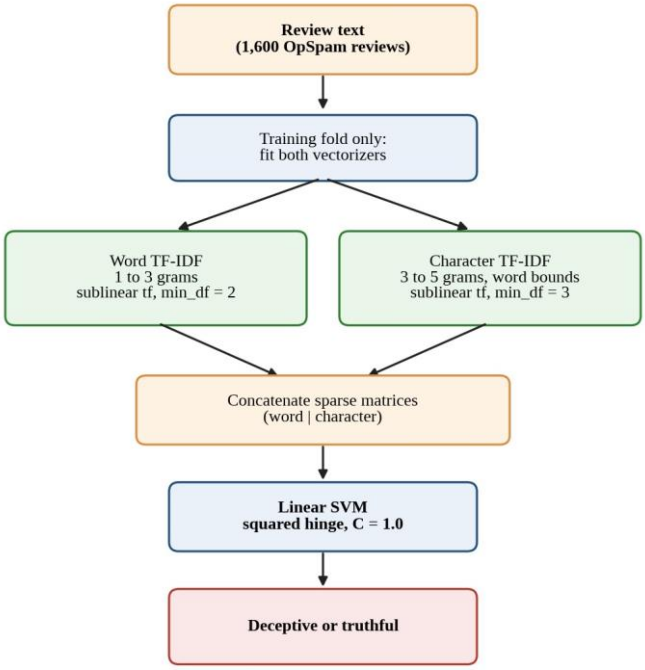


Fig. 1. Pipeline of the word and character TF-IDF fusion model. Both vectorizers are fit on the training fold only.

C. Evaluation Protocol

Three splits were used. The headline split is stratified 5-fold cross-validation on label and polarity with seed 42, the same type of split as [6]. For variance we repeated stratified 5-fold ten times with different shuffles (seed 7), giving 50 test folds, and compared models on those folds with the corrected resampled t-test [25]. Finally, we ran 5-fold cross-validation grouped by hotel, with four hotels in every test fold, to see how models behave on hotels they have never seen. We did not run BERT ourselves. Its numbers are those published in [6], on that paper's own fold assignments, so differences of a point or so between its figures and ours should be read against our fold-to-fold standard deviation of 1.3 to 1.6 points.

V. EXPERIMENTAL SETUP

All code is Python 3 with scikit-learn 1.8.0 [23], run on a CPU. The linear SVM uses the library defaults apart from C = 1 (also the default), and logistic regression uses 1,000 iterations. No hyperparameter search was run. The fusion configuration is the only one we tried after seeing the baselines, and the word-only and character-only ablations were run afterwards to explain it. Fitting and predicting one fusion model takes 2.1 s per fold on average (2.3 s at most), including vectorization.

VI. RESULTS

A. Comparison under the BERT Paper's Protocol Table III sets our numbers beside those of Kennedy et al. [6]. Under the same type of split the fusion model scores 91.12% (standard deviation 1.27) on seed 42 and 90.78% (1.44) over 50 folds, against 89.1% for their BERT. That is a lead of 2.0 and 1.7 points. Their bootstrap figure for BERT is 90.5%, which leaves 0.6 and 0.3 points. Fig. 2 places all of these next to the literature.

TABLE III

ACCURACY (%) OF THIS STUDY AND OF THE MODELS IN [6]. "10×5" IS THE MEAN OVER TEN REPEATS OF STRATIFIED 5-FOLD. "BY HOTEL" HOLDS OUT WHOLE HOTELS

Model	5-fold [K]	Boot. [K]	Seed 42	10×5	By hotel
NB, unigram	n/a	n/a	85.56	85.92	85.38
LR, uni+bigram	85.6	86.9	88.19	87.92	87.06

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 10, October 2026

DOI: 10.17148/IJARCCE.2026.151085

SVM, uni+bigram	86.4	88.2	89.38	89.44	88.44
SVM, word 1 to 3	n/a	n/a	90.25	89.96	89.12
SVM, char 3 to 5	n/a	n/a	90.31	89.72	89.06
SVM, word + char	n/a	n/a	91.12	90.78	89.12
FFNN, BoW	88.8	88.3	n/a	n/a	n/a
LSTM, BoW	87.6	87.6	n/a	n/a	n/a
BERT, fine-tuned	89.1	90.5	n/a	n/a	n/a

[K] denotes the values published in the BERT paper.

Our classical baselines also come out higher than theirs. Their logistic regression and SVM score 85.6% and 86.4%, ours 88.19% and 89.38% on seed 42. We did not set out to improve on those baselines and cannot say why the gap exists. Bigram features are one plausible contributor, but their feature settings are not published. What the table does show is that a plain unigram and bigram SVM, at 89.44% over ten repeats, is already level with the BERT figure before any fusion is involved.

B. What the Fusion Adds

Character n-grams alone score no higher than word n-grams alone: 89.72% against 89.96% over ten repeats (Table III). Combining the two reaches 90.78%, which is 0.8 points above the better single view and 1.34 above the unigram and bigram SVM. The fusion beat that SVM in 40 of the 50 folds (Fig. 3), but the corrected resampled t-test gives $p = 0.053$ [25]. We regard the improvement as probable and not established.

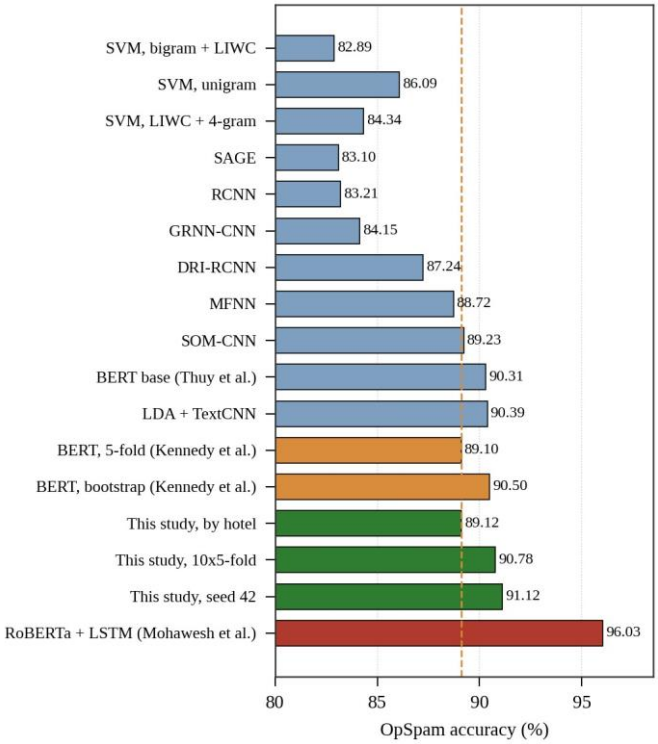


Fig. 2. Accuracy on OpSpam: results compiled in [15, Table 4] (blue), the BERT figures of [6] (orange), this study (green) and the RoBERTa and LSTM result of [15] (red). The dashed line marks 89.1%, the 5-fold BERT accuracy of [6].

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 10, October 2026

DOI: 10.17148/IJARCCE.2026.151085

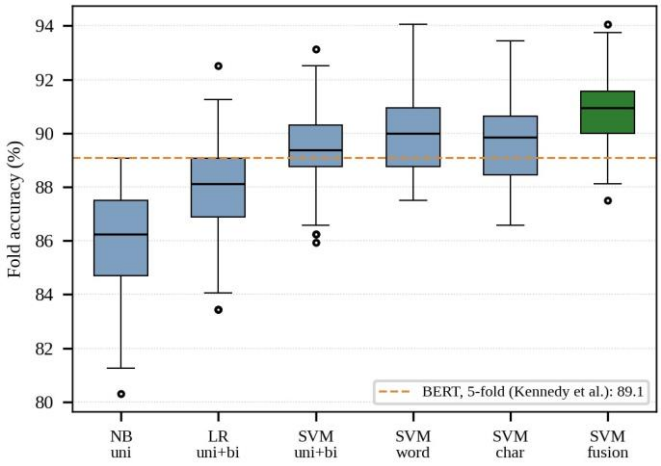


Fig. 3. Accuracy over 50 test folds (ten repeats of stratified 5-fold) for each model. Boxes span the quartiles. The dashed line is the 5-fold BERT accuracy reported in [6].

C. Held-Out Hotels

With whole hotels held out, every n-gram model loses accuracy relative to its ten-repeat mean, by between 0.5 and 1.7 points (Table III). The fusion loses the most, falling to 89.12% (standard deviation 1.44), and it ties the word-only model at 89.12%. The BERT figure of [6] is 89.1%, but it was measured on stratified splits that do not hold out hotels, so this is not a like-for-like comparison either. The result does suggest that part of the n-gram models' accuracy comes from hotel-specific vocabulary that leaks across ordinary folds.

D. Audit of the 96.03% Result

Mohawesh et al. [15] report 96.03% on OpSpam for a RoBERTa and LSTM hybrid, more than five points above any other row of Table II. We could not rerun their model, so this section examines only how the number was produced. The paper describes an 80/20 train-test split and reports no cross-validation, confidence interval or run-to-run variance. Its OpSpam precision, recall and accuracy (91.48%, 99.31% and 96.03%) are hard to reconcile with one balanced test set of 320 reviews. Solving for the share of deceptive reviews that would make them consistent gives about 40%, which is 3.6 standard deviations below the 50% a random 20% sample should contain, and 96.03% of 320 is not a whole number of reviews. The authors may have averaged over several runs, which the paper does not say. None of this shows the model is weak. It does mean the figure should not sit beside cross-validated results without a caveat, and we mark it that way in Table II and the figures.

VII. DISCUSSION

Under the BERT paper's own protocol, a linear SVM on word and character n-grams finishes 1.7 to 2.0 points ahead of fine-tuned BERT. We regard that as a fair finding and a narrow one. The BERT figure is a published number that we did not reproduce, our folds differ from theirs, the fusion's gain over a simpler SVM is not significant at the 5% level, and with hotels held out the n-gram variants converge at about 89%. A defensible reading is that on 1,600 reviews n-gram features and fine-tuned BERT are roughly level, with the n-gram models far cheaper to run: about two seconds per fold on a CPU and no pretrained weights.

We had expected character n-grams to carry the gain, on the idea that subword tokenizers smooth over the spelling and punctuation habits that distinguish rushed crowd-worker text from genuine reviews. The ablation does not support that idea. Character-only and word-only models perform alike, and what helps is combining them, which suggests the two views are partly complementary. We did not test the tokenization explanation directly, for instance by comparing BERT with a character-level neural model, so it remains a hypothesis.

The study has clear limits. OpSpam is small, covers one domain and one city, and its deceptive half was written to order, so its resemblance to real fraud is unknown. Kennedy et al. found the Yelp data much harder: their SVM and logistic regression score 72.1% and 71.3% there [6]. Accuracy on OpSpam should not be expected to carry over to platform data. We also chose the fusion configuration after seeing baseline results on the same folds. Only one configuration was tried, so selection effects should be small, but no separate test set was held out.

VIII. ERROR ANALYSIS AND PRACTICAL IMPLICATIONS

A. Interpreting the Feature Views

The word and character representations capture different levels of textual regularity. Word n-grams preserve short

DOI: 10.17148/IJARCCE.2026.151085

phrases and local word order, so they can represent constructions such as repeated praise, unusual adjective combinations, or recurring phrase templates. Character n-grams instead encode fragments within words, including spelling patterns, suffixes, punctuation habits, and partial word forms. Their usefulness does not depend on a complete word being present in the vocabulary. This can help when reviewers use inconsistent spelling or when a small dataset contains many rare tokens. However, character fragments are not intrinsically deception indicators: legitimate writers also vary their spelling and punctuation, and a model may learn topic or author cues that happen to correlate with the labels.

The ablation results in this study support a restrained interpretation. Word-only and character-only models are close in repeated cross-validation, while their concatenation produces a modest increase. This pattern is consistent with complementary information, but it does not prove that the character view identifies deception-specific behaviour. A stronger explanation would require inspecting influential features and reviewing examples where the combined model differs from each single-view model. In particular, top-weighted terms should be checked for hotel names, source-specific vocabulary, sentiment words, and artifacts of the corpus construction. If a feature is predictive because it identifies where a review came from rather than whether it is deceptive, it may fail when moved to another platform.

B. Error Analysis Needed for Deployment

Aggregate accuracy describes the proportion of correct predictions, but it does not show which errors are made. In deceptive-review screening, false positives may wrongly flag genuine customers, while false negatives allow fabricated reviews to remain visible. The operational cost of these errors depends on how the system is used. A tool that prioritizes reviews for human moderation can tolerate a different threshold from a system that automatically removes content. For this reason, future evaluation should report the confusion matrix, precision, recall, macro-F1, and class-specific error rates alongside accuracy. Precision-recall analysis is also useful when the real-world proportion of deceptive reviews differs from the balanced 50% prevalence in OpSpam.

The present study does not provide a manually adjudicated error analysis, so no claim is made about specific error categories or their frequency. A useful follow-up would sample false positives and false negatives from each outer test fold, hide model predictions from reviewers, and assign a small set of documented error labels. Possible categories include sentiment intensity, generic praise, specific factual detail, unusual language, topic mismatch, and source or hotel cues. Two annotators could label a shared subset and report agreement before resolving disagreements. This would help distinguish genuine linguistic signals from dataset shortcuts and provide evidence for whether the extra character features improve errors that matter in practice.

C. Thresholds, Calibration, and Human Review

The linear SVM produces a decision score rather than a directly calibrated probability. A deployment workflow should therefore avoid interpreting its raw margin as the probability that a review is fake. If confidence estimates are required, calibration must be fitted using training data only, preferably within the cross-validation procedure, and assessed on held-out data. The decision threshold should be selected using an explicit cost trade-off rather than automatically using the default boundary. One practical arrangement is a three-way queue: high-risk reviews receive priority for inspection, low-risk reviews remain unflagged, and borderline cases are sent for manual review. This design keeps the model as a decision-support mechanism instead of treating its output as proof of deception.

IX. REPRODUCIBILITY AND THREATS TO VALIDITY

A. Data and Label Validity

OpSpam is valuable because it is public, balanced, and widely used, but its labels require careful interpretation. Deceptive reviews were produced by paid crowd workers, whereas truthful reviews were collected from online sources. The two groups can therefore differ in writing conditions, source, and collection process as well as in deception. A classifier may exploit these differences even if it does not learn a general signal of dishonest intent. The corpus also covers hotels in one city and does not represent the full range of products, services, languages, or review-platform policies. Results should consequently be described as performance on this benchmark, not as a general estimate of real-world fake-review detection.

A more demanding validation set would include reviews collected from multiple platforms and time periods, with labels established independently of the linguistic features used by the model. Such labels are difficult to obtain because the truthfulness of an online review is rarely observable from text alone. Future work should document the evidence used to label each review, separate uncertain cases, and measure the effect of label noise. When possible, external validation should preserve the original model and threshold without retraining on the test domain. If domain adaptation is performed, its data requirements and evaluation protocol should be reported separately.

B. Split Design and Model Selection

The study uses both stratified folds and folds grouped by hotel because they answer different questions. Stratified

 This work is licensed under a Creative Commons Attribution 4.0 International License

DOI: 10.17148/IJARCCE.2026.151085

cross-validation estimates performance when reviews from the same set of hotels may appear in training and testing. Hotel-grouped validation instead tests transfer to hotels unseen during training and is a more stringent check of dependence on hotel-specific vocabulary. The drop under grouped evaluation is evidence that split design affects the measured score; it is not, by itself, proof that all of the lost performance is caused by leakage. Hotel identity may be associated with legitimate differences in language, polarity, and review content. The important point is that the two protocols measure different generalization settings and should not be presented as interchangeable.

There is also a model-selection limitation. The word-plus-character configuration was chosen after baseline results were observed on the same corpus, and the ablations were examined afterward. The resulting estimates can be optimistic because the evaluation data influenced the choice of configuration. A stronger design would reserve an untouched test set or use nested cross-validation: inner folds would select n-gram ranges, minimum document frequencies, and regularization settings, while outer folds would estimate final performance. All preprocessing, feature selection, and probability calibration must be fitted inside the training portion of each fold. Repeatedly trying configurations against the same test folds should be avoided.

C. Statistical and Computational Reporting

The corrected resampled t-test gives $p = 0.053$ for the comparison between the fusion model and the plain unigram-bigram SVM. This result does not cross the conventional 0.05 threshold, so the current evidence is insufficient to claim a statistically established improvement. At the same time, a p-value just above 0.05 does not prove that the models are equivalent. The estimated effect, variability across folds, evaluation design, and practical cost should be considered together. Future reports should provide confidence intervals for the metrics, paired per-fold comparisons, and a pre-specified primary metric. Because repeated folds overlap in their training data, uncertainty estimates should use methods that account for this dependence.

The reported training time of about two seconds per fold indicates that the classical pipeline is inexpensive on the hardware used in this study. It should not be treated as a universal speed benchmark: runtime depends on processor, memory, software versions, sparse-matrix operations, and corpus size. A reproducible report should record hardware, package versions, random seeds, vectorizer settings, and exact fold assignments. It should also distinguish training time from inference latency and preprocessing time. Transformer comparisons should include the cost of fine-tuning and inference, as well as model size and memory needs, rather than comparing only accuracy. These measurements would make the efficiency trade-off more useful for practitioners choosing between methods.

D. Recommended Evaluation Protocol

A practical next experiment is a nested, hotel-grouped comparison in which every model receives the same outer folds. The classical baselines, word-only SVM, character-only SVM, and fusion model should be trained from scratch within each outer training partition. BERT and RoBERTa should use the identical partitions, with their hyperparameters selected without access to the outer test fold. The main report should include accuracy, macro-F1, precision, recall, confidence intervals, and per-hotel performance. An external dataset should then be used to assess transfer beyond OpSpam. This protocol is more computationally demanding than the current baseline study, but it would address the main uncertainty: whether the apparent advantage persists when model selection, hotel identity, and architecture are controlled consistently.

X. FINAL OBSERVATION OF LITERATURE REVIEW

A. Key Trends Observed

Three patterns stand out. First, accuracy on OpSpam has stayed within a narrow band: eleven of the twelve rows in Table II fall between 82.9% and 90.4% [15]. The architectures behind them changed several times (Fig. 4), from SVMs to CNNs, recurrent networks and BERT, and the points rise slowly and unevenly. The oldest feature-based result, 91.2% [5], is still above most of what followed, though it was measured on the smaller original subset. Second, recent work adds goals other than accuracy: SHAP explanations [15], sentiment-aware training [13] and compact models [12]. Third, the highest figure, 96.03%, rests on a weaker evaluation than most points below it.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 10, October 2026

DOI: 10.17148/IJARCCE.2026.151085

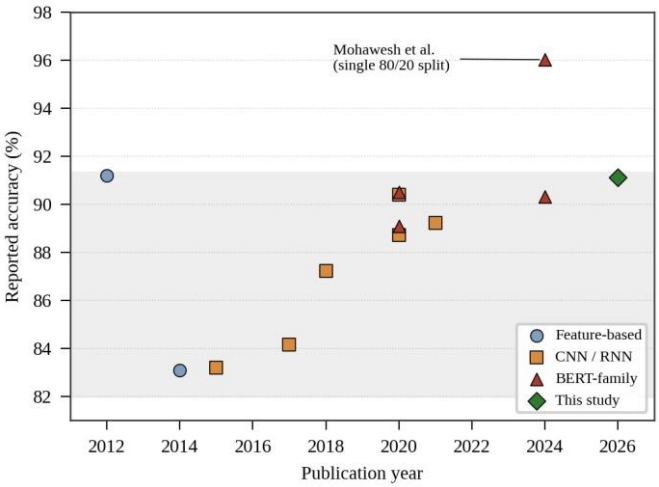


Fig. 4. Reported OpSpam accuracy by publication year. Values are from [15] except Feng et al. (from [6]), Kennedy et al. and this study. The shaded band covers 82 to 91.3%. The Feng et al. figure predates the negative-polarity half of the corpus.

B. Overall Research Gap

Three gaps remain. Protocols differ even on one corpus, with an 80/20 split in [15] and 5-fold plus bootstrap in [6], so some of the ranking between papers reflects how the numbers were produced. Evaluation that holds out hotels, which is closer to deployment, is absent from the papers we read, and we found that n-gram models lose up to 1.7 points under it. And every result in Table I is on English text. Non-English review corpora exist, for example Italian cultural-heritage reviews [26], but in the papers we read we found no OpSpam-style comparison between classical and transformer models in another language.

XI. CONCLUSION AND FUTURE WORK

A linear SVM on word and character TF-IDF features scores 91.12% on OpSpam under stratified 5-fold cross-validation, or 90.78% averaged over ten repeats, ahead of the 89.1% published for fine-tuned BERT under the same type of split [6]. The lead is real in those numbers and fragile in others. A plain n-gram SVM is already at 89.4%, the fusion's gain over it has $p = 0.053$, and holding out hotels brings every variant down to about 89%. The 96.03% reported for a RoBERTa and LSTM hybrid comes from an evaluation we could not match against any of this.

Four pieces of follow-up work are planned. We will rerun BERT and RoBERTa ourselves on the folds used here, including the hotel-grouped split, so that no number in the comparison is borrowed. We will test the tokenization hypothesis with a character-level neural baseline. We will repeat the hotel-grouped protocol with a hybrid of a transformer and n-gram features. And we will carry the same protocol to French reviews, for which we know of no labeled deception benchmark, starting with the question of how such a dataset could be built.

REFERENCES

[1] N. Jindal and B. Liu, "Review spam detection," in Proc. 16th Int. Conf. World Wide Web (WWW), New York, NY, USA, 2007, pp. 1189–1190.

[2] R. Mohawesh et al., "Fake reviews detection: A survey," IEEE Access, vol. 9, pp. 65771–65802, 2021.

[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: Human Lang. Technol., Portland, OR, USA, 2011, pp. 309–319.

[4] M. Ott, C. Cardie, and J. T. Hancock, "Negative deceptive opinion spam," in Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics: Human Lang. Technol. (NAACL-HLT), Atlanta, GA, USA, 2013, pp. 497–501.

[5] S. Feng, R. Banerjee, and Y. Choi, "Syntactic stylometry for deception detection," in Proc. 50th Annu. Meeting Assoc. Comput. Linguistics (Short Papers), Jeju Island, South Korea, 2012, pp. 171–175.

[6] S. Kennedy, N. Walsh, K. Sloka, A. McCarren, and J. Foster, "Fact or factitious? Contextualized opinion spam detection," arXiv:2010.15296, Oct. 2020.

[7] J. Li, M. Ott, C. Cardie, and E. Hovy, "Towards a general rule for identifying deceptive opinion spam," in Proc. 52nd Annu. Meeting Assoc. Comput. Linguistics (Long Papers), Baltimore, MD, USA, 2014, pp. 1566–1576.

[8] S. Lai, L. Xu, K. Liu, and J. Zhao, "Recurrent convolutional neural networks for text classification," in Proc. 29th AAAI Conf. Artif. Intell., Austin, TX, USA, 2015.

 This work is licensed under a Creative Commons Attribution 4.0 International License

DOI: 10.17148/IJARCCE.2026.151085

[9] Y. Ren and D. Ji, "Neural networks for deceptive opinion spam detection: An empirical study," Inf. Sci., vol. 385, pp. 213–224, 2017.

[10] W. Zhang, Y. Du, T. Yoshida, and Q. Wang, "DRI-RCNN: An approach to deceptive review identification using recurrent convolutional neural network," Inf. Process. Manage., vol. 54, pp. 576–592, 2018.

[11] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics: Human Lang. Technol., Minneapolis, MN, USA, 2019, pp. 4171–4186.

[12] D. T. T. Thuy, L. T. M. Thuy, N. C. Bach, T. T. Duc, H. G. Bach, and D. D. Cuong, "Designing a deep learning-based application for detecting fake online reviews," Eng. Appl. Artif. Intell., vol. 134, 2024, Art. no. 108708.

[13] R. Catelli, H. Fujita, G. De Pietro, and M. Esposito, "Deceptive reviews and sentiment polarity: Effective link by exploiting BERT," Expert Syst. Appl., vol. 209, 2022.

[14] S. B. Hyder et al., "BERT-based deceptive review detection in social media: Introducing DeceptiveBERT," IEEE Trans. Comput. Social Syst., vol. 11, no. 6, pp. 7234–7243, Dec. 2024.

[15] R. Mohawesh, H. Bany Salameh, Y. Jararweh, M. Alkhalaileh, and S. Maqsood, "Fake review detection using transformer-based enhanced LSTM and RoBERTa,"Int. J. Cogn. Comput. Eng., vol. 5, pp. 250–258, 2024, doi: 10.1016/j.ijcce.2024.06.001.

[16] L. Cagnina and P. Rosso, "Classification of deceptive opinions using a low dimensionality representation," in Proc. 6th Workshop Comput. Approaches Subjectivity, Sentiment Social Media Anal., 2015, pp. 58–66.

[17] D. H. Fusilier, M. Montes-y-Gómez, P. Rosso, and R. G. Cabrera, "Detecting positive and negative deceptive opinions using PU-learning," Inf. Process. Manage., vol. 51, pp. 433–443, 2015.

[18] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in Proc. 14th Int. Joint Conf. Artif. Intell. (IJCAI), San Francisco, CA, USA, 1995, pp. 1137–1143.

[19] N. Cao, S. Ji, D. K. Chiu, M. He, and X. Sun, "A deceptive review detection framework: Combination of coarse and fine-grained features," Expert Syst. Appl., 2020, Art. no. 113465.

[20] C. Jiang, X. Zhang, and A. Jin, "Detecting online fake reviews via hierarchical neural networks and multivariate features," in Proc. Int. Conf. Neural Inf. Process. (ICONIP), Bangkok, Thailand, 2020, pp. 730–742.

[21] A. Neisari, L. Rueda, and S. Saad, "Spam review detection using self-organizing maps and convolutional neural networks," Comput. Secur., vol. 106, 2021, Art. no. 102274.

[22] E. Stamatatos, "A survey of modern authorship attribution methods," J. Amer. Soc. Inf. Sci. Technol., vol. 60, no. 3, pp. 538–556, Mar. 2009.

[23] F. Pedregosa et al., "Scikit-learn: Machine learning in Python," J. Mach. Learn. Res., vol. 12, pp. 2825–2830, 2011.

[24] C. Cortes and V. Vapnik, "Support-vector networks," Mach. Learn., vol. 20, no. 3, pp. 273–297, 1995.

[25] C. Nadeau and Y. Bengio, "Inference for the generalization error," Mach. Learn., vol. 52, no. 3, pp. 239–281, Sep. 2003.

[26] R. Catelli et al., "A new Italian cultural heritage data set: Detecting fake reviews with BERT and ELECTRA leveraging the sentiment," IEEE Access, vol. 11, pp. 52214–52225, 2023, doi: 10.1109/ACCESS.2023.3277490.