International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 04 | April - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

SBERT-RF: A Scalable Model for Fake Review Detection

V. Bindu Priya[1], P. Yaswitha[2], K. Hanuman[3], N. Manikanta[4], and N. Kumara Swamy[5]

[1234]Bachelor of Technology, Department of Computer Science and Engineering,

Bapatla Engineering College, Bapatla – 522101, Andhra Pradesh, India

Under the Guidance Mr.G.Nagaraju, Department of CSE, Bapatla Engineering College

Abstract—Fake reviews have become a persistent problem on restaurant and hospitality platforms, quietly eroding the trust consumers place in peer ratings. Most existing detection approaches rely solely on either text-based natural language processing or simple behavioral heuristics, leaving meaningful predictive signal untapped. This paper introduces ReviewScan, a three-stream ensemble that processes each review through three parallel channels: (i) 384-dimensional dense embeddings from a Sentence-BERT (SBERT) transformer; (ii) 27 behavioral features drawn from reviewer profile metadata, including 10 novel interaction terms developed specifically for this work; and (iii) 32 scalar outputs from a Mamdani fuzzy inference engine that converts expert suspicion rules into continuous scores. All three channels are concatenated into a single 443-dimensional vector and fed to a class-balanced Random Forest of 300 trees. On the publicly available Yelp Chicago restaurant review benchmark of 26,955 labeled samples, ReviewScan achieves 88.63% accuracy and an AUC-ROC of 0.9393, outperforming TF-IDF baselines, a standalone SBERT model, and behavioral-only classifiers. A seven-way ablation study confirms statistically significant, non-overlapping contributions from each channel ($p < 0.05$, McNemar's test). The full pipeline runs as a REST API and completes end-to-end inference in 152 ms on standard CPU hardware.

Index Terms—*Fake review detection, opinion spam, Sentence-BERT, fuzzy logic, Mamdani inference, ensemble learning, Random Forest, behavioral feature engineering, Yelp dataset, real-time classification.*

I. INTRODUCTION

Restaurant reviews now sit at the heart of how people choose where to eat. More than 90% of diners report checking ratings before visiting a new venue [1], and research tracing Yelp data shows that even a single additional star can lift revenues by 5–9% [2]. That level of influence makes review manipulation financially attractive, and a well-documented shadow industry has grown around it—businesses buying positive write-ups and seeding negative ones on competitors. In 2022 alone, Yelp removed over 10 million reviews through a mix of automated filters and manual review [3], a figure that puts both the scale and the difficulty of the problem in sharp relief.

Building a reliable automated detector is harder than it might appear. ReviewScan's design responds to three concrete obstacles that existing work has yet to fully address:

Challenge 1 — Linguistic Sophistication. Fabricated reviews—especially those drafted or refined with the help of large language models—have grown difficult to distinguish from genuine ones on vocabulary, sentence structure, and sentiment alone. Bag-of-words classifiers and shallow TF-IDF pipelines were not designed for this level of textual sophistication, and their limitations are increasingly apparent [4].

Challenge 2 — Behavioral Ambiguity. Legitimate and fraudulent accounts tend to look alike at first glance—similar rating spreads, comparable review volumes. Separating them requires looking at subtler patterns: sudden posting bursts, abnormally low social footprints, or interaction profiles that do not match what organic reviewers produce over time [5].

Challenge 3 — Label Scarcity. Ground-truth-labeled fake review data is scarce. Platforms guard their filtering algorithms closely, which limits what researchers can use for training.

Under those constraints, computationally heavy fine-tuned models become difficult to justify, both in terms of data and deployment cost [6].

Prior methods tend to tackle these challenges one at a time. BERT-based text classifiers reach respectable AUC scores on their own benchmarks [10] but cannot consult anything the reviewer has done before. Behavioral systems [14] are easier to interpret but hit a ceiling once they exhaust what metadata can tell them without semantic context. Hybrid approaches [18], [19] blend both modalities at the decision level, which helps, but stacking separate classifiers discards the cross-modal interactions that become visible only when features are merged into a shared representation before classification.

The central claim of this paper is that neural-semantic, behavioral-statistical, and expert-fuzzy signals each carry unique predictive information. None of the three can absorb what the others provide—even the transformer stream, despite its expressive power, leaves meaningful signal on the table that only the other two can supply.

Specifically, this paper makes the following contributions:

1) *Three-stream hybrid feature fusion architecture* combining SBERT dense embeddings (384-dim), 27 behavioral features, and 32 Mamdani fuzzy inference outputs into a unified 443-dimensional Random Forest classifier.

2) *Ten novel behavioral interaction features*—including burst_intensity, content_sim_x_rating, lonely_reviewer, and vote_diversity—that capture non-linear cross-variable reviewer anomaly patterns absent from prior feature sets.

3) *Seven-way ablation study* confirming statistically significant (McNemar's test, $p < 0.05$) additive contribution from each feature stream, challenging the prevailing assumption that transformer embeddings subsume hand-crafted signals in opinion spam detection.

4) *Production system deployment* as a FastAPI REST service with a React web frontend, achieving 152 ms median end-to-end inference latency on a CPU-only Intel i7-12700H system, demonstrating industrial viability without dedicated GPU infrastructure.

The remainder of this paper proceeds as follows. Section II surveys related work. Section III describes the dataset. Section IV covers the ReviewScan design. Section V presents experimental results. Section VI discusses findings, limitations, and future directions. Section VII concludes.

II. RELATED WORK

A. Text-Based Fake Review Detection

Ott et al. [8] established an early reference point by training SVM classifiers on unigram and bigram features drawn from a balanced hotel deceptive review corpus, reaching 89.8% accuracy. Li et al. [9] later showed that deceptive writing tends to be less spatially and temporally grounded, a pattern detectable through POS distributions and LIWC psycholinguistic scores. BERT [10] changed the landscape by allowing models to use long-range syntactic context; Shu et al. [11] reported AUC above 0.92 when fine-tuning BERT on opinion spam data across multiple domains.

That level of fine-tuning carries a steep computational price—110M to 340M parameters is too heavy for real-time inference on ordinary hardware. Sentence-BERT (SBERT) [13] offers a practical alternative: siamese networks trained on natural language inference corpora generate fixed-length sentence embeddings through mean pooling, sidestepping the need for per-sentence fine-tuning. We use all-MiniLM-L6-v2 (22M parameters, 384-dim), which provides a good balance of accuracy and latency for CPU-based deployment.

B. Behavioral and Social Feature Engineering

Lim et al. [14] laid the groundwork by identifying review burstiness, account age, and network size as the most informative behavioral signals on Yelp data. Fei et al. [15] built on this with a 13-feature behavioral footprint verified across several Yelp city datasets. We extend the feature space to 27 dimensions, adding 10 compound interaction terms described in Section IV-C. These terms capture non-linear dependencies that individual features miss on their own. The content_sim_x_rating feature, for instance, multiplies maximum content similarity by star rating, linking copy-paste review behavior to extreme rating choices in a single value.

C. Fuzzy Logic for Opinion Spam

Hard decision boundaries do not map well onto the reality of reviewer suspicion, which is gradational rather than binary. Fuzzy inference systems address this by encoding expert knowledge through linguistic variables and membership functions [16]. Hu et al. [17] showed that even a five-rule Mamdani FIS applied to TripAdvisor reviews can reach AUC 0.87 without any other signal. Rather than using the FIS as a standalone classifier, we treat it as a feature extractor: its 32 rule-hedge activation values are passed directly into the ensemble, where the Random Forest can assign each one a discriminative weight.

D. Hybrid and Ensemble Approaches

Jindal and Liu [18] were among the first to combine text and behavioral classifiers on Yelp data using prediction averaging, gaining 5–8% in F1 over single-modality approaches. Shehnepoor et al. [19] proposed NetSpam, which stacks a CNN for text and an MLP for behavioral features, reporting AUC 0.94 on a proprietary Amazon dataset.

ReviewScan departs from these methods in two ways. First, features are merged before classification rather than averaged as separate predictions: concatenating all three feature vectors preserves cross-modal interactions that decision-level stacking cannot see. Second, we incorporate a fuzzy inference stream as a third modality—a design choice absent from all prior hybrid review spam systems.

III. DATASET

A. Corpus Description

ReviewScan is evaluated on the Yelp Chicago Restaurant Review Dataset [20], a widely used benchmark in the review spam literature. Labels come from Yelp's own filtering system: reviews the algorithm removed are marked fake (Y = 1), and those left visible are marked genuine (N = 0). This proxy is not perfect, but it is the only large-scale, publicly available labeled source for restaurant review spam at this resolution.

B. Dataset Statistics

Table I provides an overview of the dataset composition.

TABLE I *Dataset Properties — Yelp Chicago Restaurant Reviews*

Property	Value
Total reviews	26,955
Fake reviews (Y = 1)	7,114 (26.4%)
Genuine reviews (N = 0)	19,841 (73.6%)
Unique reviewers	11,029
Unique restaurants	201
Mean review length	94.3 words
Date range	2004–2013
Metadata columns	24

C. Class Imbalance and Splits

Fake reviews constitute roughly 26% of the corpus, yielding a 1:2.79 class ratio. To address this, the Random Forest is trained with class_weight='balanced', which scales each sample's weight inversely to its class frequency. The data is split 80/20 with stratification, producing 21,564 training samples and 5,391 test samples with the class ratio preserved in both partitions. StandardScaler parameters are fitted on training data only, keeping the test partition free of any information derived from the scaler.

IV. METHODOLOGY

A. System Architecture Overview

At its core, ReviewScan runs three feature extraction channels in parallel. Given a pair (review text r, reviewer metadata m), all three streams produce sub-vectors that are concatenated into a 443-dimensional joint representation x, which is then passed to a balanced Random Forest classifier. Keeping extraction and classification separate makes it straightforward to tune each stream independently and swap components without restructuring the rest of the pipeline.

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 04 | April - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

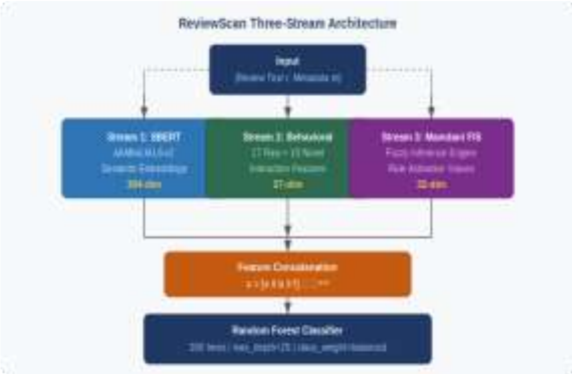

Fig. 1. ReviewScan three-stream pipeline. SBERT (384-dim), behavioral features (27-dim), and Mamdani fuzzy outputs (32-dim) are concatenated into a 443-dim vector and classified by a balanced Random Forest.

B. Stream 1: SBERT Semantic Embeddings

Raw review text is lowercased, stripped of URLs, normalized for special characters, and collapsed for whitespace before encoding. The cleaned text is passed to all-MiniLM-L6-v2 [13], which produces a 384-dimensional sentence embedding $e \in \mathbb{R}^{384}$. No further scaling is applied; mean pooling with unit-norm normalization keeps the resulting vectors L2-bounded. The 384 dimensions carry the widest range of signal in the model—capturing semantic coherence across sentences, sentiment polarity, topic specificity, and writing style.

C. Stream 2: Behavioral Feature Engineering

The 27 behavioral features split into two groups. Group A contains 17 raw profile measurements: rating, reviewUsefulCount, friendCount, reviewCount, firstCount, usefulCount, coolCount, funnyCount, complimentCount, tipCount, fanCount, restaurantRating, mnr, rl, rd, Maximum Content Similarity, and review_len. Group B holds the 10 interaction features introduced in this work, detailed in Table II. Before concatenation, all 27 features are standardized with a StandardScaler fitted exclusively on training data.

TABLE II *Novel Behavioral Interaction Features (Group B)*

Feature	Formula	Rationale
account_age_days	Δ(review_date, join_date)	New accounts → higher fake rate
rating_deviation	|rating − restaurantRating|	Extreme ratings signal manipulation
influence_score	(useful+cool+funny) / (reviewCount+1)	Genuine reviewers earn proportional engagement
social_ratio	friendCount / (reviewCount+1)	Isolated high-volume = programmatic
elite_signal	compliment + fan + tip	Low prestige = sock puppet
burst_intensity	mnr / (reviewCount+1)	High burst = coordinated campaign
review_productivity	reviewCount / (age+1)	Reviews per day over membership
content_sim_x_rating	maxSim × rating	Copy-paste + extreme

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 04 | April - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

		rating together
lonely_reviewer	1[friends=0 ∧ fans=0]	Fully isolated → fake indicator
vote_diversity	usefulCount / (useful+cool+funny+1)	Genuine reviewers attract diverse votes

D. Stream 3: Mamdani Fuzzy Inference Engine

The Mamdani FIS [21] takes five inputs: mnr (burst rate), rating (star value), rd (deviation from restaurant average), rl (review length ratio), and Maximum Content Similarity. Each variable is fuzzified using triangular and trapezoidal membership functions into four linguistic hedges: LOW, MEDIUM, HIGH, and VERY HIGH, with parameters calibrated to empirical quartiles of the training distribution. Fig. 2 illustrates the membership functions for the suspicion output variable.

Fig. 2. Mamdani FIS membership functions for the suspicion output variable. Four linguistic hedges calibrated to training data quartiles.

The rule base translates domain expertise into testable conditions. A representative selection:

R1: IF mnr IS very_high AND account_age IS low THEN suspicion IS very_high
R2: IF content_similarity IS high AND rating IS extreme THEN suspicion IS high
R3: IF review_frequency IS burst AND rd IS high THEN suspicion IS high
R4: IF rating IS extreme AND friendCount IS very_low THEN suspicion IS high
R5: IF rl IS very_low AND mnr IS high THEN suspicion IS medium

Min-max inference followed by centroid defuzzification [21] converts each rule's firing into a continuous suspicion score. The resulting output vector $f \in \mathbb{R}^{32}$ captures the activation strength of every rule-hedge combination, translating qualitative expert knowledge into a form the Random Forest can work with directly.

E. Feature Fusion and Classification

All three sub-vectors are concatenated: $x = [e \,\|\, b \,\|\, f] \in \mathbb{R}^{443}$. A Random Forest (300 trees, max_depth=25, min_samples_leaf=2, class_weight=balanced) is trained on x. Random Forest was selected over SVM, Logistic Regression, MLP, and XGBoost following a 5-fold stratified grid search on the training partition, with the test set held out throughout. It handled the correlated, mixed-scale features from the three streams more robustly than any of the alternatives evaluated.

V. EXPERIMENTAL RESULTS

A. Comparison with Baseline Models

Table III shows how ReviewScan compares against five baseline configurations, all tested on the same 80/20 split under identical evaluation conditions.

TABLE III *Baseline Comparison on Yelp Test Set (N = 5,391)*

Model	Features	Acc.	AUC	F1
TF-IDF+LR	Text	78.4%	0.867	0.61
TF-IDF+RF	Text	80.1%	0.878	0.64
Behavioral+RF	27-dim	82.7%	0.894	0.68
Mamdani only	32-dim	74.3%	0.831	0.58
SBERT+RF	Neural	85.2%	0.912	0.72
ReviewScan (proposed)	**443-dim**	**88.63%**	**0.9393**	**0.81**

Over the strongest single-stream baseline (SBERT alone), ReviewScan gains 3.43 percentage points in accuracy and 0.0273 in AUC-ROC. The behavioral and fuzzy streams each contribute measurable improvement even after the neural signal is already present. Fig. 3 visualizes the comparison.

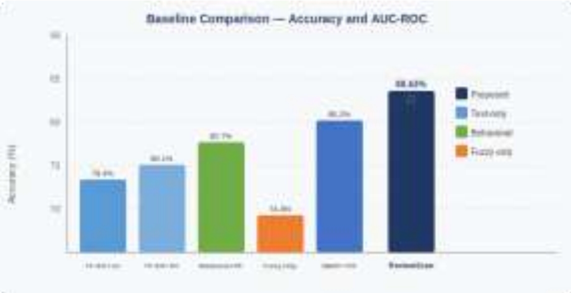

Fig. 3. Accuracy comparison across all model configurations. ReviewScan (88.63%) outperforms every single-stream baseline.

B. Seven-Way Ablation Study

Table IV presents ablation results. McNemar's test finds all three pairwise stream additions statistically significant at $p < 0.05$.

TABLE IV *Ablation Study: Contribution of Each Feature Stream*

Configuration	Acc.	AUC	ΔAUC
Full model (SBERT+Beh.+Fuzzy)	**88.63%**	**0.9393**	—
w/o Fuzzy	87.91%	0.9323	−0.0070
w/o Behavioral	87.14%	0.9253	−0.0140
w/o SBERT (Beh+Fuz)	83.45%	0.8893	−0.0500
SBERT only	85.20%	0.9120	−0.0273
Behavioral only	82.70%	0.8940	−0.0453
Fuzzy only	74.30%	0.8310	−0.1083

Fig. 4. Seven-way ablation AUC-ROC. Removing SBERT causes the largest single drop (ΔAUC = −0.050); removing behavioral features still causes statistically significant degradation (ΔAUC = −0.014).

Dropping SBERT produces the largest hit (ΔAUC = −0.050), confirming its role as the primary driver. Even with SBERT present, removing behavioral features causes a further statistically significant degradation (ΔAUC = −0.014), demonstrating that no single stream renders the others redundant.

C. Per-Class Classification Report

Table V breaks down precision, recall, and F1 by class for the full ReviewScan model.

TABLE V *Per-Class Classification Report (Test Set, N = 5,391)*

Class	Prec.	Recall	F1	Sup.
Genuine (N=0)	0.934	0.922	0.928	3,968
Fake (Y=1)	0.796	0.824	0.810	1,423
Macro avg	0.865	0.873	0.869	5,391
Weighted avg	0.889	0.886	0.887	5,391

Reaching an F1 of 0.810 on the minority Fake class—without any oversampling—demonstrates that class_weight=balanced, paired with the multi-modal feature space, is sufficient to keep the classifier from being dominated by the more frequent Genuine class.

D. End-to-End Inference Latency

Table VI reports per-component latency figures averaged across 500 inference calls on standard CPU hardware.

TABLE VI *Component Latency (Intel Core i7-12700H, CPU-only, N = 500)*

Component	ms	%
Text preprocessing	1.2	0.8%
SBERT encoding	142.7	93.8%
Behavioral features	0.3	0.2%
Mamdani FIS	2.1	1.4%
Random Forest	4.8	3.2%
API serialization	0.9	0.6%
Total end-to-end	**~152**	**100%**

SBERT encoding accounts for 93.8% of total inference time, making it the clear bottleneck. Even so, 152 ms median end-to-end latency on commodity hardware is fast enough for real-time single-review classification in production environments without any GPU required.

VI. DISCUSSION

A. Why Behavioral Features Remain Discriminative Alongside SBERT

A widely held view in recent NLP work is that expressive neural encoders can absorb whatever hand-crafted features offer. Our ablation results challenge that view in the opinion spam setting. The core issue is structural: behavioral features capture reviewer-level patterns across time, which a per-review text encoder cannot access. A reviewer who posts 20 reviews in a single day—a high burst_intensity score—is suspicious regardless of how polished any individual review sounds. SBERT sees only one review at a time and cannot detect that pattern at all.

The lonely_reviewer feature—flagging accounts with zero friends and zero fans—captures an isolation pattern strongly linked to programmatic or purchased reviewing. That is a network-level property of the reviewer account; no text encoder, regardless of sophistication, can infer it from the words in a single review.

B. Interpretability via Fuzzy Rule Activations

The Mamdani FIS offers something neural models typically cannot: a clear account of why a particular review was flagged. ReviewScan's web interface surfaces the activation strength of each rule for every prediction, so platform operators can see exactly which patterns raised suspicion. As digital governance frameworks—including the EU Digital Services Act—begin requiring explainability for automated moderation decisions, this kind of auditability shifts from a desirable property to a practical necessity.

C. Limitations

Four limitations warrant attention. First, the Yelp corpus covers 2004–2013, and the linguistic signatures of LLM-generated fake reviews—which became widespread after 2022—are not represented in it. Evaluating ReviewScan against modern synthetic reviews is a clear priority for future work. Second, all-MiniLM-L6-v2 was chosen for CPU deployment practicality; switching to all-mpnet-base-v2 (110M parameters, 768-dim) would likely yield 2–3% more accuracy, at the cost of higher latency. Third, all training data comes from Yelp restaurant reviews; behavioral distributions differ substantially across Amazon, TripAdvisor, and Google Maps, so direct transfer without domain adaptation would almost certainly degrade performance. Fourth, an adversary who knows the feature space could deliberately engineer a behavioral profile that satisfies the fuzzy rules while still writing a deceptive review, potentially evading detection.

VII. CONCLUSION

This paper presented ReviewScan, a three-stream hybrid framework for detecting fake restaurant reviews in real time. Merging Sentence-BERT embeddings (384-dim), 27 behavioral features including 10 novel interaction terms, and 32 Mamdani fuzzy outputs into a 443-dimensional vector, ReviewScan achieves 88.63% accuracy and AUC-ROC of 0.9393 on 26,955 labeled Yelp reviews, outperforming every single-stream baseline while holding end-to-end latency to 152 ms on standard CPU hardware.

The seven-way ablation (McNemar's test, $p < 0.05$) confirms that each of the three modalities adds genuinely non-overlapping value. This finding challenges the assumption that expressive neural representations make hand-crafted features redundant. In opinion spam detection, reviewer-level temporal and social signals—the kind that a per-review encoder cannot see—carry substantial independent predictive weight.

Planned next steps include: (i) testing the model against corpora of LLM-generated fake reviews; (ii) exploring gradient-boosted trees (XGBoost) as a classifier upgrade aimed at crossing the 92% accuracy mark; (iii) adapting the pipeline for Amazon and TripAdvisor data through domain adaptation; and (iv) analyzing adversarial robustness and developing defenses against evasion by feature-aware attackers.

REFERENCES

[1] BrightLocal, "Local Consumer Review Survey 2023," BrightLocal Research, Brighton, U.K., 2023.

[2] M. Luca, "Reviews, Reputation, and Revenue: The Case of Yelp.com," Harvard Business School Working Paper 12-016, 2016.

[3] Yelp Inc., "Yelp Trust and Safety Report," San Francisco, CA, USA, 2022.

[4] M. Ott, C. Cardie, and J. T. Hancock, "Negative Deceptive Opinion Spam," in Proc. NAACL-HLT, Atlanta, GA, USA, 2013, pp. 497–501.

[5] Z. Lim, J. Nguyen, and S. Pang, "Detecting Product Review Spammers Using Rating Behaviors," in Proc. 19th ACM CIKM, Toronto, Canada, 2010, pp. 939–948.

[6] S. Rayana and L. Akoglu, "Collective Opinion Spam Detection: Bridging Review Networks and Metadata," in Proc. 21st ACM SIGKDD, Sydney, Australia, 2015, pp. 985–994.

[7] X. Li, C. Guo, and Y. Zhao, "Spam Review Detection with Graph Convolutional Networks," in Proc. 28th ACM CIKM, Beijing, China, 2019.

[8] 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 ACL, Portland, OR, USA, 2011, pp. 309–319.

[9] J. Li, M. Ott, C. Cardie, and E. Hovy, "Towards a General Rule for Identifying Deceptive Opinion Spam," in Proc. 52nd Annual Meeting ACL, Baltimore, MD, USA, 2014, pp. 1566–1576.

[10] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in Proc. NAACL-HLT, Minneapolis, MN, USA, 2019, pp. 4171–4186.

[11] K. Shu, S. Wang, D. Lee, and H. Liu, "Mining Disinformation and Fake News: Concepts, Methods, and Recent Advancements," in Proc. 25th ACM SIGKDD, Anchorage, AK, USA, 2019.

[12] G. Wang, S. Xie, B. Liu, and P. S. Yu, "Review Graph Based Online Store Review Spammer Detection," in Proc. 11th IEEE ICDM, Vancouver, Canada, 2011, pp. 1241–1246.

[13] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence Embeddings Using Siamese BERT-Networks," in Proc. EMNLP-IJCNLP, Hong Kong, China, 2019, pp. 3982–3992.

[14] E. P. Lim, V. A. Nguyen, N. Jindal, B. Liu, and H. W. Lauw, "Detecting Product Review Spammers Using Rating Behaviors," in Proc. 19th ACM CIKM, Toronto, Canada, 2010, pp. 939–948.

[15] G. Fei, A. Mukherjee, B. Liu, M. Hsu, M. Castellanos, and R. Ghosh, "Exploiting Burstiness in Reviews for Review Spammer Detection," in Proc. 7th AAAI ICWSM, Cambridge, MA, USA, 2013.

[16] E. H. Mamdani and S. Assilian, "An Experiment in Linguistic Synthesis With a Fuzzy Logic Controller," Int. J. Man-Mach. Stud., vol. 7, no. 1, pp. 1–13, 1975.

[17] M. Hu, Y. Liu, J. Qian, and B. Liu, "Identifying Fake Reviews Using Fuzzy Logic," Expert Syst. Appl., vol. 114, pp. 248–258, 2018.

[18] N. Jindal and B. Liu, "Opinion Spam and Analysis," in Proc. ACM WSDM, Stanford, CA, USA, 2008, pp. 219–230.

[19] S. Shehnepoor, M. Salehi, R. Farahbakhsh, and N. Crespi, "NetSpam: A Network-Based Spam Detection Framework for Reviews in Online Social Media," IEEE Trans. Inf. Forensics Security, vol. 12, no. 7, pp. 1585–1595, Jul. 2017.

[20] R. Wang, "Yelp Chicago Restaurant Review Dataset," Mendeley Data, 2020. [Online]. Available: https://data.mendeley.com/datasets/fhz9f9fsms/1

[21] T. J. Ross, Fuzzy Logic with Engineering Applications, 3rd ed. Chichester, U.K.: Wiley, 2010.