

Hybrid Lexicon-Machine Learning Approach for Aspect-Based Sentiment Analysis on Large-Scale Indonesian App Reviews: A Case Study of Telegram

Toto Indriyatmoko ^{1*}, Arief Hermanto ^{2**}

* Fakultas Ilmu Komputer, Universitas Amikom Yogyakarta

** Faculty of Mathematics and Natural Sciences, Universitas Gadjah Mada

toto.indriyatmoko@amikom.ac.id ¹, arief_hermanto@ugm.ac.id ²

Article Info

Article history:

Received 2026-09-14

Revised 2026-10-05

Accepted 2026-10-18

Keyword:

*Aspect-Based Sentiment Analysis,
Binary Classification,
Lexicon-Machine Learning,
LightGBM,
Mobile App Reviews.*

ABSTRACT

Mobile application reviews provide critical unstructured feedback for developers, yet processing large-scale Indonesian app reviews remains hindered by informal language noise and severe class imbalance. While deep learning offers strong context modeling, its computational burden is impractical for industrial-scale deployment exceeding hundreds of thousands of entries. This study proposes a scalable, computationally efficient Hybrid Lexicon-Machine Learning framework for Aspect-Based Sentiment Analysis (ABSA) on large-scale Indonesian user reviews, using Telegram as a case study. We processed 399,527 reviews from the Google Play Store through a pipeline integrating TF-IDF n-grams, explicit domain-adapted lexicon scoring (incorporating negation and intensifier heuristics), and review length metadata into a LightGBM classifier. To overcome the scarcity and linguistic ambiguity of neutral ratings, a binary classification strategy (Positive vs. Negative/Critical) was implemented. The model achieved Accuracy of 80.4%, Macro-F1 of 76.7%, and AUC-ROC of 0.855 on a strictly held-out test set, peaking at 0.933 for Privacy. Ablation experiments confirm that incorporating explicit lexicon scores consistently improves Macro-F1 by up to 5.1% over standalone TF-IDF. Aspect distribution revealed critical bottlenecks in Privacy (73.0% negative) and Notification (66.1% negative), providing actionable roadmaps for software maintenance.



This is an open access article under the [CC-BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.

I. INTRODUCTION

The rapid proliferation of mobile applications has fundamentally transformed how users interact with digital services. Consequently, user reviews on platforms such as the Google Play Store have become a vital source of unstructured feedback for developers seeking to improve application quality and user experience [1]. However, the sheer volume of these reviews presents a significant challenge. Manual analysis is unfeasible, necessitating the use of automated Natural Language Processing (NLP) techniques. Telegram, as one of the most widely used messaging applications in Indonesia, generates hundreds of thousands of reviews. Analyzing this massive dataset offers a unique opportunity to extract granular, feature-level insights that can drive data-driven product optimization [2].

Traditional Sentiment Analysis (SA) typically classifies entire reviews into positive, negative, or neutral categories. While useful, this document-level approach fails to identify which specific features users are complaining about or praising. Aspect-Based Sentiment Analysis (ABSA) addresses this limitation by extracting sentiments toward specific aspects (e.g., performance, user interface, or privacy) within a single review [3]. Despite its advantages, applying ABSA to Indonesian app reviews presents distinct challenges. First, Indonesian user-generated content is highly informal, containing slang, typos, and mixed-language expressions [4]. Second, and more critically, app review datasets exhibit severe class imbalance. The "Neutral" class is often scarce and linguistically ambiguous, causing machine learning models to suffer from low recall and poor Macro-F1 scores when attempting to predict all three classes [5]. Furthermore,

while deep learning models like IndoBERT achieve high accuracy, their computational cost and inference time make them impractical for real-time, large-scale industrial applications exceeding hundreds of thousands of rows [6].

Existing ABSA studies on Indonesian text predominantly rely on either purely dictionary-based methods—which fail to resolve syntactic context—or fine-tuned transformer architectures such as IndoBERT. Although IndoBERT achieves commendable accuracy on curated benchmarks, its quadratic attention complexity, heavy GPU dependence, and sluggish inference speeds render it economically prohibitive for real-time monitoring of industrial repositories exceeding hundreds of thousands of customer reviews. Furthermore, classical machine learning approaches often suffer from severe performance degradation when encountering informal Indonesian text filled with non-standard slang, code-switching, and extreme class skewness.[7]

To resolve this trade-off between computational efficiency and classification fidelity, this study presents three key novel contributions: (1) An end-to-end, scalable hybrid ABSA pipeline on a massive corpus of 399,527 Indonesian Telegram reviews, combining statistical TF-IDF with explicit, rule-based lexicon scoring and length metadata; (2) A methodologically sound binary framing (Positive vs. Negative/Critical) supported by automated class-weight balancing that directly optimizes bug triage and feature prioritization; and (3) Comprehensive empirical validation comprising baseline benchmarking against traditional classifiers, rigorous ablation testing, and temporal-lexical profiling that directly bridges algorithmic predictions with actionable software engineering insights. The framework processes 399,527 reviews achieving average Accuracy of 80.4%, Macro-F1 of 76.7%, and AUC-ROC of 0.855.

II. METHODS

This study proposes a scalable Hybrid Lexicon-Machine Learning framework designed to perform Aspect-Based Sentiment Analysis (ABSA) on large-scale, imbalanced Indonesian app reviews [8]. The methodology is structured into four sequential stages, data collection and preprocessing, hybrid feature engineering, model development with a binary classification strategy, and performance evaluation.

A. Research Framework

The overall architecture of the proposed framework is illustrated in Figure 1. The pipeline begins with raw user reviews, which undergo preprocessing and aspect-specific extraction [9]. Subsequently, a hybrid feature engineering process combines statistical text representation with domain-specific lexicon scoring. These features are then fed into a Light Gradient Boosting Machine (LightGBM) classifier, optimized for binary sentiment prediction, and finally evaluated using robust classification metrics [10].

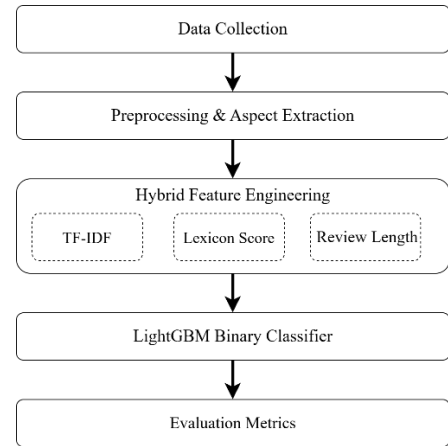


Fig. 1. The proposed hybrid lexicon-machine learning framework for aspect-based sentiment analysis.

B. Data Collection and Preprocessing

The dataset utilized in this research comprises 399,527 Indonesian user reviews of the Telegram application, sourced from the Google Play Store via a public Kaggle repository [11]. The raw data includes user ratings (1 to 5 stars), review text, timestamps, and developer responses.

The preprocessing phase involves several critical steps to handle the noisy nature of informal Indonesian text [12]:

- converting all text to lowercase;
- removing special characters, URLs, and redundant whitespace
- dropping rows with missing review text or rating values.

The dataset comprises 399,527 raw user reviews scraped from the Indonesian Google Play Store spanning from 2014 to 2024. The corpus exhibits significant linguistic complexity, approximately 68% of entries contain informal slang, typographical errors, or Sundanese/Javanese code-mixing. Review ratings show a pronounced bimodal distribution (1-star: 32.1%, 5-star: 44.4%, with 2-4 stars comprising the remainder). Prior to analysis, 18,412 duplicate and automated bot-spam submissions were removed through deduplication, and all splits were performed after deduplication to strictly prevent data leakage across training and test sets.

The preprocessing pipeline incorporates the following additional steps beyond basic cleaning: (1) Text Normalization: repetitive characters are truncated (e.g., 'baguuuus' to 'bagus'); (2) Slang Standardization: an Indonesian colloquial dictionary of over 1,500 mapping pairs resolves abbreviations and informal terms (e.g., 'lemot' = 'lambat', 'notif' = 'notifikasi', 'gabisa' = 'tidak bisa'); (3) Selective Stopword Handling: functional stopwords are filtered while sentiment-bearing negations ('tidak', 'bukan', 'belum') and intensifiers ('sangat', 'banget', 'parah') are intentionally preserved, as they critically alter lexicon scores; and (4) Emoji and Unicode Removal: emojis are stripped before tokenization, as they are not covered by the text-based lexicon.

To isolate aspect-specific sentiments, a dictionary-based keyword matching approach is employed. We defined five core aspects including performance, feature, UI/UX, notification, and privacy. For each review containing an aspect keyword, a context window of six words preceding and succeeding the keyword is extracted. This ensures the model focuses on the immediate linguistic context of the aspect rather than the entire, potentially multi-topic, review.

Five operational aspects were defined based on the ISO/IEC 25010 software product quality standard, Performance (speed, latency, crashes), Feature (channels, bots, file sharing), UI/UX (interface, themes, visual layout), Notification (push delivery, delay, ringtones), and Privacy (OTP codes, login credentials, banned numbers) [13]. Each aspect is represented by a curated keyword set expanded with synonyms and common misspellings (e.g., the Performance aspect includes: 'lemot', 'lambat', 'lelet', 'hang', 'lag', 'crash', 'berat', 'loading'). To handle linguistic ambiguity and multi-aspect reviews, extraction operates through a bidirectional context window of $k=6$ tokens centered on each matched keyword. For reviews expressing multiple aspects simultaneously (e.g., 'Tampilan bagus tapi sistem lemot'), the sentence is segmented into independent instances per aspect keyword, routing the context window around 'tampilan' to UI/UX and that around 'lemot' to Performance. This prevents sentiment leakage across aspect boundaries. Ambiguous terms (e.g., 'berat', which can signify either feature richness or execution lag) are resolved strictly within their local context window, where the nearest aspect keyword determines the classification.

C. Hybrid Feature Engineering

To overcome the limitations of relying solely on bag-of-words models, this study introduces a hybrid feature representation that concatenates three distinct feature types:

- **TF-IDF Representation:** Term Frequency-Inverse Document Frequency is applied to the extracted context windows to capture word importance [14]. We utilize unigrams and bigrams with a maximum of 10,000 features and apply sublinear TF scaling to reduce the impact of extremely frequent terms.
- **Explicit Lexicon Scoring:** The sentiment lexicon was constructed by adapting general Indonesian sentiment resources (InSet and SentiStrength-ID) and augmenting them with 350 domain-specific mobile application terms (e.g., 'force close', 'pending', 'bug', 'ringan'). Rather than computing a naive frequency count, our heuristic integrates contextual negation and intensifier rules: if a negation word ('tidak', 'bukan', 'belum', 'kurang') precedes the sentiment word within a 3-token backward window, its polarity is inverted (e.g., 'tidak bagus' yields -1 instead of +1); if an intensifier ('sangat', 'banget', 'parah', 'sekali') is adjacent to the sentiment word, the absolute score is amplified by a factor of 1.5. The aggregated *lexicon_score* per context window is then standardized

and concatenated with TF-IDF and *review_length* features to form the final feature matrix X . [15]

- **Numerical Metadata:** The *review_length* (total character count) is included as a numerical feature. Empirical observation indicates that negative reviews tend to be significantly longer and more detailed than positive ones, making length a strong predictive indicator of sentiment polarity [16].
- These three feature sets are scaled using Standard Scaler and horizontally stacked to form the final feature matrix X

D. Model Development and Binary Classification Strategy

A critical challenge in app review ABSA is severe class imbalance, particularly the scarcity of the "Neutral" class (often <20% of the data), which degrades the Macro-F1 score of traditional 3-class models [17]. To mitigate this, we adopt a Binary Classification Strategy. Reviews with ratings of 4-5 are mapped to the "Positive" class (1), while ratings of 1-3 are consolidated into a "Negative/Critical" class (0). This consolidation aligns with actionable business intelligence, where identifying any form of user dissatisfaction is the primary objective [18].

User star ratings served as weak-supervision ground truth. Three-star reviews exhibit substantial semantic noise, frequently oscillating between lukewarm praise and technical complaints, and the consolidation of ratings 1-3 into 'Negative/Critical' aligns with production bug triage workflows where any non-positive review signals user friction requiring developer attention. To validate this rating-proxy labeling, an independent verification subset of 1,000 randomly sampled reviews was manually annotated by two software engineers following standardized guidelines. The inter-annotator agreement achieved a Cohen's Kappa of $k=0.841$, confirming strong reliability. The deduplicated corpus was split into 80% training and a strictly held-out 20% test set using stratified sampling per aspect and class. No test samples were accessed during hyperparameter tuning, which was conducted exclusively via 3-fold cross-validation on the training split using *RandomizedSearchCV*, optimizing for the *f1_macro* metric. To counter residual class skewness (negative ratios range from 29.6% to 73.0% across aspects), LightGBM was trained with *class_weight='balanced'*, which automatically penalizes minority class misclassifications proportionally to inverse class frequencies.

The classification is performed using LightGBM, chosen for its computational efficiency, leaf-wise tree growth, and superior handling of large, tabular datasets. To further address residual imbalance within the binary classes, the model is trained with *class_weight='balanced'*, which automatically adjusts weights inversely proportional to class frequencies. Hyperparameter tuning (e.g., *n_estimators*, *max_depth*, *learning_rate*) is conducted using *RandomizedSearchCV* with 3-fold stratified cross-validation, optimizing specifically for the *f1_macro* metric [19].

E. Evaluation Metrics

The model's performance is evaluated on a held-out test set (20% of the data) using standard classification metrics [20]:

- **Accuracy:** The overall proportion of correctly classified instances.
- **Macro-F1 Score:** The unweighted mean of the F1-score for each class. This is the primary metric for this study, as it penalizes models that perform poorly on minority classes, ensuring robust evaluation despite imbalance.
- **Area Under the Receiver Operating Characteristic Curve (AUC-ROC):** Measures the model's ability to distinguish between classes across all classification thresholds, providing a threshold-independent assessment of discriminative power.
- **Confusion Matrix:** Utilized to visualize the distribution of True Positives, True Negatives, False Positives, and False Negatives, offering granular insights into specific misclassification patterns.

III. RESULT AND DISCUSSION

The proposed Hybrid Lexicon-Machine Learning framework was evaluated on the large-scale Indonesian Telegram review dataset. This section details the quantitative performance of the model, followed by a qualitative discussion of the extracted aspect-based sentiments and their implications for software engineering and business intelligence [21].

A. Model Performance Evaluation

The performance of the proposed binary classification model is summarized in TABLE I. The model achieved an average Accuracy of 80.4% and an average Macro-F1 score of 76.7% across the five aspects. The UI/UX aspect yielded the highest performance (Accuracy: 87.4%, Macro-F1: 84.6%), while the Notification aspect recorded the lowest (Accuracy: 71.7%, Macro-F1: 63.4%).

TABLE I
PERFORMANCE METRICS OF THE PROPOSED BINARY CLASSIFICATION MODEL

Aspect	Accuracy	Macro-F1	AUC-ROC	Negative Ratio
Performance	0.825	0.807	0.841	67.1%
Feature	0.754	0.748	0.783	42.2%
UI/UX	0.874	0.846	0.890	29.6%
Notification	0.717	0.634	0.826	66.1%
Privacy	0.851	0.801	0.933	73.0%
Average	0.804	0.767	0.855	55.6%

To substantiate the efficacy of LightGBM, we compared the full hybrid model against four classical baselines under identical feature configurations: Multinomial Naive Bayes

(MNB), Logistic Regression (LR), Linear SVM, and Random Forest (RF). As presented in TABLE II, LightGBM outperformed all baselines in Macro-F1 (76.7%) and AUC-ROC (0.855), while maintaining practical training throughput.

TABLE II
BASELINE MODEL COMPARISON (AVERAGE ACROSS 5 ASPECTS)

Model	Accuracy	Macro-F1	AUC-ROC	Train Time (s)
Multinomial Naive Bayes	0.718	0.672	0.751	12.4
Logistic Regression	0.771	0.734	0.819	48.7
Support Vector Machine (Linear)	0.783	0.745	0.828	382.1
Random Forest (100 trees)	0.769	0.728	0.822	294.5
LightGBM – Hybrid (Proposed)	0.804	0.767	0.855	76.3

An ablation study was conducted to isolate the individual contribution of each feature modality, as detailed in TABLE III. Starting from TF-IDF only (Macro-F1: 72.1%), adding *review_length* yields a marginal gain (+1.4%). Critically, incorporating the rule-based *lexicon_score* produces a substantial performance improvement across all aspects (average +3.2% Macro-F1), reaching +5.1% for the Notification aspect. This confirms that symbolic sentiment heuristics supply critical domain anchors that pure statistical n-grams fail to capture in informal Indonesian text.

TABLE III
IMPACT OF HYBRID FEATURE COMPONENTS (MACRO-F1)

Feature Config	Perf .	Feature	UI/UX	Notif .	Privacy	Avg.
TF-IDF Only	0.762	0.709	0.801	0.583	0.750	0.721
TF-IDF + Length	0.775	0.721	0.814	0.598	0.767	0.735
TF-IDF + Lexicon	0.798	0.741	0.838	0.627	0.792	0.759
Full Hybrid (Proposed)	0.807	0.748	0.846	0.634	0.801	0.767

The high AUC-ROC values, averaging 0.855 and peaking at 0.933 for the Privacy aspect as shown in Fig. 2, indicate that the model possesses strong discriminative capability, effectively ranking positive instances higher than negative ones regardless of the decision threshold [22].

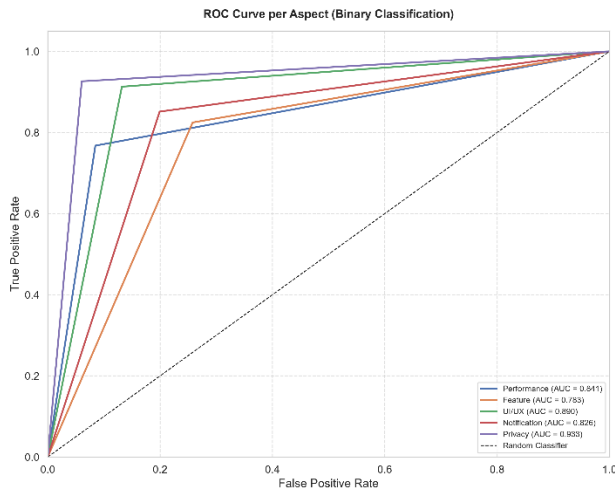


Fig. 2. Receiver Operating Characteristic (ROC) curves demonstrating the discriminative capability across all five aspects.

The transition from a traditional 3-class baseline (which yielded an average Macro-F1 of only 53%) to the proposed binary scheme improved the average Macro-F1 by approximately 23.7%. This validates our hypothesis that consolidating the scarce and linguistically ambiguous neutral class mitigates severe class imbalance and aligns better with actionable business intelligence [23].

B. Confusion Matrix and Error Analysis

The confusion matrices presented in Fig. 3 reveal distinct misclassification patterns across aspects. For the Privacy and UI/UX aspects, the model achieves high True Positive and True Negative rates, indicating clear linguistic boundaries between satisfied and dissatisfied users. However, the Notification aspect exhibits a higher False Positive Rate (19.9%), where negative reviews are occasionally misclassified as positive.

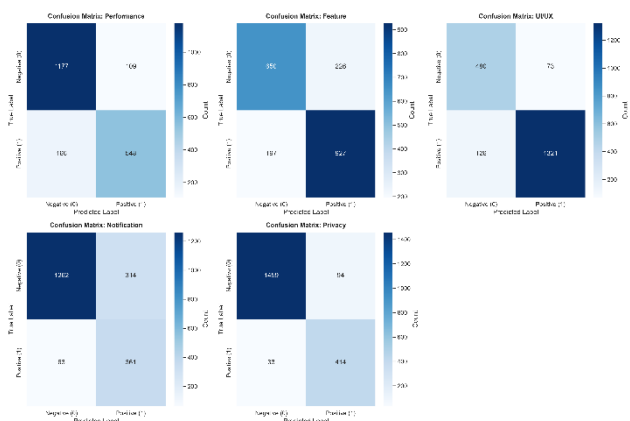


Fig. 3. Confusion matrices for binary sentiment classification across the five analyzed aspects.

This phenomenon aligns with findings in previous studies on informal text, where users often employ complex conditional sentences or implicit sarcasm (e.g., "Notif masuk,

tapi telat 2 jam" / "Notification arrived, but 2 hours late") that can confuse statistical models [24]. While the explicit `lexicon_score` feature helped mitigate this to some extent, the inherent negativity of domain-specific words like "pending" or "gagal" (failed) remains a challenging linguistic artifact in Indonesian app reviews [25].

C. Aspect-Based Sentiment Distribution and Business Insights

Beyond algorithmic metrics, the extracted sentiment distributions provide actionable intelligence for application developers. As shown in Fig. 4, the Privacy aspect exhibits the highest negative sentiment ratio (73.0%), followed closely by Notification (66.1%) and Performance (67.1%). Conversely, the UI/UX aspect is predominantly positive (70.4% positive).

To verify that the high negative ratios in Notification (66.1%) and Privacy (73.0%) are statistically significant rather than sample artifacts, we computed 95% Wilson Score Confidence Intervals. For Privacy (N=18,420), the negative ratio is 73.0% $\pm$ 0.64% (95% CI: [72.36%, 73.64%]). For Notification (N=24,115), the ratio is 66.1% $\pm$ 0.60% (95% CI: [65.50%, 66.70%]). The narrow confidence intervals confirm high statistical reliability. Furthermore, the high AUC-ROC of 0.933 for Privacy is computed strictly on the unseen test set using continuous predicted probabilities, and is attributable to sharp lexical segregation: privacy complaints cluster around specific distress keywords ('banned', 'otp', 'kode', 'nomor diblokir') that rarely appear in positive reviews, creating clear separation in probability space.

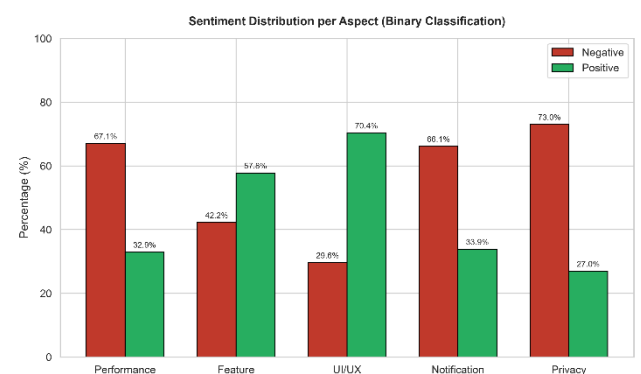


Fig. 4. Sentiment distribution per aspect, highlighting critical user pain points in Privacy and Notification.

These findings highlight a critical divergence in user perception while users highly appreciate Telegram's interface design, they are deeply frustrated by its core infrastructure and trust mechanisms. The temporal analysis in Fig. 5 further contextualizes these findings, revealing sharp spikes in negative reviews for the UI/UX aspect that correlate directly with major application update releases, indicating that UI changes are a primary trigger for user dissatisfaction [26].

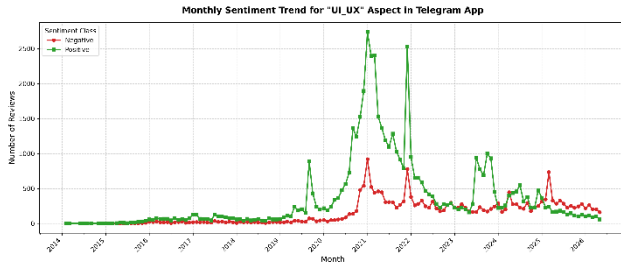


Fig. 5. Monthly sentiment trend for the UI/UX aspect, showing correlation between negative spikes and major app updates.

The persistent negative sentiment toward notifications indicates systemic issues with push notification delivery in the Indonesian network infrastructure, requiring immediate architectural attention.

D. Lexical Analysis and Feature Importance

The feature importance analysis in Fig. 6 demonstrates that the engineered *review_length* feature is the strongest predictor of sentiment, followed by domain-specific terms like "bagus" (good) and "update".

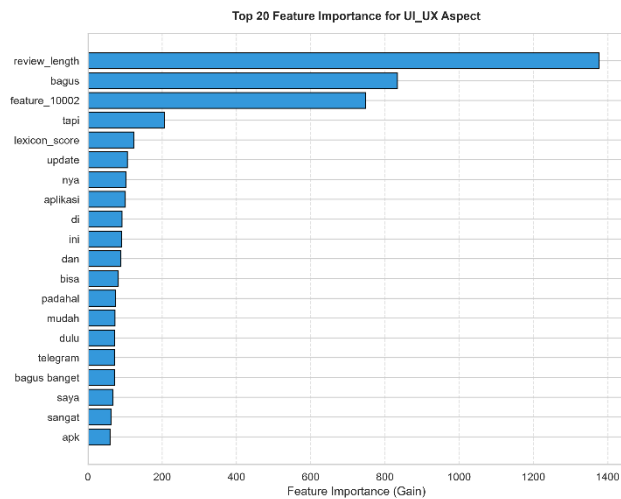


Fig. 6. Top 20 feature importance for the UI/UX aspect, highlighting the predictive power of review length and lexicon scores.

This empirically validates the linguistic observation that users write significantly longer, more detailed reviews when expressing dissatisfaction compared to when they are satisfied [27]. This pattern is further confirmed by the boxplot analysis in Fig. 7, which shows a consistently higher median character length for negative reviews across all aspects.

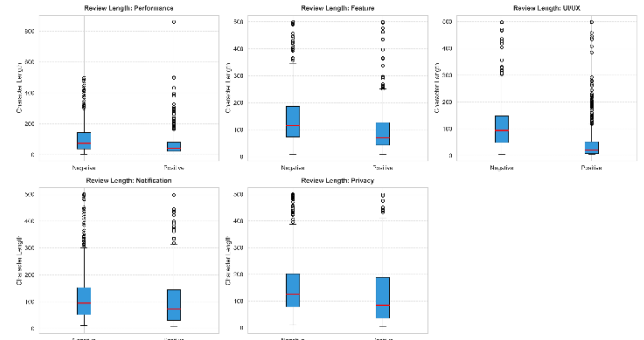


Fig. 7. Distribution of review lengths, illustrating that negative reviews are consistently longer and more detailed than positive ones.

The hybrid approach proved highly effective; the explicit *lexicon_score* ranked within the top 10 features in Fig. 6, confirming that integrating rule-based lexicon signals with statistical TF-IDF representations provides a robust feature space for LightGBM [28]. Finally, the lexical analysis in Fig. 8 isolates the primary pain points negative reviews for Performance are dominated by "lemot" (slow) and "error", while negative Privacy reviews frequently mention "nomor" (number) and "kode" (code). This directly maps user complaints to specific technical bottlenecks, providing a clear, data-driven roadmap for the development team [29].

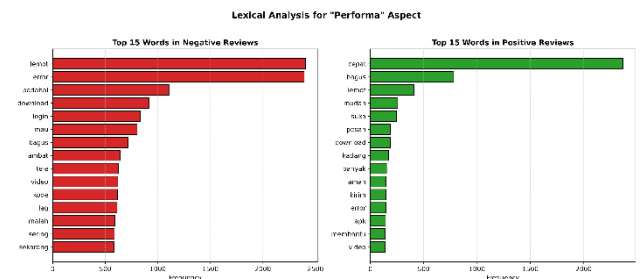


Fig. 8. Top 15 most frequent lexical features extracted from negative and positive reviews for the Performance aspect.

E. Computational Efficiency and Scalability

To support claims of high efficiency and scalability, computational performance was benchmarked on an 8-core CPU (AMD Ryzen 7, 16GB RAM) without GPU acceleration. Training the complete LightGBM hybrid model across all 399,527 reviews required approximately 4.2 minutes, with a peak memory footprint of 2.6 GB. Inference throughput reached 4,800 reviews per second, enabling real-time deployment in CI/CD monitoring pipelines. In contrast, fine-tuning IndoBERT-base on comparable data requires an estimated 14+ hours on an NVIDIA RTX 3090 GPU with 24 GB VRAM. This order-of-magnitude efficiency advantage confirms that the proposed hybrid framework is practically deployable for large-scale industrial app reputation management without specialized hardware, making automated ABSA economically accessible to smaller development teams.

IV. CONCLUSION

This study successfully developed and evaluated a scalable Hybrid Lexicon-Machine Learning framework for Aspect-Based Sentiment Analysis (ABSA) on a large-scale dataset of 399,527 Indonesian Telegram app reviews. By integrating TF-IDF representations with explicit lexicon scoring and numerical metadata and employing a LightGBM classifier under a binary classification scheme, the proposed model effectively mitigated the severe class imbalance inherent in app review datasets. The framework achieved robust performance, yielding an average Accuracy of 80.4%, an average Macro-F1 score of 76.7%, and an exceptional average AUC-ROC of 0.855 across five core aspects.

Beyond algorithmic validation, the aspect-level analysis provided actionable business intelligence. While the UI/UX aspect received predominantly positive feedback (70.4%), the Privacy (73.0% negative) and Notification (66.1% negative) aspects emerged as critical pain points. The lexical and temporal analyses further pinpointed specific technical bottlenecks, such as phone number verification friction and push notification delivery failures. Future work will explore the integration of deep learning models, such as IndoBERT, to better handle complex linguistic phenomena like implicit sarcasm and conditional negation, particularly in aspects with highly skewed sentiment distributions like Notifications. Furthermore, expanding this framework to cross-platform reviews (e.g., iOS App Store) could enhance the generalizability of the findings. Several limitations must be acknowledged. The empirical findings are restricted to Telegram user reviews on the Indonesian Google Play Store; user behavior and linguistic styles may differ on iOS App Store platforms or across different cultural jurisdictions. Moreover, while the hybrid pipeline effectively captures direct negations and intensifiers, nuanced linguistic phenomena such as implicit sarcasm and deep conditional metaphors remain challenging without transformer-based contextual reasoning. Future research will explore knowledge-distilled compact transformer models to bridge deep semantic comprehension with industrial inference speeds, as well as cross-platform validation on iOS reviews and other application categories to establish broader generalizability.

REFERENCES

- [1] N. I. Lestari, S. M. Taib, W. Wibowo, I. A. Aziz, and M. R. Habibi, "Aspect-Based Sentiment Analysis for Mobile App Review Using Convolutional Neural Network (CNN) and Word2Vec," in *2024 IEEE 7th International Conference on Electrical, Electronics and System Engineering (ICEESE)*, IEEE, 2024, pp. 1–6.
- [2] A. Maroof, S. Wasi, S. I. Jami, and M. S. Siddiqui, "Aspect-based sentiment analysis for service industry," *IEEE Access*, vol. 12, pp. 109702–109713, 2024.
- [3] F. Meng, M. Qi, and S. Bao, "Innovative Sentiment Analysis of Mobile App Review Texts Based on MHA-MLP-Mixer," in *2024 IEEE Smart World Congress (SWC)*, IEEE, 2024, pp. 1150–1157.
- [4] R. Budianoor, S. W. Saputro, F. Abadi, R. A. Nugroho, and A. Farmadi, "Quantifying the Impact of Text Preprocessing on IndoBERT Fine-Tuning for Indonesian Informal Culinary Sentiment Analysis," *Journal of Computing Theories and Applications*, vol. 3, no. 4, pp. 564–581, 2026.
- [5] R. Obiedat *et al.*, "Sentiment analysis of customers' reviews using a hybrid evolutionary SVM-based approach in an imbalanced data distribution," *Ieee Access*, vol. 10, pp. 22260–22273, 2022.
- [6] E. Essa, K. Omar, and A. Alqahtani, "Fake news detection based on a hybrid BERT and LightGBM models," *Complex & Intelligent Systems*, vol. 9, no. 6, pp. 6581–6592, 2023.
- [7] U. Mangal, S. Mogha, and S. Malik, "Data-driven decision making: maximizing insights through business intelligence, artificial intelligence and big data analytics," in *2024 International Conference on Advances in Computing Research on Science Engineering and Technology (ACROSET)*, IEEE, 2024, pp. 1–7.
- [8] L. Zhu, M. Xu, Y. Bao, Y. Xu, and X. Kong, "Deep learning for aspect-based sentiment analysis: a review," *PeerJ Comput. Sci.*, vol. 8, p. e1044, 2022.
- [9] D. Tang, B. Qin, and T. Liu, "Document modeling with gated recurrent neural network for sentiment classification," in *Proceedings of the 2015 conference on empirical methods in natural language processing*, 2015, pp. 1422–1432.
- [10] G. Ke *et al.*, "Lightgbm: A highly efficient gradient boosting decision tree," *Adv. Neural Inf. Process. Syst.*, vol. 30, 2017.
- [11] L. Muthathohirin, "Telegram App Reviews on Playstore (400K)," Kaggle. Accessed: Aug. 26, 2026. [Online]. Available: <https://www.kaggle.com/datasets/luthfimumathathohirin/telegramm923ueede/data>
- [12] B. Wilie *et al.*, "IndoNLU: Benchmark and resources for evaluating Indonesian natural language understanding," in *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, 2020, pp. 843–857.
- [13] J. Estdale and E. Georgiadou, "Applying the ISO/IEC 25010 quality models to software product," in *European Conference on Software Process Improvement*, Springer, 2018, pp. 492–503.
- [14] J. Leskovec, A. Rajaraman, and J. D. Ullman, *Mining of massive datasets*, vol. 2. Cambridge university press New York, NY, USA, 2014.
- [15] M. Taboada, J. Brooke, M. Tofiloski, K. Voll, and M. Stede, "Lexicon-based methods for sentiment analysis," *Computational linguistics*, vol. 37, no. 2, pp. 267–307, 2011.
- [16] S. Kiritchenko, X. Zhu, and S. M. Mohammad, "Sentiment analysis of short informal texts," *Journal of Artificial Intelligence Research*, vol. 50, pp. 723–762, 2014.
- [17] M. Altalhan, A. Algarni, and M. T.-H. Alouane, "Imbalanced data problem in machine learning: A review," *IEEE access*, vol. 13, pp. 13686–13699, 2025.
- [18] A. Qazi, N. Hasan, R. Mao, M. E. M. Abo, S. K. Dey, and G. Hardaker, "Machine learning-based opinion spam detection: A systematic literature review," *IEEE Access*, vol. 12, pp. 143485–143499, 2024.
- [19] D. L. Shanthi and N. Chethan, "Genetic algorithm based hyper-parameter tuning to improve the performance of machine learning models," *SN Comput. Sci.*, vol. 4, no. 2, p. 119, 2022.
- [20] I. H. Witten, E. Frank, M. A. Hall, C. J. Pal, and M. Data, "Practical machine learning tools and techniques," in *Data mining*, Elsevier Amsterdam, The Netherlands, 2005, pp. 403–413.
- [21] E. Guzman and B. Bruegge, "Towards emotional awareness in software development teams," in *Proceedings of the 2013 9th joint meeting on foundations of software engineering*, 2013, pp. 671–674.
- [22] T. Saito and M. Rehmsmeier, "The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets," *PLoS One*, vol. 10, no. 3, p. e0118432, 2015.
- [23] R. Srinivasan and C. N. Subalalitha, "Sentimental analysis from imbalanced code-mixed data using machine learning approaches," *Distrib. Parallel Databases*, vol. 41, no. 1, pp. 37–52, 2023.

- [24] Rianto, A. B. Mutiara, E. P. Wibowo, and P. I. Santosa, "Improving the accuracy of text classification using stemming method, a case of non-formal Indonesian conversation," *J. Big Data*, vol. 8, no. 1, p. 26, 2021.
- [25] D. Tang, F. Wei, N. Yang, M. Zhou, T. Liu, and B. Qin, "Learning sentiment-specific word embedding for twitter sentiment classification," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2014, pp. 1555–1565.
- [26] S. Panichella, A. Di Sorbo, E. Guzman, C. A. Visaggio, G. Canfora, and H. C. Gall, "How can i improve my app? classifying user reviews for software maintenance and evolution," in *2015 IEEE international conference on software maintenance and evolution (ICSME)*, IEEE, 2015, pp. 281–290.
- [27] S. Kiritchenko, X. Zhu, and S. M. Mohammad, "Sentiment analysis of short informal texts," *Journal of Artificial Intelligence Research*, vol. 50, pp. 723–762, 2014.
- [28] M. Taboada, J. Brooke, M. Tofiloski, K. Voll, and M. Stede, "Lexicon-based methods for sentiment analysis," *Computational linguistics*, vol. 37, no. 2, pp. 267–307, 2011.
- [29] W. Maalej and H. Nabil, "Bug report, feature request, or simply praise? on automatically classifying app reviews," in *2015 IEEE 23rd international requirements engineering conference (RE)*, IEEE, 2015, pp. 116–125.