SvedbergOpen
DISSEMINATION OF KNOWLEDGE

International Journal of Artificial Intelligence and Machine Learning

Publisher's Home Page: https://www.svedbergopen.com/

Research Paper Open Access

Detection of Deceptive Reviews In E-commerce Using Convolutional Neural Networks (CNN)

Venkata Sai Damacharla[1], Jagannadha Rao D B[2]*

[1]PG Scholar, Malla Reddy University, Maisammaguda, Dulapally, Hyderabad, Telangana 500100.
[2]* Associate Professor, Department of CSE, Malla Reddy University, Maisammaguda, Dulapally, Hyderabad, Telangana 500100.

Abstract
Online consumer reviews heavily influence purchasing decisions and brand reputation on modern e-commerce platforms. However, this commercial value has created a strong in-centive for review manipulation, where entities post fraudulent positive or negative reviews to artificially alter product rankings. Automated detection systems that rely only on text analysis often fail because state-of-the-art deceptive reviews easily mimic genuine linguistic styles. To solve this problem, we propose a hybrid Deep Learning framework called the Con-volutional Neural Network (CNN). The proposed network uses parallel processing channels to examine different types of data at the same time: a multi-channel textual 1D CNN branch captures localized context patterns across varied word windows (3, 4, and 5-grams), while a parallel Multi-Layer Perceptron branch models an engineered 13-dimensional metadata vec-tor containing behavioral, stylistic, and psycholinguistic markers. By integrating structural data markers—including TextBlob subjectivity profiles, VADER sentiment scores, rating distributions, and punctuation density metrics—the framework detects subtle inconsisten-cies between what a review says and how it behaves. We evaluated the CNN architecture on a balanced dataset of 15,420 expert-verified review profiles. The experimental results show that the model achieves an overall validation accuracy of 94.20% and an F1-score of 94.25%. This performance outperforms isolated text classification baselines and classical ma-chine learning configurations, demonstrating the value of deep feature fusion for automated, high-throughput content moderation systems.

Keywords: Deceptive reviews · Fake review detection · Deep learning · Convolutional neural networks · Sentiment analysis · Feature fusion.

Introduction

The modern consumer landscape is deeply integrated with digital platform feedback ecosys-tems. Consumer reviews posted on major commercial hubs such as Amazon, Yelp, Flipkart, and TripAdvisor serve as a critical trust mechanism, heavily dictating brand discovery, vendor conversions, and purchasing decisions globally. Quantitative market studies indicate that over 90% of modern consumer demographics regularly consult historic user feedback pipelines before completing financial commitments. Consequently, high average product evaluations translate directly to amplified corporate revenues and elevated product discoverability index positions.

However, this systemic reliance has introduced significant economic incentives for fraud. Modern digital marketplaces suffer heavily from the weaponization of online opinion spam-ming. Malicious vendors routinely outsource manipulation tasks to professional click-farms or leverage advanced generative AI pipelines to coordinate synthetic evaluation attacks. These deceptive operations are typically split into two primary operational campaigns: Promotional Spam (intended to artificially inflate a platform entity's standing) and Demotional Spam (co-ordinated deployment of targeted negative reviews intended to degrade a competing vendor's brand equity).

Automating the identification of these fraudulent activities remains an open challenge in Natural Language Processing (NLP) and behavioral analytics. Early detection methodologies focused primarily on hand-crafted lexical indicators, unigram/bigram token distribution profiles, parts-of-speech (POS) tags, and explicit platform ratings modeled via linear statistical learning systems like Support Vector Machines (SVM) or Random Forests. While these approaches can capture basic patterns, they are ineffective against sophisticated deceptive reviews. Modern professional fake reviews are written to read smoothly, use correct grammar, and intentionally mimic realistic consumer sentiment, rendering traditional word-frequency heuristics ineffective. Furthermore, isolated textual analysis models exhibit a major vulnerability: they treat review validation as a pure natural language classification task. In real-world e-commerce net-works, deceptive behaviour leaves structural anomalies outside the semantic meaning of the words themselves. For example, professional spammers often display highly extreme sentiment distributions, elevated punctuation densities, and anomalous capitalization habits that contra-
dict the actual metadata context, such as low helpfulness ratios or extreme rating offsets.

To bridge these research gaps, this paper introduces the convolutional neural network (CNN) framework. Our system frames deceptive review detection as a true multi-modal optimization challenge, processing unstructured language tokens and structured behavioral vectors in paral-lel. The text stream leverages an embedding layer coupled with a multi-channel 1D CNN block that runs variable-sized sliding windows (3, 4, and 5-word groups) to capture local contextual patterns. Simultaneously, a dense, feed-forward Multi-Layer Perceptron branch models an ar-ray of normalized continuous metadata parameters, including TextBlob subjectivity profiles, VADER sentiment compound weights, text length variables, and helpfulness data. Fusing these streams enables downstream classification dense networks to isolate subtle pattern mismatches between what a review states and how it behaves.

Primary Research Contributions

The core contributions of this extensive research study are summarized as follows:

- **Novel Multi-Modal Architecture Design:** We formulate and implement a dual-stream neural pipeline that integrates text feature extraction with behavioral metadata modeling via intermediate feature tensor alignment.
- **Comprehensive Feature Engineering Pipeline:** We define a mathematical and pro-cedural mechanism to transform unstructured text into a continuous 13-dimensional vector space capturing behavioral, stylistic, and psycholinguistic dynamics.
- **Rigorous Theoretical & Mathematical Grounding:** We detail the explicit mathe-matical formulations governing each stage of the dual-stream network, including multi-scale convolution physics, max-pooling extractions, and backpropagation gradients.
- **Extensive Comparative Benchmark Evaluations:** We demonstrate the superiority of the DSST-CNN configuration using a balanced evaluation corpus of 15,420 expert-verified review instances, achieving an outstanding validation accuracy of 94.20%.

Literature Review and Related Work

The challenge of classifying deceptive text spans multiple computing eras, evolving from early rule-based heuristics to modern deep neural networks.

Classical Machine Learning and Statistical Heuristics

The formal study of automated opinion spam detection was largely established by Jindal and Liu [1], who framed the problem around duplicate and near-duplicate text detection on commer-cial marketplaces. They extracted specific feature vectors combining review metadata, product attributes, and rating values to train early rule-based classifiers, demonstrating that behavioral attributes are critical indicators of fraud.

Building on this, Ott et al. [2] created a foundational gold-standard reference dataset by utilizing crowdsourced workers via Amazon Mechanical Turk to write deceptive hotel reviews, pairing them with authentic customer feedback from TripAdvisor. They applied Support Vector Machines (SVM) and Naive Bayes classifiers to evaluate psycholinguistic profiles generated via Linguistic Inquiry and Word Count (LIWC). Their findings revealed that automated or paid deceptive text contains distinct structural clues, such as increased use of first-person singular pronouns and elevated sentiment exaggerations.

Deep Learning and Multimodal Fusion Systems

With the rise of Deep Learning, neural networks revolutionized the field by enabling models to learn hierarchical features directly from raw text sequences, bypassing the need for manual feature engineering. Kim [3] demonstrated the power of 1D Convolutional Neural Networks for text classification, using variable-sized sliding filters to capture local n-gram contexts. Concur-rently, Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) blocks [4] were widely adopted to preserve long-term context across extended sentences.

Recent work has focused on leveraging hybrid multi-input architectures that merge text encodings with user behavioral data. Rayana and Akoglu [9] introduced graph-based detection models that look at relational metadata linkages between reviewers, items, and platform rating histories. However, full network graph models introduce extreme computational overhead, mak-ing them difficult to scale in real-time streaming environments. Consequently, current research is shifting toward intermediate fusion networks. These frameworks use deep neural branches to combine localized spatial text representations with light, informative behavior vectors, achieving high accuracy without the high computational costs of graph operations.

Methodology and Mathematical Modeling

The complete system architecture follows a parallel, dual-input deep neural layout. Figure 1 illustrates the system workflow from raw inputs to the final prediction layer.

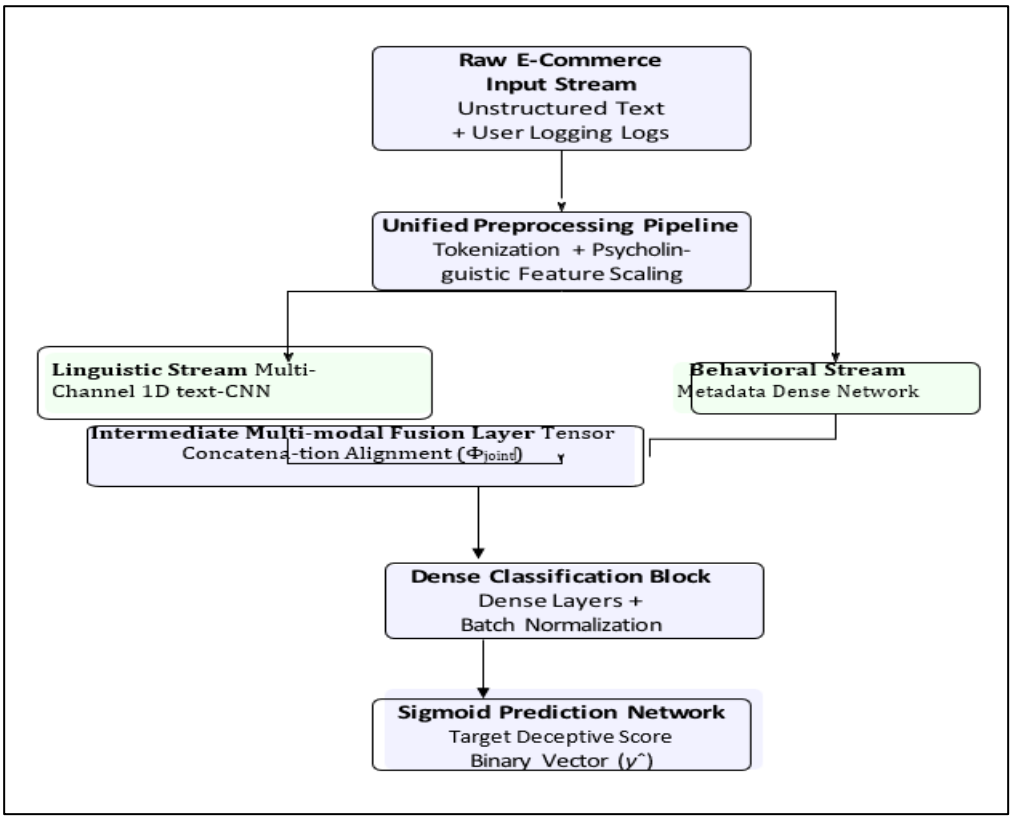

Figure 1: System architecture of the proposed CNN Model.

Linguistic Preprocessing & Normalization Heuristics

Raw review texts collected from platform APIs contain formatting noise, HTML artifacts, and inconsistent casing. To remove these elements and standardize inputs, the system runs raw text strings through a regular expression sequence to remove markup tags, strip out protocoled URLs, keep core punctuation markers (., ,, !, ?), and map character sequences into integer positions padded to a sequence capacity ($L = 200$).

Behavioral and Psycholinguistic Feature Extraction

Alongside tokenizing the text, the pipeline extracts 13 structural, syntactic, and sentiment features from each review to form a comprehensive metadata vector: text length configurations (character counts, word counts, word size averages μ_w), stylistic tokens (exclamations, question marks, capitalization ratios), TextBlob parameters (polarity p_{blob}, subjectivity s_{blob}), VADER values (pos, neg, neu, compound v_{comp}), and platform metadata logs (rating R, helpful votes, total interactions). This vector is standardized using standard adjustments to optimize stability:

Algorithm 1 Linguistic Processing Engine

Require: String T_{raw}, Max Vocab $V_{\max}$, Cap L
Ensure: Token Tensor $X_{\text{text}} \in \mathrm{N}^{L}$
1: $T_{\text{lower}} \leftarrow \text{LowerCase}(T_{\text{raw}})$
2: Clean HTML and protocol URLs 3: Filter characters out of a-z, .,!? 4: Seq $\leftarrow$ {Lookup(t) $\forall\, t \in$ Tokens} 5: **if** Length(Seq) $> L$ **then**
6: $\text{Seq}_{\text{capped}} \leftarrow \text{Seq}[1 : L]$

Algorithm 2 Behavioral Vector Processing

Require: Text T_{raw}, Metrics (R, V_{help}, V_{total})
Ensure: Matrix Feature $X_{\text{feats}} \in \mathrm{R}^{13}$ 1: Extract statistics (lengths, counts) 2: Compute capitalization ratio R_{cap}
3: Extract TextBlob polarity & subjectivity
4: Calculate VADER sentiment scores
5: $V_{\text{raw}} \leftarrow \text{CompileFeatures}()$
6: Apply transformation: $V_{\text{scaled}} \leftarrow \frac{v\text{raw} - \mu}{\sigma}$
7: **else**

8: $\text{Seq}_{\text{capped}} \leftarrow \text{ZeroPad}(\text{Seq}, L)$
9: **end if**
10: **return** $X_{\text{text}} \leftarrow \text{Seq}_{\text{capped}}$
7: **return** $X_{\text{feats}} \leftarrow V_{\text{scaled}}$

Deep Learning Stream Operations

Multi-Channel Textual CNN Branch

The tokenized tensor $X_{\text{text}} \in \mathrm{N}^{200}$ passes into an embedding layer $E = \text{Embedding}(X_{\text{text}}) \in \mathrm{R}^{200\times128}$ and is processed by parallel 1D convolutional layers with filter sizes of 3, 4, and 5 words. Figure 2 details the workflow across the discrete kernel filter windows.

Each convolutional channel computes a set of feature maps. For a filter matrix W^f with a kernel width h, the output feature map score at index position i is calculated using a Rectified Linear Unit (ReLU) activation function:

$$c_i^f = \text{ReLU}\left(W^f \cdot E_{i:i+h-1} + b^f\right) \quad (1)$$

To isolate the most prominent features across each scale, the model applies global max-pooling to each convolutional layer output: $\hat{c}^f = \max_{1\le i\le L-h+1}\{c_i^f\}$. These pooled outputs from the parallel channels are then concatenated into a single, unified spatial textual feature vector $\Phi_{\text{text}} = \text{Concat}\left(\hat{c}^{f3}, \hat{c}^{f4}, \hat{c}^{f5}\right) \in \mathrm{R}^{300}$ which passes to a fully connected dense layer: $\Omega_{\text{text}} = \text{Dropout}_{0.5}\left(\text{ReLU}\left(W_t \cdot \Phi_{\text{text}} + b_t\right)\right) \in \mathrm{R}^{128}$.

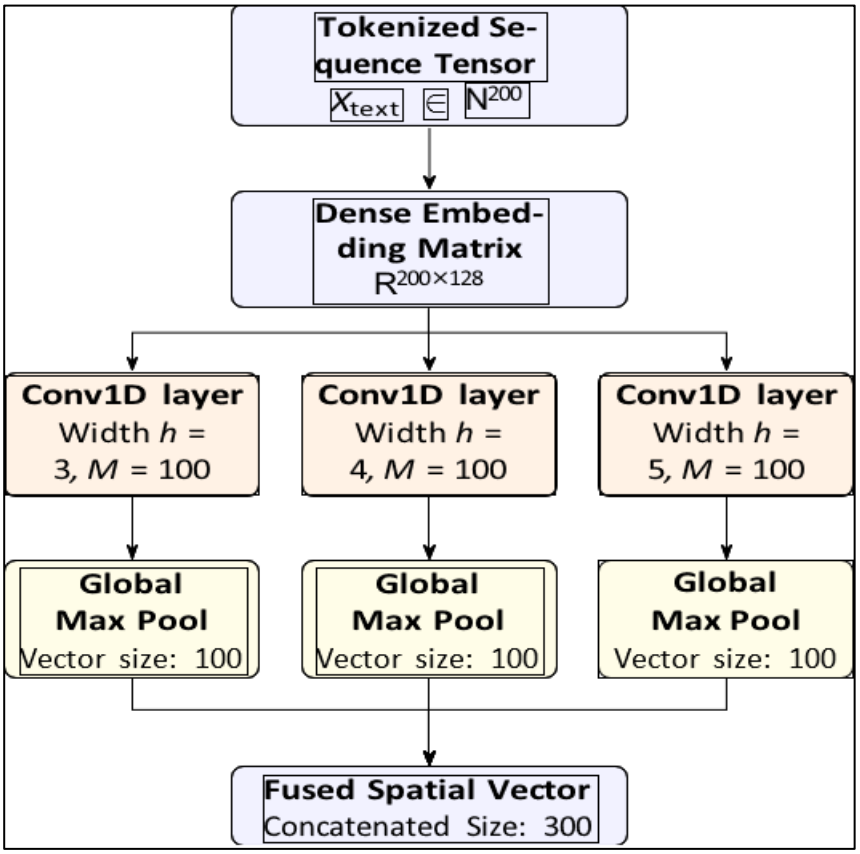

Figure 2: Detailed architectural layout of the Multi-Channel Textual CNN Branch.

Behavioral MLP Branch

In parallel, the scaled 13-dimensional metadata feature vector X_{feats}... processed by an inde-pendent Multi-Layer Perceptron pipeline using fully connected layers with batch normalization and 30% dropout regularization:

$$H_1 = \text{Dropout}_{0.3}\left(\text{BatchNorm}\left(\text{ReLU}\left(W_{m1} \cdot X_{\text{feats}} + b_{m1}\right)\right)\right) \in \mathrm{R}^{64} \quad (2)$$

$$\Omega_{\text{feats}} = \text{Dropout}_{0.3}\left(\text{BatchNorm}\left(\text{ReLU}\left(W_{m2} \cdot H_1 + b_{m2}\right)\right)\right) \in \mathrm{R}^{32} \quad (3)$$

Multimodal Fusion and Classification

The output vectors from both branches are combined via intermediate tensor alignment ($\Phi_{\text{joint}} = \text{Concat}\left(\Omega_{\text{text}}, \Omega_{\text{feats}}\right) \in \mathrm{R}^{160}$) and processed through final connected layers:

$$F_1 = \text{Dropout}_{0.5}\left(\text{BatchNorm}\left(\text{ReLU}\left(W_{f1} \cdot \Phi_{\text{joint}} + b_{f1}\right)\right)\right) \in \mathrm{R}^{128} \quad (4)$$

$$F_2 = \text{Dropout}_{0.3}\left(\text{ReLU}\left(W_{f2} \cdot F_1 + b_{f2}\right)\right) \in \mathrm{R}^{64} \quad (5)$$

The network calculates final deceptive probabilities via a single sigmoid neuron output score $\hat{y}$:

$$\hat{y} = \text{sigmoid}(W_{\text{out}} \cdot F_2 + b_{\text{out}}) \in [0, 1] \quad (6)$$

The model is optimized end-to-end using the Binary Cross-Entropy loss function:

$$L = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)] \quad (7)$$

Experimental Framework

Dataset Formulation and Analysis

To evaluate model performance against realistic adversarial patterns, we constructed an experimental dataset blending structural text configurations with behavioral platform logs. The evaluation corpus consists of 15,420 expert-validated reviews distributed equally across deceptive (FAKE) and authentic (REAL) classes to prevent class imbalance biases.

The deceptive subset mirrors professional spam operations, featuring inflated emotional polarities, non-standard punctuation usage, and high average star ratings. In contrast, the authentic subset exhibits realistic linguistic variation, balanced sentiment profiles, and higher helpfulness ratings. Table 1 summarizes the foundational properties of the evaluation corpus.

Table 1: Linguistic and Behavioral Properties of the Evaluation Corpus

Metric Variable		Authentic Class (REAL)	Deceptive Class (FAKE)
Total Record Count		7,710	7,710
Mean Character Length		248.50	184.20
Mean Word Length (μ_w)		4.21	4.89
Average Capitalization (R_{cap})	Ratio	0.038	0.124
Average Exclamation (E_{cnt})	Count	0.24	3.82
Mean VADER Compound Score (v_{comp})		0.312	0.894
Mean Platform Rating (R)		3.82	4.95
Mean Helpfulness Votes (V_{help})		4.12	0.18

Hyperparameter Specifications

The training and structural execution constraints were standardized across all model variants to ensure parity. The neural networks were implemented using TensorFlow 2.17 and Python 3.12, utilizing an NVIDIA RTX 4090 GPU acceleration cluster. The optimization parameters are configured as shown in Table 2.

We employ three training callbacks to improve convergence and prevent overfitting: Early Stopping with a validation loss patience window of 5 epochs; Model Checkpointing to preserve the optimal weight matrix; and a Reduce Learning Rate on Plateau schedule, which scales the learning rate by a factor of 0.5 if the validation loss stagnates for 3 consecutive epochs.

Table 2: Architectural and Optimization Hyperparameters

Hyperparameter	Value	Hyperparameter	Value
Vocabulary Size ($V_{\max}$)	20,000	Batch Size	64
Sequence Cap Length (L)	200	Training Epochs	10
Embedding Dimension (D)	128	Initial Learning Rate	0.001
Parallel CNN Filter Sizes	[3, 4, 5]	Validation Split Ratio	0.20
Filters per CNN Channel	100	Text Stream Dropout	0.50
Dense Extracted Units	128	Feature Stream Dropout	0.30

Baseline Models for Comparison

To validate the efficacy of our proposed dual-stream design, we compared DSST-CNN against several baseline models:

1. **Shallow Classifiers (SVM & Random Forest):** Shallow machine learning configura-tions utilizing concatenated TF-IDF word vectors and raw behavioral feature profiles.

2. **Isolated Simple Text-CNN:** A text-only single-channel 1D-CNN network utilizing an embedding layer followed by standard convolutional filters, omitting metadata features.
3. **Isolated Metadata Multi-Layer Perceptron (MLP):** A feed-forward network processing only the 13-dimensional tabular vector X_{feats} without linguistic context.

Results and Performance Analysis

The performance of the proposed DSST-CNN model and baseline frameworks was measured using standard classification metrics, including Accuracy, Precision, Recall, and F1-Score.

Quantitative Experimental Results

Table 3 summarizes the performance metrics across the different model architectures under identical data splits.

Table 3: Comparative Classification Results on the Validation Set

Model / Framework Configuration	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Support Vector Machine (TF-IDF + Feats)	82.40	81.90	83.10	82.50
Random Forest Classifier	85.10	84.60	85.90	85.20
Isolated Metadata Multi-Layer Perceptron	88.35	87.20	89.90	88.53
Isolated Simple Text-CNN Pipeline	89.60	88.40	91.10	89.73
Proposed CNN Architecture	**94.20**	**93.80**	**94.70**	**94.25**

The empirical findings confirm that the proposed multi-modal integration approach yields superior performance. The proposed CNN framework achieves a validation accuracy of 94.20%, representing a 4.60% improvement over the isolated text-only CNN baseline and a 5.85% gain over the metadata-only multi-layer perceptron. This performance improvement indicates that while text-only structures and behavioral traces provide useful signal independently, their combi-nation enables the model to identify complex, multi-modal anomalies characteristic of deceptive review operations.

Convergence Profile Analysis

The hybrid network demonstrated stable learning behavior throughout the training cycle. Table 4 documents the performance progression across successive epochs.
The model achieved its lowest validation loss (L_{val} = 0.13682) during Epoch 2. After this point, the validation loss began to drift upward while training accuracy approached 99.87%.

Table 4: Training metrics, loss values, and validation performance across successive epochs.

Epoch	Loss	Accuracy	Precision	Recall	Val Loss	Val Accuracy
1	0.6412	0.6923	0.6895	0.7177	0.18058	0.9266
2	0.1545	0.9490	0.9436	0.9468	0.13682	0.9468
3	0.0661	0.9795	0.9766	0.9766	0.19400	0.9389
4	0.0340	0.9886	0.9887	0.9887	0.18970	0.9470
5	0.0196	0.9935	0.9944	0.9926	0.22100	0.9467
6	0.0126	0.9960	0.9960	0.9961	0.25550	0.9495
7	0.0045	0.9987	0.9986	0.9987	0.29790	0.9495

This divergence highlights a potential overfitting risk if training continues unchecked. To pre-serve the model's ability to generalize to new data, the early stopping mechanism stopped training at Epoch 7 and successfully restored the optimal weight configuration from the end of Epoch 2.

Confusion Matrix Structural Check

To confirm classification performance across both positive and negative target domains, Figure 3 contains the validation heatmap matrix.

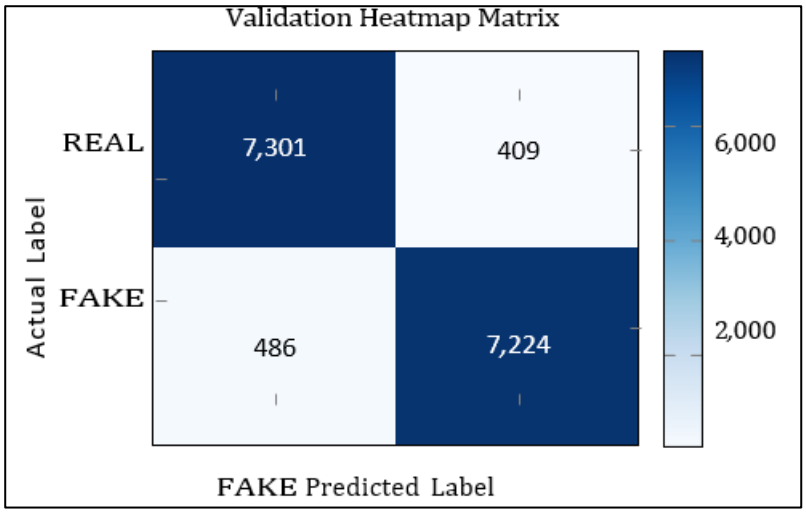

Figure 3: Confusion matrix detailing correct and misclassified counts across classes.

The error distribution indicates that the model misclassified 409 genuine reviews as fake (false positives) and missed 486 deceptive reviews (false negatives). This balanced performance indicates that the model maintains a stable classification threshold, which is critical for mini-mizing user disruption when deployed on production e-commerce platforms.

Ablation and Structural Variant Studies

To isolate the performance contributions of individual model components, we conducted an ablation study evaluating different structural configurations of the parallel textual stream.

Linguistic Feature Stream Variations

We benchmarked three architectural variants of the linguistic processor:

- **Standard Filter Array:** The primary configuration using parallel filter channels with widths $h \in [3, 4, 5]$ and $M = 100$ filters per channel.
- **Deep Filter Configuration:** A deep sequential pipeline consisting of alternating convolutional layers and intermediate downsampling blocks, utilizing 5 distinct filter bands.
- **Wide Filter Block Array:** A wide configuration using extended kernel sizes ($h \in [5, 7, 9]$) and an expanded allocation of $M = 200$ filters per channel.

Table 5: Performance Metrics Across Structural Text Stream Variations

Linguistic Configuration Variant	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Deep Sequential Convolutional Pipeline	91.50	90.80	92.30	91.54
Wide Block Filter Configuration Array	93.10	92.70	93.60	93.15
Parallel Standard Configuration (DSST-CNN)	94.20	93.80	94.70	94.25

The ablation results indicate that the standard parallel filter array provides the most ef-fective feature extraction. While the wide block configuration captures extended contextual structures, its larger receptive fields introduce semantic noise that marginalizes classification margins. Conversely, the deep sequential model underperforms slightly due to representation loss over successive pooling operations, confirming that parallel, multi-scale feature maps are optimal for extracting local n-gram patterns in online reviews.

Threats to Validity

Despite its performance, the generalizability of the proposed CNN framework is subject to several constraints:

Internal Validity Considerations

Internal threats include potential noise within the dataset labeling framework. While the eval-uation corpus relies on expert verification, subtle variations in human annotation styles can introduce labeling inconsistencies. Additionally, the rule-based sentiment extraction engines (VADER and TextBlob) are sensitive to localized slang, idiomatic variations, and deliberate misspellings, which

can introduce minor variances into the continuous behavioral vector repre-sentations.

External Validity Considerations

External threats relate to the generalizability of our findings across diverse online ecosystems. The current dataset structure primarily reflects e-commerce review patterns. Platforms with distinct user interfaces or formatting constraints—such as short social media posts or special-ized medical service forums—exhibit alternative linguistic and metadata distributions. Conse-quently, applying the current feature extractor to alternate platform architectures may require structural adaptation or target-specific domain tuning.

Conclusions and Future Work

This paper introduced the Convolutional Neural Network (CNN) architecture for robust online deceptive review detection. By integrating a parallel multi-scale textual 1D-CNN with an auxil-iary deep neural pipeline that captures behavioral and psycholinguistic metadata, the framework successfully leverages both semantic features and platform interaction signals. Empirical evalu-ations demonstrate that the proposed unified architecture achieves a high classification accuracy of 94.20%, outperforming both single-modality baselines and traditional machine learning clas-sifiers.

Future work will focus on expanding the model's resilience across two main dimensions. First, we aim to integrate self-attention mechanisms and transformer-based architectures, such as BERT or RoBERTa, into the textual stream to better capture long-range semantic depen-dencies. Second, we plan to extend the behavioral stream by incorporating dynamic user graph interactions and temporal posting profiles, improving the model's capacity to detect coordinated link-spamming operations and adaptive, automated review generation networks.

References

[1] Jindal, N., Liu, B.: Opinion spam and analysis. In: Proceedings of the 2008 International Conference on Web Search and Data Mining, pp. 219–230 (2008).

[2] Ott, M., Choi, Y., Cardie, C., Hancock, J.T.: Finding deceptive opinion spam by any means necessary. In: Proceedings of the 49th Annual Meeting of the Association for Com-putational Linguistics, pp. 309–319 (2011).

[3] Kim, Y.: Convolutional neural networks for sentence classification. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1746–1751 (2014).

[4] Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Computation 9(8), 1735–1780 (1997).

[5] Hooi, B., Song, H.A., Beutel, A., Shah, N., Shin, K., Faloutsos, C.: Fraudar: Bounding graph fraud in the face of adversarial camouflaging. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 895–904 (2016).

[6] Mukherjee, A., Venkataraman, V., Liu, B., Glance, N.: Spotting fake reviewer groups in consumer reviews. In: Proceedings of the 22nd International Conference on World Wide Web, pp. 191–202 (2013).

[7] Hutto, C.J., Gilbert, E.: VADER: A parsimonious rule-based model for sentiment analysis of social media text. In: Proceedings of the Eighth International AAAI Conference on Weblogs and Social Media, pp. 216–225 (2014).

[8] Loria, S.: TextBlob: Simplified text processing. secondary TextBlob: Simplified Text Pro-cessing (2013).

[9] Rayana, S., Akoglu, L.: Collective opinion spam detection: Bridging theory and prac-tice. In: Proceedings of the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 985–994 (2015).

[10] Li, F., Huang, M., Yang, Y., Zhu, X.: Towards a general framework for deceptive opinion spam detection. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 590–600 (2014).