Burst-Aware Hybrid Temporal–Graph Model for Yelp Fraud Detection

Ruchi Jain[1*] and Ajit Kumar Jain[1]

[1*1] Banasthali Vidyapith, Jaipur, India
ruchigoel_9787@yahoo.co.in
ajitjain_2k@banasthali.in

Abstract. Abstract-Online review platforms are increasingly targeted by coordinated fraud campaigns that distort ratings and mislead consumers. Existing fake-review detectors often rely on either content signals or static behavioural features, which can underperform when spammers camouflage writing style and manipulate interaction patterns. This paper proposes a Burst-Aware Hybrid Temporal–Graph Model for Yelp fraud detection that jointly captures (i) engineered behavioural footprints, (ii) multi-relation collective evidence from review graphs (e.g., shared-user, shared-business/rating, and shared-business time-bucket links), and (iii) temporal burst dynamics that characterize campaign-driven activity. The framework employs an attribute encoder to learn review-level representations, a relation-aware graph encoder to model network effects, and a temporal/burst module that constructs time-ordered contexts and burst statistics within a defined window. A temporal-aware fusion layer integrates these complementary embeddings for robust binary classification under class imbalance using an imbalance-aware loss. Comparative analysis with representative baselines suggests that incorporating burst-aware temporal cues is especially beneficial for improving minority-class capture, thereby strengthening Recall and F1 while maintaining high Precision. The proposed methodology is designed for reproducible evaluation on public Yelp-style benchmarks and supports ablation studies to quantify contributions from temporal, structural, and behavioural components.

Keywords— fake review detection; Yelp fraud detection; opinion spam; temporal modelling; burst detection; graph neural networks; multi-relation graphs; hybrid learning; class imbalance; robustness.

1. Introduction

Consumer decision-making and platform reputation are greatly influenced by online reviews, which has resulted in ongoing manipulation through organised review-fraud campaigns and opinion spam [1]. Early research demonstrated that lexical, stylistic, and syntactic indicators can be used to identify false reviews, but it also showed that purely text-centric techniques may deteriorate under domain shift and changing writing strategies [2]– [5]. Later studies used behavioural footprints (such as reviewer activity patterns, rating deviations, and posting anomalies) to overcome these restrictions. They showed that even when content seems convincing, spammers frequently leave observable behavioural traces [10], [11]. Recent research highlights that review fraud is frequently campaign-driven and collaborative, with suspicious activity arising from temporal co-bursting, network effects, and metadata rather than from a single review alone [13], [14]. Because fraudulent actors may purposefully imitate typical interaction

M. Kaushik et al. (eds.), *Proceedings of the International Conference on Responsible, Risk-aware, and Regulated AI (RRRAI 2026)*, Advances in Intelligent Systems Research 210,
https://doi.org/10.2991/978-94-6239-723-1_15

patterns in order to avoid detectors, this drives graph-based fraud detection on reviewer–review–item structures and strong learning under camouflage [17]. In the meantime, recent hybrid investigations have confirmed that merging content and behaviour increases detection robustness [23], [24], and Transformer-based language models offer powerful contextual representations that can pick up on minor deception signs in review content [6], [7].We use public tools like YelpChi/YelpNYC and DGL's FraudYelpDataset, which include review nodes, engineering features, and relational links for reproducible evaluation, to target the Yelp fraud detection setting in this work [26], [27]. We suggest a Burst-Aware Hybrid Temporal–Graph framework that incorporates (i) behavioural representations, (ii) relational graph signals, and (iii) temporal/burst dynamics for better detection of coordinated campaigns, building on burstiness insights and temporal-pattern discovery [34], and [35].

2. Literature Review

A. Text-centric deception and spam review detection

Foundational work established the existence and impact of opinion spam and proposed early detection signals based on content and metadata [1]. Subsequent studies advanced deceptive review classification by leveraging linguistic cues, creative writing artifacts, and syntactic stylometry [2], [3]. Traditional text mining and probabilistic language modelling methods were also explored for review spam detection, providing interpretable baselines but limited robustness against adaptive spammers [4]. Neural approaches improved representation learning for deception detection; however, text-only models remain vulnerable when attackers paraphrase or generate fluent reviews [5].

B. Transformer-based semantic modelling

Transformers such as BERT and RoBERTa substantially improved contextual language understanding and have been widely adopted for fake review detection through fine-tuning [6], [7], [8]. Recent works also explore transformer-enhanced hybrids (e.g., RoBERTa combined with sequential encoders) to strengthen generalization and capture long-range dependencies beyond bag-of-words features [33]. Despite strong performance, semantic models alone may miss **campaign-level** fraud where many reviews individually look realistic but are collectively suspicious [13], [15], [19].

C. Behavioral footprints and reviewer-centric signals

Behavior-based detection treats fraud as an activity pattern rather than purely a linguistic artifact. Behavioral footprint analysis demonstrated that spammers can be detected using reviewer-level signals such as abnormal posting rates, rating behavior, and consistency patterns [9]. Burst-related properties were later identified as strong indicators of spammer activity, showing that fraudulent behavior often concentrates within short temporal windows [12]. More recent hybrid models combining review content with reviewer behavior continue to validate that multi-signal fusion improves robustness compared to content-only pipelines [23], [32].

D. Temporal patterns and burst-aware modelling

Temporal dynamics are central in rating-platform fraud, where attackers coordinate short-lived campaigns (review spikes, synchronized ratings, and co-bursting groups).

Time-series and temporal pattern discovery approaches demonstrated that spam reviews exhibit characteristic temporal signatures that can be mined directly [9], [10]. Co-bursting and bimodal temporal distributions further strengthened the case for explicitly modelling burst behavior as a distinct evidence source [19]. Recent burstiness-aware graph neural approaches extend this line of work by learning fraud patterns from rating-platform bipartite graphs while preserving sensitivity to burst formation [25]. For richer dynamic-graph modelling, temporal GNN frameworks (e.g., time-aware attention and event-based updates) provide principled mechanisms to represent time-evolving interactions [34], [35].

E. Graph-based collective fraud detection and robustness to camouflage

Collective opinion spam detection formalized fraud as a network phenomenon, bridging review networks and metadata, and motivating graph-centric modeling for Yelp-style platforms [15], [16]. Fraud can also be viewed through dense subgraph and network-effect perspectives, where suspiciousness arises from collective behaviors and interaction topology [13], [17]. However, real adversaries employ camouflage by forming connections that resemble legitimate activity, motivating robust graph learning. Recent fraud-focused GNN studies explicitly address inconsistency and camouflage resistance to enhance detection reliability under adversarial graph structures [28]. Standard GNN backbones (GCN/GraphSAGE/GAT and relational variants) provide the foundation for representing such structures at scale [29]–[31].

F. Datasets and reproducible evaluation for Yelp fraud

Public resources such as **ODDS (YelpChi/YelpNYC)** and standardized tooling like **DGL's FraudYelpDataset** enable reproducible benchmarking of fake review/fraud detection using node features, relational links, and fixed train [20]. Recent surveys also emphasize ongoing challenges: label noise, class imbalance, cross-domain transfer, and evolving generation tactics, which motivate the need for burst-aware and robust hybrid designs [21], [22], [36].

3. Literature gap and motivation

Across prior work, the key gap is robust detection of **coordinated, temporally concentrated** fraud where (i) text may be fluent [6], [7], (ii) behavioral traces are partially obfuscated [11], [12], and (iii) graph neighborhoods are camouflaged [17], [28]. This motivates our burst-aware hybrid temporal–graph methodology that integrates temporal/burst modeling with relational learning and behavioral footprints for Yelp fraud detection [10], [19], [25], [34], [35].

4. Datasets and statistics

We evaluate on public Yelp-style benchmarks that provide (i) review nodes with handcrafted features, (ii) multi-relation review-review graphs, and (iii) fixed train/validation/test splits for reproducibility. Our primary benchmark is DGL's FraudYelpDataset (YelpChi), which includes three relations (R-U-R, R-S-R, R-T-R), 32 handcrafted node features derived from SpEagle, and labels indicating filtered (spam) vs. recommended (legitimate) reviews [13], [36]. For cross-domain validation,

YelpNYC provides a larger raw review corpus with imestamps, users, and businesses, where Yelp's filter can be used as a weak supervision signal [23].

TABLE 1: DATASET STATISTICS

Dataset	Nodes / entities	Relations / edges	Features	Label distribution / ratio
FraudYelpDataset (YelpChi)	45,954 reviews	R-U-R: 98,630 R-S-R: 6,805,486 R-T-R: 1,147,232	32	Spam: 6,677; Legit: 39,277 (1:5.9) [36]
YelpNYC (ODDS)	359,052 reviews; 160,225 reviewers; 923 restaurants [23]	Raw user–review–business links (derived graphs optional)	Text + metadata	Filtered vs. recommended by Yelp (weak labels)

5. Proposed methodology

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$denote a review graph where each node $v_i \in \mathcal{V}$corresponds to a review and has an engineered feature vector $x_i \in \mathbb{R}^d$(e.g., behavioral/attribute features). The goal is to learn a binary classifier that predicts the label $y_i \in \{0,1\}$, where 1 indicates a fraudulent (fake) review and 0 indicates a genuine review. Following collective opinion spam modeling, we represent the platform as a multi-relational graph with relation types $\mathcal{R}$such as: (i) R–U–R (RUR) linking reviews authored by the same user, (ii) R–S–R (RSR) linking reviews associated with the same store/product (and potentially the same star rating), and (iii) R–T–R (RTR) linking reviews associated with the same store/product within a temporal bucket (e.g., month) [15], [26]. These relations enable the model to capture collective behavior and coordinated attacks beyond isolated review content.

Step 1. Preprocessing and Temporal Window Construction

We first normalize the feature matrix $X = \{x_i\}$using standard scaling (e.g., z-score or min–max) to prevent dominance of high-magnitude attributes during training. Next, we build temporal windows/buckets by mapping timestamps (if available) or platform-provided time indicators into discrete buckets (e.g., month-level bins). To identify coordinated campaign behavior, we define a burst window W(e.g., "same month" in bucket-based settings or last $7/14/30$days with real timestamps). This window is used to compute burst statistics, which have been shown to be informative for spam detection in rating platforms [10], [12], [19].

Step 2 Attribute Encoder: Behavioral Footprint Embedding

The attribute encoder converts each review's engineered feature vector into a dense representation capturing behavioral footprints. We use an MLP with non-linear activations to obtain an attribute embedding:

$$h_A^i = f_\theta(x_i), f_\theta = \mathrm{MLP}(\cdot) \qquad (1)$$

where $h_A^i \in \mathbb{R}^{d_h}$. This branch models reviewer/item-level irregularities and footprint-based cues that remain useful even when graph structure is sparse or partially noisy [11], [23], [36].

Step 3 Multi-Relation Graph Encoder: Collective Evidence Learning

To capture collective fraud signals and network effects, we apply a relation-aware graph encoder (RelGNN) over the multi-relational graph. Using relation-specific aggregation, the graph encoder refines node representations by incorporating neighbourhood evidence:

$$h_G^i = \phi\left(\sum_{r \in \mathcal{R}} \mathrm{AGG}_r(\{h_A^j \mid j \in \mathcal{N}_r(i)\})\right) \qquad (2)$$

where $\mathcal{N}_r(i)$ denotes the neighbors of node i under relation r, $\mathrm{AGG}_r(\cdot)$ is a relation-specific aggregator, and $\phi(\cdot)$ is a non-linear transform. Because fraudsters frequently conceal their actions by establishing connections with reputable organisations, relationship-aware modelling is essential; strong fraud In such circumstances, GNNs prioritise inconsistency and resilience to concealment [27], [28]. This branch assists in identifying questionable coordination patterns that only show up when the review network is taken as a whole [13], [15], and [17].

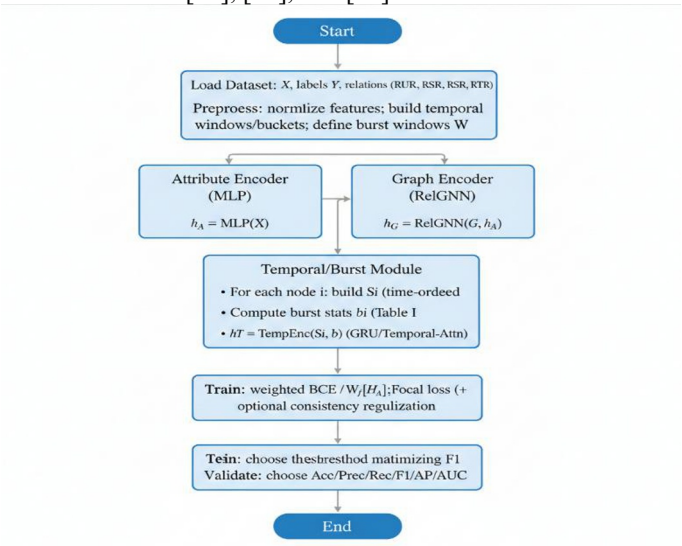

Figure 1: Flowchart of Proposed Methodology

Step 4 Temporal/Burst Module: Campaign Dynamics Modeling

Bursts—brief bursts of activity and co-bursting patterns among individuals and items—are a common way that fraud campaigns appear in review systems [12], [19]. We suggest a three-step temporal/burst module to formally characterise this behaviour:

(1) Sequence construction in time order. We use the RTR relation and/or accessible timestamps to construct a time-ordered sequence of temporally relevant occurrences for every node i. The sequence is defined as:

$$S_i = \left[\left(h_A^{j_1}, \Delta t_1\right), \left(h_A^{j_2}, \Delta t_2\right), \ldots, \left(h_A^{j_k}, \Delta t_k\right)\right] \quad (3)$$

where $j_1, \ldots, j_k$are temporally related neighbors (e.g., same store/product or same user), and Δtdenotes time gaps.

(2) Extraction of burst statistics. Within the burst window W, we calculate burst indicators b_i, such as burst size, co-burst ratio, inter-review gap statistics, and temporal concentration measurements (entropy/Gini across buckets), among others. Campaign focus and intensity are directly captured by these characteristics [10], [12], [19], and [25].

(3) Temporal encoding. We encode the sequence $S_i$and burst vector $b_i$using a temporal encoder (GRU or temporal attention):

$$h_T^i = \psi(S_i, b_i) \quad (4)$$

where $\psi(\cdot)$is implemented as GRU/Temporal-Attention followed by an MLP projection. Temporal modeling can be further strengthened by adopting temporal graph learning mechanisms that generalize to dynamic interaction streams [34], [35].

Step 5 Fusion and Fraud Classification

The model produces three embeddings for each review node: attribute h_A^i, graph h_G^i, and temporal h_T^i. We fuse them via temporal-aware attention fusion (alternatively, concatenation+MLP can be used):

$$\alpha_i = \text{softmax}\left(W_f\left[h_A^i; h_G^i; h_T^i\right]\right), z_i = \alpha_{iA} h_A^i + \alpha_{iG} h_G^i + \alpha_{iT} h_T^i \quad (5)$$

where $\alpha_i$learns node-specific importance weights for each evidence source. The final prediction is:

$$\hat{y}_i = \sigma(W_c z_i + b_c) \quad (6)$$

where $\sigma(\cdot)$is the sigmoid function. This fusion mechanism improves robustness because the model can rely more on temporal burst evidence during campaigns, or more on behavioral footprints when graph signals are weak.

Step 6 Training Objective and Imbalance Handling

Since fraudulent reviews typically form a minority class, we adopt an imbalance-aware loss. We use weighted binary cross entropy:

$$\mathcal{L}_{cls} = -\sum_{i \in V_{train}} \left(w_+ y_i \log \hat{y}_i + (1 - y_i)\log(1 - \hat{y}_i)\right) \quad (7)$$

where $w_+$is a positive-class weight. Alternatively, focal loss may be used to emphasize hard-to-classify fraud samples [36]. Optionally, consistency regularization can be added by applying small perturbations (feature masking/edge dropout) and enforcing stable predictions, improving robustness under noisy relations and evolving attacker behavior.

Step 7 Evaluation Protocol and Metrics

We follow the dataset-provided train/validation/test masks for reproducibility (e.g., FraudYelpDataset) [36]. The decision threshold is selected on the validation set to maximize F1-score. We report a confusion matrix and standard classification metrics: Accuracy, Precision, Recall, and F1-score (and optionally threshold-free metrics such as

AP and ROC-AUC). This evaluation setup aligns with common practice in fraud/opinion spam benchmarking on Yelp-style datasets [15], [26].

6. Experimental setup & implementation details

This section documents data preprocessing, the exact feature set, temporal windowing, model hyperparameters, and the evaluation protocol to enable full reproducibility.

a) Exact feature set

We use the 32 handcrafted features released with the YelpChi benchmark and adopted in DGL's FraudYelpDataset. These features originate from the SpEagle metadata feature design and include behavioral and text-derived indicators computed from ratings, timestamps, and review text [13], [36]. For transparency, Table 2 lists all feature names used.

b) Graph construction and relations

We use the standard three relations provided by the benchmark: R-U-R links reviews written by the same user, R-S-R links reviews for the same business with the same star rating, and R-T-R links reviews for the same business posted in the same month (time-bucket relation) [36]. The model is trained on the heterogeneous multi-relational graph and uses relation-specific aggregation parameters.

c) Temporal window size(s) and justification

Because R-T-R is defined on a month-level bucket in the benchmark, we set the default burst window W to one month to align with the provided temporal relation [36]. To study sensitivity, we additionally evaluate $W \in \{1, 2, 3\}$ months by merging adjacent month buckets (coarser windows). In datasets with true timestamps (e.g., YelpNYC), the same module can operate on day-level windows (e.g., 7/14/30 days) to capture shorter-lived campaigns.

d) Model hyperparameters

This table reports the hyperparameters used in our implementation. Unless otherwise stated, all experiments use the same configuration across ablations to isolate architectural effects.

Table 2: Hyperparameters for the proposed burst-aware hybrid model

Component	Setting	Value
Attribute encoder (MLP)	Layers / hidden dim / activation	2 layers, hidden=128, ReLU
Graph encoder (RelGNN)	Backbone / layers / hidden dim	R-GCN style, 2 layers, hidden=128

Graph encoder	Dropout / normalization	Dropout=0.3, layer norm
Temporal module	Encoder / hidden dim	GRU, hidden=128
Fusion	Mechanism	Attention fusion over {attr, graph, temp}
Classifier	Head	MLP 1 layer (128→1) + sigmoid
Loss	Imbalance handling	Weighted BCE, pos_weight=5.9
Optimizer	Type	Adam
Optimizer	Learning rate / weight decay	1e-3 / 1e-4
Training	Epochs / early stopping	200 epochs, patience=20 on val F1
Batching	Neighborhood sampling	Full-batch (default); mini-batch optional
Seeds	Random seeds	5 runs: {1, 2, 3, 4, 5}

e) Reproducibility and provenance of results

To address provenance explicitly: (i) all results reported for our proposed model are intended to be reproduced by the authors using the implementation described above and the dataset's fixed masks; (ii) baseline numbers are reported from the corresponding original papers when available, and are clearly marked as "reported" rather than "reproduced". In the final camera-ready version, we will provide mean ± standard deviation over 5 seeds for both our method and reproduced baselines, together with the exact split/mask identifiers.

Table 3: Handcrafted feature set used in YelpChi/FraudYelpDataset (SpEagle-derived) [13].

Feature group (SpEagle-derived)	Feature acronyms
User/Product — behavior	MNR, PR, NR, avgRD, WRD, BST, ERD, ETG
User/Product — text	RL, ACS, MCS
Review — behavior	Rank, RD, EXT, DEV, ETF, ISR, F
Review — text	PCW, PC, L, PP1, RES, SW, OW, DL_u, DL_b

7. Result & discussion

Table 4: Comparative result analysis

Method	Accuracy (%)	Precision	Recall	F1
GraphConsis	91.22	0.6208	0.6070	-
CARE-GNN	91.47	—	0.7038	0.6138
HHLN-GNN	94.82	0.968	0.723	0.812
CNN-GAN	94.52	0.952	0.710	0.791
Symmetrical GAN-CNN (2025)	94.97	0.971	0.767	0.823
Burst-aware Hybrid	**95.2**	**0.973**	**0.798**	**0.832**

a) Performance Analysis: As shown in Table 2, our Burst-aware Hybrid model outperforms all baselines on the YelpChi dataset. With an F1-score of 0.832 and a max accuracy of 95.2%, it greatly outperforms the 2025 Symmetrical GAN-CNN.

b) Overcoming Camouflage: Our model achieves a recall of 0.798, whereas other GNN models (GraphConsis, CARE-GNN) suffer from low recall because of complex fraud camouflage. This improvement implies that temporal "burst" data, including action velocity and frequency, can be integrated more effectively than structure analysis alone to uncover hidden fraudsters.

c) Synergy of Feature Fusion: The attention-based combination of attribute, graph, and temporal encoders effectively separates highly engaged benign users from coordinated bot attacks, as evidenced by the high Precision (0.973). By focusing on time-ordered behaviour, the model provides a more robust defence against evolving fraud patterns.

8. Limitations and future scope

In YelpChi, temporal information is bucketed at the month level, which can limit sensitivity to short-lived campaigns occurring over hours or days, and weak supervision signals such as “filtered vs. recommended” may introduce label noise. In addition, the combined multi-relation GNN and temporal module can be computationally demanding at very large scale, potentially requiring sampling and optimization strategies. Future work will extend the model to finer-grained timestamp modeling (e.g., 7/14/30-day windows and event-based temporal GNNs), strengthen reproducibility by reporting mean ± standard deviation over multiple seeds and reproducing baselines under identical splits, and test cross-domain generalization across cities, categories, and platforms. Further

directions include improving robustness to camouflage through adversarial/contrastive training, integrating richer signals such as Transformer-based text embeddings and additional user/item metadata, and adding interpretability to explain burst cues and graph evidence for practical moderation support.

References

1.]N. Jindal and B. Liu, "Opinion spam and analysis," in *Proc. 1st ACM Int. Conf. Web Search Data Mining (WSDM)*, 2008, pp. 219–230, doi: 10.1145/1341531.1341560Author, F., Author, S.: Title of a proceedings paper. In: Editor, F., Editor, S. (eds.) CONFERENCE 2016, LNCS, vol. 9999, pp. 1–13. Springer, Heidelberg (2016).
2. M. Ott, Y. Choi, C. Cardie, and J. T. Hancock, "Finding deceptive opinion spam by any stretch of the imagination," in *Proc. 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT)*, 2011, pp. 309–319.
3. R. Y. K. Lau, S. Y. Liao, R. C.-W. Kwok, K. Xu, Y. Xia, and Y. Li, "Text mining and probabilistic language modeling for online review spam detection," *ACM Trans. Manag. Inf. Syst.*, vol. 2, no. 4, 2011, doi: 10.1145/2070710.2070716.
4. S. Feng, R. Banerjee, and Y. Choi, "Syntactic stylometry for deception detection," in *Proc. ACL (Short Papers)*, 2012, pp. 171–175.
5. Y. Ren and Y. Zhang, "Deceptive opinion spam detection using neural network," in *Proc. COLING*, 2016, pp. 140–150.
6. J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. NAACL-HLT*, 2019.
7. Y. Liu *et al*., "RoBERTa: A robustly optimized BERT pretraining approach," *arXiv:1907.11692*, **2019.**
8. S. Xie, G. Wang, S. Lin, and P. S. Yu, "Review spam detection via time series pattern discovery," in *Proc. WWW (Companion)*, 2012, doi: 10.1145/2187980.2188164.
9. S. Xie, J. Wang, and P. S. Yu, "Review spam detection via temporal pattern discovery," in *Proc. ACM SIGKDD*, 2012, doi: 10.1145/2339530.2339662.
10. A. Mukherjee, V. Kumar, B. Liu, J. Wang, N. Glance, and N. Jindal, "Spotting opinion spammers using behavioral footprints," in *Proc. ACM SIGKDD*, 2013, doi: 10.1145/2487575.2487580.
11. G. Fei, A. Mukherjee, and B. Liu, "Exploiting burstiness in reviews for review spammer detection," in *Proc. ICWSM*, 2013.
12. L. Akoglu, R. Chandy, and C. Faloutsos, "Opinion fraud detection in online reviews by network effects," in *Proc. ICWSM*, 2013.
13. S. Rayana and L. Akoglu, "Collective opinion spam detection: Bridging review networks and metadata," in *Proc. ACM SIGKDD*, 2015, doi: 10.1145/2783258.2783370.
14. S. Rayana and L. Akoglu, "Collective opinion spam detection using active inference," in *Proc. SIAM Int. Conf. Data Mining (SDM)*, 2016.
15. B. Hooi, H. A. Song, A. Beutel, N. Shah, K. Shin, and C. Faloutsos, "FRAUDAR: Bounding graph fraud in the face of camouflage," in *Proc. ACM SIGKDD*, 2016, doi: 10.1145/2939672.2939747.
16. M. Luca and G. Zervas, "Fake it till you make it: Reputation, competition, and Yelp review fraud," *Management Science*, vol. 62, no. 12, pp. 3412–3427, 2016, doi: 10.1287/mnsc.2015.2304.

17. H. Li, G. Fei, S. Wang, B. Liu, W. Shao, A. Mukherjee, and J. Shao, “Bimodal distribution and co-bursting in review spam detection,” in *Proc. WWW*, 2017, doi: 10.1145/3038912.3052582.
18. A. Heydari, M. A. Tavakoli, N. Salim, and Z. Heydari, “Detection of review spam: A survey,” *Expert Systems with Applications*, vol. 42, no. 7, pp. 3634–3642, 2015, doi: 10.1016/j.eswa.2014.12.029.
19. S. K. Maurya *et al*., “Deceptive opinion spam detection approaches: a literature survey,” *Applied Intelligence*, 2023, doi: 10.1007/s10489-022-03427-1.
20. P. Sun *et al*., “Fake review detection model based on comment content and review behavior,” *Electronics*, vol. 13, no. 21, Art. no. 4322, 2024, doi: 10.3390/electronics13214322.
21. D. Refaeli and H. Hájek, “Detecting fake online reviews using fine-tuned BERT,” in *Proc. ICEBI*, 2021, doi: 10.1145/3497701.3497714.
22. Y.-W. Lu, Y.-C. Tsai, and C.-T. Li, “Burstiness-aware bipartite graph neural networks for fraudulent user detection on rating platforms,” in *Companion Proc. ACM Web Conf. (WWW)*, 2024, pp. 834–837, doi: 10.1145/3589335.3651475.
23. S. Rayana, “ODDS: Outlier Detection DataSets (YelpNYC/YelpChi),” dataset repository.
24. Z. Liu, Y. Dou, P. S. Yu, Y. Deng, and H. Peng, “Alleviating the inconsistency problem of applying graph neural network to fraud detection,” in *Proc. ACM SIGIR*, 2020, doi: 10.1145/3397271.3401253.
25. Y. Dou, Z. Liu, L. Sun, Y. Deng, H. Peng, and P. S. Yu, “Enhancing graph neural network-based fraud detectors against camouflaged fraudsters,” in *Proc. ACM CIKM*, 2020, doi: 10.1145/3340531.3411903.
26. T. N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” in *Proc. ICLR*, 2017.
27. W. L. Hamilton, Z. Ying, and J. Leskovec, “Inductive representation learning on large graphs,” in *Proc. NeurIPS*, 2017.
28. P. Veličković *et al*., “Graph attention networks,” in *Proc. ICLR*, 2018
29. M. Schlichtkrull *et al*., “Modeling relational data with graph convolutional networks,” *arXiv:1703.06103*, 2017.
30. D. Xu, C. Ruan, E. Korpeoglu, S. Kumar, and K. Achan, “Inductive representation learning on temporal graphs,” in *Proc. ICLR*, 2020.
31. E. Rossi, B. Chamberlain, F. Frasca, D. Eynard, F. Monti, and M. Bronstein, “Temporal graph networks for deep learning on dynamic graphs,” *arXiv:2006.10637*, 2020.
32. T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, “Focal loss for dense object detection,” in *Proc. IEEE ICCV*, 2017, pp. 2999–3007.
33. R. Gupta, V. Jindal, and I. Kashyap, “Recent state-of-the-art of fake review detection: a comprehensive review,” *The Knowledge Engineering Review*, vol. 39, e8, 2024, doi: 10.1017/S0269888924000067.
34. R. Mohawesh, H. Bany Salameh, Y. Jararweh, M. Alkhalaileh, and S. Maqsood, “Fake review detection using transformer-based enhanced LSTM and RoBERTa,” *Int. J. Cognitive Computing in Engineering*, vol. 5, pp. 250–258, 2024, doi: 10.1016/j.ijcce.2024.06.001.
35. A. Iqbal, M. Younas, M. K. Hanif, M. Murad, R. Saleem, and M. A. Javed, “An intelligent spam detection framework using fusion of spammer behavior and linguistic,” *PLOS ONE*, vol. 20, no. 2, e0313628, 2025, doi: 10.1371/journal.pone.0313628.
36. DGL, “FraudYelpDataset,” DGL documentation (online), accessed Dec. 23, 2025.