Cross-Platform Generalization in E-Commerce App Sentiment Analysis: A Large-Scale Comparative Study of Classical, Recurrent, and Transformer Architectures

Yash Kumar Arora*, Karan Verma, and Akshay Singh

Department of Computer Science & Engineering, National Institute of Technology Delhi, New Delhi, India
242211022@nitdelhi.ac.in, karanverma@nitdelhi.ac.in, 232210005@nitdelhi.ac.in

Abstract. Sentiment analysis of mobile app reviews presents a challenging natural language processing problem owing to the short, noisy, and linguistically diverse nature of user-generated content. In this paper, we present a large-scale comparative study on approximately 1.35 million English-language reviews scraped from the Google Play Store listings of Amazon India and Flipkart, two of the largest e-commerce platforms in India. We evaluate six classical machine learning models using TF-IDF features alongside four recurrent deep learning architectures LSTM, BiLSTM, GRU, and BiGRU, on binary sentiment classification (positive vs. negative), using Macro F1 as the primary evaluation metric. To investigate cross-platform generalization, models trained on one platform are evaluated on the other without fine-tuning, and additionally benchmarked against a fine-tuned RoBERTa transformer. Our results show that Linear SVM achieves a competitive Macro F1 of 0.9011, approaching the best deep learning model (GRU: 0.9033) at a fraction of the computational cost. Cross-platform transfer reveals a pronounced asymmetry: models trained on Flipkart's data generalize effectively to Amazon's data (GRU $\Delta = +0.0019$), whereas Amazon-trained models suffer significant degradation on Flipkart (GRU $\Delta = -0.0679$). This asymmetry is consistent across all three model families including RoBERTa, demonstrating that source-domain class imbalance, rather than model capacity or pretraining strategy constitutes the primary bottleneck in e-commerce cross-platform sentiment transfer.

Keywords: Sentiment analysis; E-commerce reviews; Cross-platform generalization; LSTM; GRU; RoBERTa; TF-IDF; App store reviews; Domain adaptation

1 Introduction

The rapid expansion of mobile commerce in emerging economies has transformed app store reviews into one of the most valuable sources of consumer intelligence

A. Agnihotri et al. (eds.), *Proceedings of the Conference on Bridging Engineering Disciplines with AI and Machine Learning (BEDAIML 2026)*, Advances in Intelligent Systems Research 209,
https://doi.org/10.2991/978-94-6239-697-5_17

at scale. Amazon and Flipkart help hundreds of millions of people in India. Their reviews on the Google Play Store give in-depth opinions on things like the quality of the products, how reliable the delivery is, how easy it is to pay, and how easy it is to use the app. Automated sentiment analysis helps e-commerce companies keep track of how their brands are doing all the time, figure out which products need to be improved first, and learn about new user complaints before they get worse [1].

It's harder to deal with app store reviews than with formal review standards. They are usually very short (the median is less than 10 words after cleaning), very noisy (with misspellings, informal grammar, emojis, and Hindi-English code-mixing), and mostly positive because of rating self-selection bias. People who take the time to write a review are usually either very happy or very unhappy [17, 2]. These traits call for a strict empirical test: do traditional machine learning pipelines are easy to understand, quick to train, and able to be used without GPU infrastructure can still compete with modern recurrent deep learning architectures that need a lot more computing power?

In addition to in-domain performance, a crucial yet insufficiently examined inquiry is *cross-platform generalization*: can a model trained on Amazon reviews accurately categorize Flipkart reviews, and conversely? Amazon and Flipkart have some of the same users, but they also have very different class imbalance ratios (Amazon has a 1.6:1 positive-to-negative ratio, while Flipkart has a 5.3:1 ratio). They may also use different words, review lengths, and emojis. Understanding these distributional differences is essential for practitioners seeking to deploy a single sentiment model across multiple e-commerce properties without retraining.

This study addresses two aspects: (1) the comparative performance of classical ML and recurrent DL on large-scale noisy app reviews, and (2) the cross-platform generalization capability of these models between Amazon and Flipkart. To answer these questions, we scrape 700,000 reviews from each platform, preprocess and label them with a unified pipeline, and evaluate ten models: six classical ML approaches and four recurrent DL architectures. To further evaluate cross-platform generalization, we additionally fine-tune cardiffnlp/twitter-roberta-base-sentiment-latest [3] as a state-of-the-art transformer baseline. Our key findings are: (i) Linear SVM achieves Macro F1 of 0.9011, within 0.0022 of GRU (0.9033), training 6.6× faster; (ii) GRU is the most efficient DL architecture at 741 s; (iii) cross-platform transfer is highly asymmetric Flipkart → Amazon is near-lossless while Amazon → Flipkart degrades by 0.068–0.080 Macro F1 ; and (iv) this asymmetry persists across SVM, GRU, and RoBERTa.

The main contributions are: (1) a large-scale benchmark of 1.35M e-commerce app reviews across two major Indian platforms; (2) a systematic and reproducible ML-vs-DL comparison under identical experimental conditions; (3) the first cross-platform Amazon–Flipkart generalization study; and (4) empirical evidence that source-domain class imbalance — not model architecture drives cross-platform degradation.

2 Related Work

2.1 Sentiment Analysis: Classical and Deep Learning

Sentiment analysis has been extensively studied, with recent surveys charting the trajectory from lexicon-based methods through classical machine learning to deep neural architectures [1, 17]. Wankhade et al. [17] synthesize findings from over 200 studies and conclude that while neural models dominate standard benchmarks, classical pipelines remain highly competitive where interpretability, training speed, and infrastructure constraints are primary concerns. TF-IDF features combined with linear classifiers, especially SVM and Logistic Regression, always do well on short-text review corpora. Yadav and Vishwakarma [5] say that well-tuned SVM baselines match or beat LSTM when the average review length is less than 20 words, which is exactly what our app-review corpus is like.

On the deep learning side, LSTM and GRU [7] established the dominant recurrent paradigm for sequential text modeling. Dang et al. [9] conducted a systematic comparison of CNN, LSTM, and GRU across multiple sentiment benchmarks and confirmed that GRU achieves accuracy within 0.2–0.5% of LSTM while requiring 20–30% less training time due to its simplified two-gate mechanism, an efficiency advantage we exploit in our architecture selection. However, Zhang et al. [4] observe that the gap between deep learning and well-tuned classical baselines narrows considerably on short, noisy texts where limited sequential context reduces the benefit of recurrent modeling – a finding we investigate at scale on 1.35M app reviews.

2.2 Transformer Models for Sentiment Analysis

The Transformer architecture [10] and the pretraining–fine-tuning paradigm [11] shifted NLP toward transfer learning from large-scale unlabeled corpora. BERT [12] achieved state-of-the-art results on 11 NLP benchmarks; RoBERTa [13] improved further via dynamic masking, larger batches, and more training data. For informal, noisy text, domain-specific pretraining is particularly valuable: Barbieri et al. [3] released a Twitter-pretrained RoBERTa that consistently outperforms generic RoBERTa on social-media tasks the model we fine-tune in this study. Hartmann et al. [16] benchmark over 20 sentiment tools and find that fine-tuned RoBERTa achieves highest accuracy but remains sensitive to domain shift [14], consistent with the cross-platform degradation we observe.

2.3 App Store and E-Commerce Review Mining

App store reviews present unique analytical challenges: extreme brevity (median under 10 words), heavy noise (misspellings, emojis, code-mixing), and severe class imbalance driven by rating self-selection. Recent surveys confirm that app reviews encode topic-level sentiment about features, performance, and competitor comparisons, with distinct patterns per topic [17]. Dadhich and Thankachan [8] showed that hybrid rule-based approaches achieve competitive accuracy on

Amazon product reviews., highlighting the continued relevance of interpretable methods in e-commerce sentiment analysis. Maalej et al. [2] demonstrated that review intent – bug reports, feature requests, user experience — significantly shapes sentiment distributions, motivating binary classification approaches that separate the intent confound from polarity.

In the e-commerce domain, Keung et al. [15] released a multilingual Amazon product review corpus spanning six languages and showed that mBERT zero-shot cross-lingual transfer from English achieves only 67% accuracy on non-English reviews, underscoring the difficulty of domain transfer even within the same platform. However, most prior work addresses either feature-level extraction or intent classification rather than large-scale binary sentiment classification with *cross-platform* evaluation across distinct e-commerce properties. The question of whether models trained on Amazon generalize to Flipkart and vice versa remains unaddressed, constituting the primary gap our study fills.

2.4 Domain Adaptation and Class Imbalance

Cross-domain NLP, particularly sentiment transfer, has been the subject of continuous investigation [18]. Domain-adversarial training [19] employs a gradient reversal layer to acquire domain-invariant representations, resulting in a 3–7% enhancement in accuracy during domain shift on standard sentiment transfer benchmarks relative to non-adapted baselines. Zhuang et al. [20] conducted a comprehensive analysis of transfer learning methodologies, identifying "distribution mismatch" between source and target as the principal failure mode, which corresponds with our Amazon–Flipkart context.

Class imbalance makes generalization even harder. Buda et al. [21] demonstrated that imbalance diminishes classification accuracy by as much as 10% relatively, whereas Johnson and Khoshgoftaar [22] identified no singular mitigation strategy that consistently prevails. Most domain adaptation research makes the important assumption that the class priors for the source and target are the same. The Amazon–Flipkart scenario (1.6:1 vs. 5.3:1) contravenes this assumption, resulting in directional transfer effects that domain-invariant representations cannot address, thereby necessitating our cross-platform experimental design.

3 Dataset Description

3.1 Data Collection

English-language reviews are collected from the Google Play Store listings of Amazon India (in.amazon.mShop.android.shopping) and Flipkart (com.flipkart.android) via the open-source *google-play-scraper* Python library. Reviews are fetched in batches of 25,000, sorted by most recent, with language = ‘en’ and country = ‘in’ filters applied. Up to 700,000 reviews per platform are collected prior to preprocessing. Each record contains: reviewId (unique identifier), content (review text), score (1–5 star rating), thumbsUpCount, reviewCreatedVersion, and timestamp.

Table 1. Review distribution across platforms and sentiment classes after preprocessing. Pos:Neg gives the positive-to-negative ratio for binary classification.

Platform	Negative	Neutral	Positive	Total	Pos:Neg
Amazon	249,591	33,000	396,184	678,775	1.6:1
Flipkart	101,128	37,148	531,593	669,869	5.3:1
Combined	350,719	70,148	927,777	1,348,644	2.65:1

3.2 Preprocessing Pipeline

Deduplication and filtering:
Duplicate review ID entries and null or whitespace-only reviews are removed. Reviews with fewer than two characters post-cleaning are discarded.

Label assignment : Star ratings map to: 1–2 → *negative*; 3 → *neutral* ; 4–5 → *positive*. Neutral reviews are excluded for binary classification as they represent inherently ambiguous sentiment.

Full normalization : Applied for all ML and DL model training: (1) lowercasing; (2) URL and HTML removal; (3) emoji stripping via `demoji`; (4) contraction expansion using a 44-entry lookup map; (5) non-alphabetic character removal; (6) stop-word removal with NLTK (negations *no, not, never* and intensifiers *very, too, most* retained); (7) WordNet lemmatization via NLTK.

Light normalization: URL/HTML removal and contraction expansion only, preserving casing and punctuation for transformer tokenizers.

Derived features: Binary emoji, raw word count, cleaned word counts and character count.

After preprocessing: combined dataset **1,348,644 reviews**; binary classification subset **1,278,496 reviews**. The complete dataset has been publicly released by the authors on Kaggle under the CC BY-SA 4.0 license [23].

3.3 Dataset Statistics and Exploratory Data Analysis

Table 1 summarizes the final distribution. The positive-to-negative ratio differs sharply: Amazon 1.6:1 vs. Flipkart 5.3:1 — a 3.3× disparity with direct implications for negative-class detection and cross-platform transfer. Figure 1 presents the comprehensive exploratory analysis.

Key findings from the EDA: (i) the star rating distribution is strongly bimodal (1- and 5-star reviews dominate), confirming that users are polarized; (ii) Flipkart positive reviews (531,593) make up the largest group, which explains its 5.3:1 imbalance; (iii) negative reviews are always longer than positive ones, which makes sense since users write more detailed complaints [17]; and (iv) Flipkart users use emojis 56% more often than Amazon users, which shows that they communicate differently.

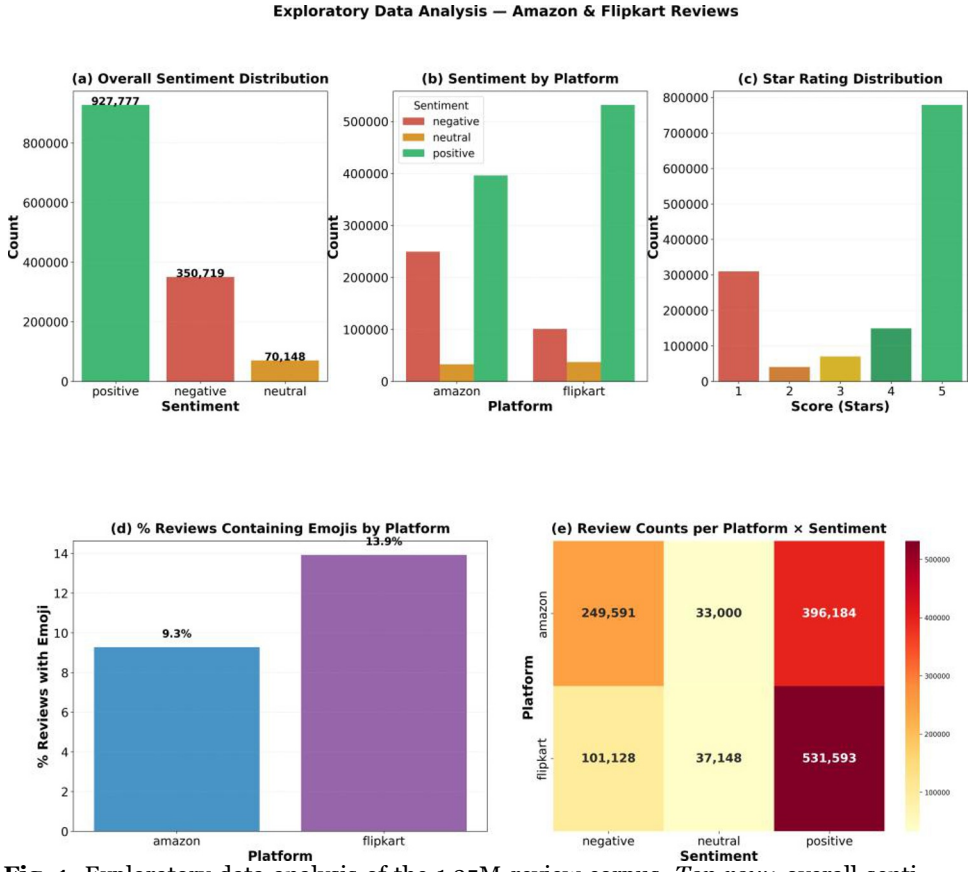

Fig. 1. Exploratory data analysis of the 1.35M-review corpus. *Top row:* overall sentiment distribution; per-platform counts; star rating distribution (bimodal: 1- and 5-star dominate). *Bottom row:* word count by sentiment class (negative: median ≈14 words; positive: ≈6); emoji prevalence by platform; review count heatmap.

3.4 Data Splits

Stratified 70/10/20 (train/val/test) split: train 894,447, val 127,799, test 255,699 reviews (72.6% positive, 27.4% negative). Per-platform stratified splits (Amazon-only: 135,396 test reviews; Flipkart-only: 120,303 test reviews) are saved for cross-platform generalization experiments.

4 Methodology

4.1 Classical Machine Learning Models

All classical models use TF-IDF with unigram and bigram tokenization (max_features = 50,000, sublinear_tf = True, min_df = 3). Class imbalance is han-

dled via class_weight = balanced (scikit-learn), proportional sample_weight for Multinomial NB, and scale_pos_weight for XGBoost. Six models are evaluated:

1. **Logistic Regression (LR):** L2 regularization, $C = 1.0$, liblinear solver.
2. **Multinomial Naive Bayes (MNB):** Laplace smoothing $\alpha = 0.1$.
3. **Linear SVM (LSVM):** LinearSVC wrapped in CalibratedClassifierCV for probability output.
4. **Random Forest (RF):** 200 estimators, balanced class weights.
5. **XGBoost (XGB):** Gradient-boosted trees with scale_pos_weight for imbalance [6].
6. **SGDClassifier (SGD):** Online linear SVM with hinge loss; trains in under 2 s.

4.2 Recurrent Deep Learning Architecture

All four recurrent models share a unified PyTorch architecture: trainable embedding (vocab 50k, dim 128, random init) → recurrent module → global max-pooling over time dimension → FC(128, ReLU, Dropout 0.3) → FC(64, ReLU) → FC(1, sigmoid). Loss: BCEWithLogitsLoss with pos_weight set to the negative-to-positive ratio per platform to address class imbalance. Optimizer: Adam, lr 10^{-3}, batch 512. Sequences padded/truncated to 100 tokens. Early stopping: patience 3, monitored on validation Macro F1.

1. **LSTM** [7]: Unidirectional, hidden 128, 2 layers, 6.54M parameters.
2. **BiLSTM**: Bidirectional LSTM, 128 units/direction, 6.68M parameters.
3. **GRU** [7]: Unidirectional, hidden 128, 2 layers, 6.51M parameters.
4. **BiGRU**: Bidirectional GRU, 128 units/direction, 6.61M parameters.

4.3 Transformer Fine-Tuning

For cross-platform experiments, we fine-tune the pretrained *Twitter-RoBERTa-base-sentiment* model [3], a RoBERTa-base model pretrained on ≈124M tweets, using a balanced stratified subset of 20,000 training reviews per platform (10k positive, 10k negative). Unlike the classical and recurrent models above which are trained from scratch and therefore benefit from the full 894k training set — pretrained transformers already encode rich linguistic and sentiment representations acquired during large-scale self-supervised pretraining [12, 13]. Fine-tuning serves only to adapt these representations to the target task, a process that typically saturates with a moderate amount of labeled data [11, 14]. Preliminary experiments confirmed this: increasing the fine-tuning set from 20k to 60k improved validation Macro F1 by less than 0.003 while tripling GPU time.

Hyperparameters: max sequence length 96, lr 2×10^{-5}, AdamW with weight decay 0.01, batch 32, 3 epochs with linear warmup.

Table 2. Classical ML model performance on the combined binary test set (255,699 reviews). Val F1 = validation Macro F1; P-Neg = negative-class precision; R-Neg = negative-class recall. t(s) = execution time in seconds. Best values per column in **bold**.

Model	Val F1	Test F1	Acc.	F1-Neg	F1-Pos	P-Neg	R-Neg	t (s)
Logistic Regression	0.8964	0.8962	0.9190	0.8477	0.9448	0.8755	0.8216	12.8
Multinomial NB	0.8802	0.8802	0.9046	0.8263	0.9342	0.8253	0.8272	<1
Linear SVM	**0.9013**	**0.9011**	**0.9244**	**0.8531**	**0.9491**	**0.9141**	0.7998	111.9
Random Forest	0.8878	0.8877	0.9114	0.8360	0.9393	0.8497	0.8227	602.6
XGBoost	0.8952	0.8946	0.9188	0.8440	0.9451	0.8923	0.8007	226.8
SGDClassifier	0.8866	0.8864	0.9101	0.8344	0.9383	0.8429	0.8262	1.1

4.4 Evaluation Protocol

Primary metric: **Macro F1** – unweighted mean of per-class F1 scores, robust to class imbalance. Secondary metrics: accuracy, per-class F1, precision, recall, and training time. All models use the same stratified combined test set (255,699 reviews). Cross-platform performance drop: $\Delta = F1_{\text{cross-platform}} - F1_{\text{in-domain}}$ (negative value = degradation; positive = improvement).

5 Results and Discussion

5.1 Baseline Verification

ZeroR (always predicts *positive*): accuracy 72.6%, Macro F1 0.4205. Stratified random: accuracy 60.25%, Macro F1 0.5007. These confirm that accuracy is a misleading metric under imbalance, and that Macro F1 is the appropriate choice for this task.

5.2 Classical ML Performance

Table 2 reports full metrics for all six classical models. Linear SVM achieves the best classical Macro F1 (0.9011), with the highest negative precision (P-Neg: 0.9141). However, its relatively low R-Neg (0.7998) indicates a conservative negative-class boundary: it misses ≈20% of true negatives while generating fewer false positives. Logistic Regression provides a more balanced precision-recall trade-off (P-Neg: 0.8755, R-Neg: 0.8216). SGDClassifier's online learning enables 1.1 s training at only 0.0147 below SVM, making it the optimal production baseline without GPU requirements. Random Forest underperforms despite its 602.6 s training cost, as tree splits struggle with TF-IDF's sparse high-dimensional feature space.

5.3 Recurrent Deep Learning Performance

Table 3 shows the results of all four recurrent models. LSTM trains for ten periods (val F1: 0.9034); BiLSTM peaks at epoch 7 (0.9035) and then stops early, reaching 0.9031 even though it has 140k more parameters and takes 1.68× longer to train. The GRU model has the highest validation F1 score (0.9039) and stops at epoch 5 in 741 s – making it the fastest DL model. BiGRU takes 6 epochs and 2.71× more time (2,009 s) to get a lower score (0.9026) which shows that bidirectionality adds overhead without helping on short reviews.

Table 3. Recurrent deep learning results on the combined binary test set. Best values per column in **bold**.

Model	Params	Epochs	Val F1	Test F1	F1-Neg	F1-Pos	t (s)
LSTM	6.54M	10	0.9034	0.9034	**0.8569**	**0.9499**	1,085
BiLSTM	6.68M	7	0.9035	0.9031	0.8568	0.9494	1,823
GRU	6.51M	5	**0.9039**	0.9033	**0.8569**	0.9496	**741**
BiGRU	6.61M	6	0.9028	0.9026	0.8559	0.9492	2,009

5.4 Full Model Comparison and Analysis

Comparative performance summary: All DL models outperform all classical ML models but by a narrow margin (<0.003 Macro F1). On short reviews (median 6–14 words), TF-IDF captures most discriminative signal, leaving limited room for sequential modeling [4]. Bidirectionality is consistently ineffective: BiLSTM and BiGRU underperform at 1.68×–2.71× the training cost. GRU provides the best overall result (Macro F1: 0.9033) at the lowest DL training cost (741 s). For deployments without GPU infrastructure, Linear SVM is the optimal choice at only 0.0022 below GRU.

Tables 2 and 3 together comprise the full 10-model comparison. Figure 2 provides a visual comparison across model families.

5.5 Cross-Platform Generalization

Table 4 reports cross-platform results. Figures 3 and 4 provide full visualizations.

Amazon → Flipkart direction. All three models degrade by 0.068–0.080 Macro F1, driven by negative-class collapse: F1-Neg drops from ≈0.85 in-domain to 0.69–0.72 cross-platform, while F1-Pos stays at ≈0.95. Models trained on Amazon's 1.6:1 distribution set a decision boundary for a higher negative base rate. When they are used on Flipkart's 5.3:1 distribution, many true negatives are incorrectly classified as positives. This is a classic case of a train-test class distribution mismatch effect [21].

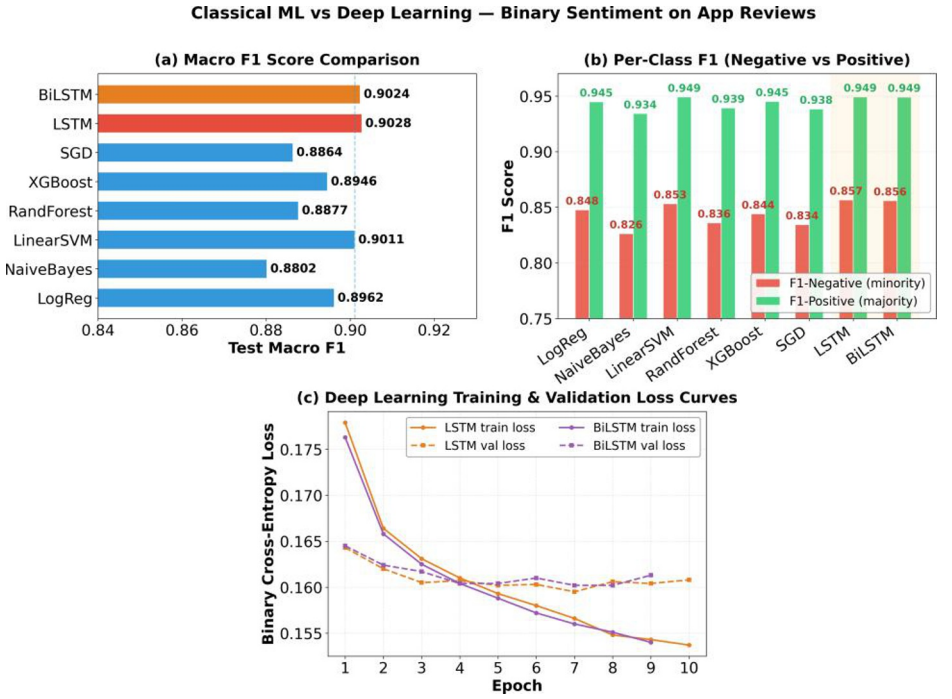

Fig. 2. Direct ML-vs-DL comparison grouped by model family. *(a)* Macro F1 by family. *(b)* Per-class F1 breakdown (F1-Neg and F1-Pos). *(c)* LSTM/BiLSTM training and validation loss curves. All DL models outperform all ML models in Macro F1 and F1-Neg, but the margin is minimal (< 0.003), confirming that TF-IDF captures most discriminative signal in short app reviews.

Flipkart → Amazon direction. GRU exceeds its in-domain score (Δ = +0.0019), LinearSVM degrades by only 0.0161, and RoBERTa improves from 0.8336 to 0.9052 (Δ = +0.0715). Flipkart's harder 5.3:1 training environment forces more conservative negative-class boundaries that transfer well to Amazon's more balanced 1.6:1 distribution [20].

Cross-platform generalization summary. Class imbalance in the source domain is the binding constraint for cross-platform generalization, independent of model architecture or pretraining. RoBERTa's Twitter pretraining achieves the best in-domain score (0.9174 on Amazon) but is not immune to domain shift: it degrades by 0.0791 in Amazon → Flipkart, identical to SVM (0.0797). Both GRU and RoBERTa reach 0.9052 in Flipkart → Amazon, confirming that the bottleneck is distributional rather than representational.

5.6 Summary of Findings

– **Model comparison:** Linear SVM (Macro F1: 0.9011) is competitive with GRU (0.9033) at 6.6× lower training cost. All DL models outperform all

Table 4. Cross-platform generalization: LinearSVM, GRU, and RoBERTa. $\Delta < 0$: degradation; $\Delta > 0$: improvement. RoBERTa in-domain score is on 20k balanced subset; cross-platform on full target test set.

Model	Direction	In-Domain	Cross-P	Δ	F1-Neg	F1-Pos
LinearSVM	Amazon → Flipkart	0.9011	0.8214	−0.0797	0.6944	0.9483
GRU	Amazon → Flipkart	0.9033	0.8354	−0.0679	0.7168	0.9540
RoBERTa	Amazon → Flipkart	0.9174	0.8382	−0.0791	0.7214	0.9551
LinearSVM	Flipkart → Amazon	0.9011	0.8850	−0.0161	0.8611	0.9089
GRU	Flipkart → Amazon	0.9033	0.9052	+0.0019	0.8828	0.9277
RoBERTa	Flipkart → Amazon	0.8336	0.9052	+0.0715	0.8841	0.9262

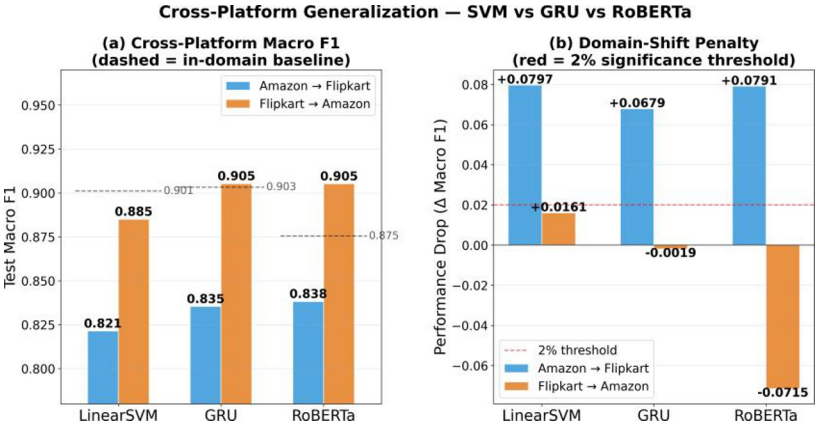

Fig. 3. Full cross-platform comparison for LinearSVM, GRU, and RoBERTa across both transfer directions. The asymmetry is consistent across all three architectures: Amazon → Flipkart degrades 0.068–0.080 in Macro F1, while Flipkart → Amazon is near-lossless or improves. This verifies that distributional differences, as opposed to model-specific traits, are the primary driver of the asymmetry.

classical models by <0.003. Bidirectional architectures are consistently worse than unidirectional on short app reviews.

- **Cross-platform transfer:** Transfer is highly asymmetric. Flipkart → Amazon: near-lossless for all models (GRU Δ = +0.0019, RoBERTa Δ = +0.0715). Amazon → Flipkart: 0.068–0.080 Macro F1 drop concentrated in the negative class, consistent across SVM, GRU, and RoBERTa.
- **Class imbalance:** In-domain imbalance is mitigated via cost-sensitive losses (class_weight='balanced', pos_weight), and Macro F1 is used throughout as an imbalance-robust metric. Despite these mitigations, the *cross-platform distributional mismatch* (Amazon 1.6:1 vs. Flipkart 5.3:1) still drives asymmetric transfer degradation — a distinct, harder problem that per-model reweighting does not resolve.

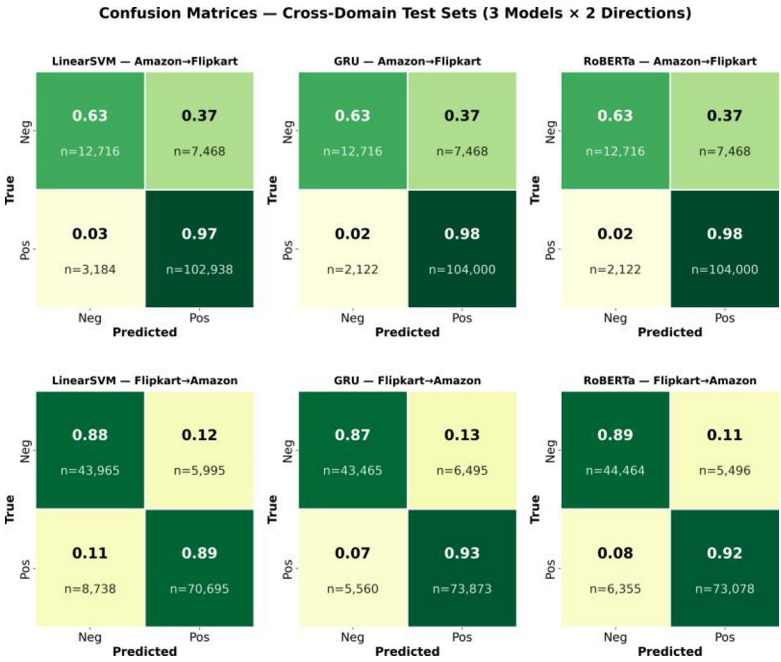

Fig. 4. Cross-platform confusion matrices for all three model families (rows: LinearSVM, GRU, RoBERTa; columns: Amazon → Flipkart, Flipkart → Amazon). Amazon → Flipkart: negative recall collapses to 0.55–0.60 across all models. Flipkart → Amazon: both classes are classified accurately with no visible degradation.

– **Practical recommendation:** When deploying across both platforms, train on the platform with higher class imbalance (Flipkart) to obtain better cross-platform negative-class coverage.

6 Conclusion

We presented a large-scale empirical study benchmarking six classical ML models and four recurrent DL architectures on 1.35M Amazon and Flipkart reviews, with the first cross-platform generalization study augmented by a RoBERTa transformer baseline.

Key conclusions: (1) Linear SVM with TF-IDF achieves competitive Macro F1 (0.9011) with no GPU requirements; (2) GRU provides the best DL result (0.9033) at the highest training efficiency (741 s); (3) bidirectional architectures (BiLSTM, BiGRU) are ineffective on short-text reviews; (4) cross-platform transfer is highly asymmetric — Flipkart-trained models generalize to Amazon, but not vice versa; and (5) even with cost-sensitive loss functions and imbalance-robust evaluation, the cross-platform class distribution mismatch (1.6:1 vs. 5.3:1) remains the dominant driver of degradation, independent of model architecture or pretraining.

These findings provide actionable guidelines for practitioners: prefer the higher-imbalance platform as the training source for better generalization.

7 Future Work

Future work will investigate adversarial domain adaptation [19], pseudo-labeling with unlabeled target-domain data, and dynamic rebalancing strategies. Extending to additional Indian e-commerce platforms (Meesho, Myntra), multilingual Hindi-English code-mixed reviews, and aspect-level cross-platform sentiment transfer represent promising further directions.

References

1. M. Birjali, M. Kasri, and A. Beni-Hssane, "A comprehensive survey on sentiment analysis: Approaches, challenges, and trends," *Knowledge-Based Systems*, vol. 226, p. 107174, 2021. [Online]. Available: https://doi.org/10.1016/J.KNOSYS.2021.107134
2. W. Maalej, Z. Kurtanovic, H. Nabil, and C. Stanik, "On the automatic classification of app reviews," *Requirements Engineering*, vol. 21, no. 3, pp. 311–331, 2016. [Online]. Available: https://doi.org/10.1007/s00766-016-0251-9
3. F. Barbieri, J. Camacho-Collados, L. Espinosa-Anke, and L. Neves, "TweetEval: Unified benchmark and comparative evaluation for tweet classification," in *Findings of EMNLP*, 2020. [Online]. Available: https://doi.org/10.48550/arXiv.2010.12421
4. L. Zhang, S. Wang, and B. Liu, "Deep learning for sentiment analysis: A survey," *WIREs Data Mining and Knowledge Discovery*, vol. 8, no. 4, e1253, 2018. [Online]. Available: https://doi.org/10.48550/arXiv.1801.07883
5. A. Yadav and D. K. Vishwakarma, "Sentiment analysis using deep learning architectures: A review," *Artificial Intelligence Review*, vol. 53, no. 6, pp. 4335–4385, 2020. [Online]. Available: https://doi.org/10.1007/s10462-019-09794-5
6. T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, 2016, pp. 785–794. [Online]. Available: https://doi.org/10.1145/2939672.2939785
7. Y. Yu, X. Si, C. Hu, and J. Zhang, "A review of recurrent neural networks: LSTM cells and network architectures," *Neural Computation*, vol. 31, no. 7, pp. 1235–1270, 2019. [Online]. Available: https://doi.org/10.1162/neco_a_01199
8. A. Dadhich and B. Thankachan, "Sentiment analysis of Amazon product reviews using hybrid rule-based approach," *SN Computer Science*, vol. 5, no. 4, p. 384, 2024. [Online]. Available: https://doi.org/10.1007/978-981-16-2877-1_17
9. N. C. Dang, M. N. Moreno-García, and F. De la Prieta, "Sentiment analysis based on deep learning: A comparative study," *Electronics*, vol. 9, no. 3, p. 483, 2020. [Online]. Available: https://doi.org/10.48550/arXiv.2006.03541
10. A. Vaswani *et al.*, "Attention is all you need," in *Advances in Neural Information Processing Systems*, vol. 30, 2017. [Online]. Available: https://doi.org/10.48550/arXiv.1706.03762
11. J. Howard and S. Ruder, "Universal language model fine-tuning for text classification," in *Proc. 56th Annual Meeting of the Association for Computational Linguistics*, 2018, pp. 328–339. [Online]. Available: https://doi.org/10.48550/arXiv.1801.06146

12. 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, pp. 4171–4186. [Online]. Available: https://doi.org/10.18653/v1/N19-1423
13. Y. Liu *et al.*, "RoBERTa: A robustly optimized BERT pretraining approach," *arXiv preprint arXiv:1907.11692*, 2019. [Online]. Available: https://doi.org/10.48550/arXiv.1907.11692
14. S. Gururangan, A. Marasovic, S. Swayamdipta, K. Lo, I. Beltagy, D. Downey, and N. A. Smith, "Don't stop pretraining: Adapt language models to domains and tasks," in *Proc. 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 8342–8360. [Online]. Available: https://doi.org/10.48550/arXiv.2004.10964
15. P. Keung, Y. Lu, G. Szarvas, and N. A. Smith, "The multilingual Amazon reviews corpus," in *Proc. EMNLP*, 2020, pp. 4563–4568. [Online]. Available: https://doi.org/10.48550/arXiv.2010.02573
16. J. Hartmann, M. Heitmann, C. Siebert, and C. Schamp, "More than a feeling: Accuracy and application of sentiment analysis," *International Journal of Research in Marketing*, vol. 40, no. 1, pp. 75–90, 2023. [Online]. Available: https://doi.org/10.1016/j.ijresmar.2022.05.005
17. M. Wankhade, A. C. S. Rao, and C. Kulkarni, "A survey on sentiment analysis methods, applications, and challenges," *Artificial Intelligence Review*, vol. 55, no. 7, pp. 5731–5780, 2022. [Online]. Available: https://doi.org/10.1016/j.ijresmar.2022.05.005
18. A. Ramponi and B. Plank, "Neural unsupervised domain adaptation in NLP — A survey," in *Proc. 28th Int. Conf. Computational Linguistics*, 2020, pp. 6838–6855. [Online]. Available: https://doi.org/10.48550/arXiv.2006.00632
19. Y. Ganin *et al.*, "Domain-adversarial training of neural networks," *Journal of Machine Learning Research*, vol. 17, no. 59, pp. 1–35, 2016. [Online]. Available: https://jmlr.org/papers/v17/15-239.html
20. F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He, "A comprehensive survey on transfer learning," *Proceedings of the IEEE*, vol. 109, no. 1, pp. 43–76, 2021. [Online]. Available: https://doi.org/10.48550/arXiv.1505.07818
21. M. Buda, A. Maki, and M. A. Mazurowski, "A systematic study of the class imbalance problem in convolutional neural networks," *Neural Networks*, vol. 106, pp. 249–259, 2018. [Online]. Available: https://doi.org/10.1016/j.neunet.2018.07.011
22. J. M. Johnson and T. M. Khoshgoftaar, "Survey on deep learning with class imbalance," *Journal of Big Data*, vol. 6, no. 1, pp. 1–54, 2019. [Online]. Available: https://doi.org/10.1186/s40537-019-0192-5
23. Y. K. Arora "Amazon & Flipkart App Review Sentiment Dataset," Kaggle, 2025. [Online]. Available: https://www.kaggle.com/datasets/yashkumararora/amazonflipkart-app-review-sentiment-dataset