Online-Journals.org

iJIM International Journal of Interactive Mobile Technologies

iJIM | eISSN: 1865-7923 | Vol. 20 No. 13 (2026) | OPEN ACCESS

https://doi.org/10.3991/ijim.v20i13.60857

PAPER

# Feature Engineering and Classifier Evaluation Using Comparative NLP for Mobile Game and Educational Application Recommendation

Arar Al Tawil[1,2], Siti Hazyanti Mohd Hashim[1](✉)

[1]School of Computer Sciences, Universiti Sains Malaysia, Penang, Malaysia

[2]Faculty of Information Technology, Applied Science Private University, Amman, Jordan

sitihazyanti@usm.my

**ABSTRACT**

Mobile application (APP) reviews published on platforms such as the Google Play Store carry rich signals relevant to recommendation prediction, yet prior work has not systematically compared individual text representation and classifier combinations in this context. The present study addresses this gap by constructing an NLP-based evaluation framework that benchmarks 15 text representations drawn from five categories: (1) bag-of-words, (2) TF-IDF, (3) word embeddings, (4) transformer-based encodings, and (5) N-gram representations—paired with five machine learning classifiers across two independent datasets: 9,664 mobile game reviews and 14,344 educational APP reviews. Class imbalance is corrected using SMOTE applied exclusively to training partitions, and model interpretability is examined through random forest (RF) feature importance scores. On the game review dataset, sentiment-augmented hybrid representations outperform purely lexical approaches, whereas BERT yields the strongest results on the educational dataset. RF achieves the highest Macro-F1 in both domains. These findings confirm that no single configuration dominates universally and underline the practical necessity of domain-sensitive evaluation strategies.

**KEYWORDS**

mobile application reviews, sentiment analysis, text representation, recommendation classification, machine learning

## 1 INTRODUCTION

The mobile gaming industry has become one of the largest economic sectors in the world's entertainment industry. In 2023, the sector generated almost 100 dollars of revenue and recognized an almost 49 percent share of the more than 180 billion dollars in value of the games market with an active 3.38 billion players [1]. Today, almost everyone has a high-speed internet connection on their phone, and

Al Tawil, A., Hashim, S. H. M. (2026). Feature Engineering and Classifier Evaluation Using Comparative NLP for Mobile Game and Educational Application Recommendation. *International Journal of Interactive Mobile Technologies (iJIM)*, 20(13), pp. 18–40. https://doi.org/10.3991/ijim.v20i13.60857

Article submitted 2026-02-01. Revision uploaded 2026-04-22. Final acceptance 2026-05-07.

smartphones' hardware is improving constantly, and because of this, gaming has gone from being a hobby to the most popular digital activity. The introduction of dedicated video games for esports, such as PUBG Mobile and Mobile Legends: Bang Bang and Arena of Valor have boosted the mobile platform even more. These games have been developing competitive ecosystems with professional leagues, media partnerships, and large audiences [2]. As a result, operators of platforms have found that automated, data-driven recommendation systems are a practical necessity of development. User-generated reviews in mobile distribution platforms can be considered one of the most informative signals in recommendation tasks. Studies show that most digital consumers rely on peer reviews before downloading an app. In particular, consumers are more likely to look at peer reviews for free-to-play games, where they have no financial cost against a bad experience [3]. The tens of millions of reviews racked up on Google Play and the Apple App Store are a treasure trove of attitudinal data [4]. However, this information is unstructured, requiring sophisticated automated methods for sentiment classification and recommendation prediction that allow one to glean actionable insights from it. The recommendation architectures that have been put in place mainly depend upon collaborative filtering and content-based filtering techniques that work on structured metadata, i.e., genre labels, aggregate star ratings, download statistics, etc. [5] While such approaches may serve as helpful starting points, they fail to account for the subtle, multi-dimensional quality signals embedded in the language of user reviews. Natural language processing and supervised machine learning provide a complementary way to let models learn non-linear mappings from textual features to recommendation labels [6]. Prior research showed that transformer-based representations can efficiently classify sentiment on the review corpus [7]. The framework established in this study extends beyond gaming to mobile learning environments. These environments include language-learning platforms, gamified assessment tools, and multi-subject tutoring applications (APPs). The automated analysis of reviews can similarly be used in these environments for quality monitoring and content recommendation in educational technology. Despite the use of NLP pipelines, the analysis of mobile game reviews is complex and challenging. To assess a mobile game, various dimensions need to be evaluated simultaneously: the touch interface responsiveness, frame-rate stability, network latency in multiplayer mode, the perceived fairness of free-to-play monetization mechanisms, and competitive integrity [8]. The structural attributes of mobile platforms have additional restrictions: shorter and more frequent play sessions, integration with social networks, and cross-platform compatibility. Each of these restrictions leaves a clearly detectable mark on the language and sentiment of reviews [9]. Concerns regarding the quality of matchmaking, responsiveness of balance patches, efficiency of anti-cheat systems, and quality of spectator infrastructure are also raised by players within the esports segment [2]. Hence, these domain features make a systematic and empirical comparison of feature engineering strategies relevant and timely. Several important gaps still exist despite the growing body of research on game analytics and review-based recommendation. Most existing studies focus on the PC or console platforms, which limits their applicability in the mobile context. Domain adaptation is not generally used in NLP representations; aspect-level sentiment analysis on quality dimensions specific to mobile games is not yet developed [10]. An insufficient number of empirical comparisons of alternative feature engineering strategies are done within one common experimental set-up. Moreover, user review datasets usually suffer from class imbalance, wherein the overwhelming

majority of user reviews is positive. The present study seeks to address these shortcomings through the following contributions:

- A systematic comparative evaluation of 15 text representation methods applied to mobile APP user reviews, spanning classical frequency-based methods (Bag-of-Words, TF-IDF, n-grams), character-level representations, distributed word embeddings (Word2Vec, FastText, GloVe, Doc2Vec), transformer-based sentence encodings (Sentence-BERT), and domain-specific sentiment-enhanced hybrid representations.
- Cross-classification benchmarking of five supervised learning algorithms: logistic regression, random forest (RF) [15], gradient boosting, Naïve Bayes, and decision tree (DT) under fixed hyperparameter settings to isolate the effect of feature engineering from that of algorithmic configuration.
- APP of SMOTE to correct class imbalance by generating synthetic samples exclusively within the training partitions, ensuring that test-set evaluation reflects realistic class prevalence.
- Cross-domain generalization analysis on recommendation prediction performance over two independently collected datasets—mobile game reviews and educational APP reviews—aimed at assessing the extent to which optimal representation strategies are transferable across APP domains.

## 2 RELATED WORKS

The works directly pertinent to this study span game recommendation systems, sentiment and opinion mining, text representation learning, class imbalance handling, and model interpretability.

Game recommendation research has evolved significantly with the rise of digital distribution platforms. In a survey of review-based recommendation systems that appeared between 2015 and 2022, Hasan et al. [5] note a definitive shift in the field away from bag-of-words and topic-modeling approaches and towards transformer-based encoders for preference extraction. The study showed that the cold-start problem that has plagued collaborative filtering will always be there. This makes text-based content models the most promising alternative. Zhang et al. [4] explored the automatic classification of game reviews using language models with domain adaptation and found that domain-adapted representations invariably outperformed generic pre-trained alternatives, providing direct rationale for the adopted domain-sensitive design in the present study. More and more scholars are doing sentiment analysis of APP reviews. A thorough survey of opinion mining methods such as lexicon-based, traditional machine learning, and transformer-based architectures was carried out by Wankhade et al. [6]. The mobile APP reviews were rendered an upcoming research topic requiring specialized handling. Yu et al. proposed a review analysis framework for esports using Latent Dirichlet Allocation and sentiment analysis, identifying monetization fairness, matchmaking quality, and competitive balance aspects to be most strongly associated with player sentiment; these findings contributed to the construction of the domain-specific aspect lexicons used in this study. According to the works of Jaiswal et al., BERT captured the contextual relationship between words accurately within long reviews. Further, Text-CNN worked better for short-length texts. Overall, no single model was able to outperform across the entire length distribution of reviews. The rationale for the 15-representation evaluation carried out here is supported by this finding. Many analyses have

been done on ABSA regarding mobile APP reviews from multiple sources. As per the findings of Alturayeif et al. [8], it was determined that the BERT-based model displayed greater efficiency as compared to traditional classifiers for aspect polarity tasks. However, for aspect category tasks, traditional machine learning methods still perform competitively. This nuance ultimately influenced our choice of classifier in the present study. The transformer-based benchmark framework (Representation R15) for this study was the Sentence-BERT architecture from [10]. This architecture creates embeddings of sentences that are semantically meaningful and uses the structure of a Siamese network, which reduces computational cost. It is built on the original BERT model from [7]. This study was designed through the convergence of scholarly literature on mobile learning, digital gaming, and, more importantly, the practical convergence of the two domains. Samala et al. [27] conducted a bibliometric analysis of the field in the mobile learning APPs in the higher education domain under PRISMA guidance around the years spread over 2007 to 2021 and again till 2023. They found that the research output of the field witnessed a steady increase with the passage of years. Also, it was found out that personalized content delivery and learner engagement were the common priorities that are referred to in literature and also the same challenges that automated review analysis can address in the educational apps ecosystem. In a paper published in the Educational Technology journal, Papadakis and Karakose [28] examined the relationship between gamification and learning outcomes. The authors conducted a review concerning the evidence base, which showed that the use of game mechanics when well-implemented by educators can enhance motivation and achievement in students. However, superficial game mechanics that don't adequately align incentive structures with pedagogical goals won't be impactful. This viewpoint throws clarity as to why educational app reviews have the same kind of structural quality signals as games. Despite their different vocabularies and evaluative criteria, the structural quality similarities are not surprising. Blumberg and his collaborators [29] further investigate the cognitive and motivational mechanisms through which children learn in digital game environments. The study draws attention to challenge calibration and immediate feedback mechanisms, which are also central to the adaptive design of many educational APPs and that users articulate explicitly in their review texts. These pieces of research suggest that mobile games and educational apps occupy similar locations on a spectrum of interactive, feedback-rich digital experiences. This provides a foundation for expecting that NLP frameworks designed for one will be transferable to the other, given the appropriate modifications. Three recent contributions deserve comparison with the present study. Samala and others (2007 to 2023) performed a bibliometric assessment of mobile learning literature. They note that recommendation and personalization are constantly missing from papers but do not evaluate a classifier empirically. An investigation was performed by Papadakis and Karakose [28] on gamification and student achievement quality evidence to establish motivational value both without a quantitative text classification baseline. Blumberg and colleagues [29] aimed to contribute to our understanding of game-based learning, focusing on development, but did not consider automated quality assessment from user reviews. Collectively, these works establish the pedagogical importance of mobile APP quality but do not offer a computational framework for predicting recommendation outcomes from user-generated text. The current study taken in this work aims to fill this gap, and it will constitute the first systematic cross-domain comparison of 15 NLP representations for mobile APP recommendation prediction following a corrected protocol accounting for class imbalance.

Imbalance is a well-known challenge in review-based classification tasks, as user corpora are usually biased toward positive ratings. Classifiers that are traditionally trained on this classification task have a pronounced bias towards the majority class in imbalanced distributions [11]. As a result, their overall accuracy gets inflated at the expense of the minority class sensitivity. Dablain et al. [13] developed DeepSMOTE for synthesizing minority-class samples. Their results showed that learning the underlying manifold structure instead of linearly interpolating in the feature space significantly improves the quality of synthesized samples. Thus, SMOTE-based oversampling can be taken as a principled default. Leevy et al. [17] confirmed through systematic investigation that standard SMOTE with $k = 5$ nearest neighbors is competitive with more complex variants when applied to text feature vectors. Consequently, in the present study, SMOTE is applied, but only to the training partitions following the stratified splitting, so as to not contaminate test-set evaluation with synthetics. Interpreting models has become a vital topic of investigation as well as a regulatory issue. SHAP [14] entails a unified game-theoretic framework that explains the contribution of each feature towards a given prediction. The RF algorithm offers global estimates of feature importance for free, via the construction of the ensemble [15]. Aria et al. [25] showed that the mean decrease in impurity and permutation importance ranking lead to the same conclusion for high-dimensional text features. Thus, using the default feature importance may also be appropriate in this case. Text representation is the main method of the research. Word2Vec, which is the seminal distributed word embedding technique, was presented by Mikolov et al. [21]. FastText [22] built upon this work to use character N-grams for overcoming issues pertaining to morphological variation. GloVe [23] uses factorization of word co-occurrence matrices to capture global statistical structure. Doc2Vec [24] represents an extension of the embedding paradigm to the document level. Insights specific to the domain have been drawn from Hamari et al. [9], who found that monetization fairness is the strongest predictor of player recommendation intent, and Khalid et al. [18], who established that issues related to performance, functionality, and updates account for most issues within Google Play reviews. These insights were used in creating mobile-specific feature sets. All experiments relating to classification were implemented using scikit-learn [19]. Recent studies in iJIM have examined NLP and machine learning in mobile contexts. Tegegnie [30] proposed a multi-task framework for mining mobile app reviews, integrating sentiment classification, feedback categorization, and rating prediction using deep learning. Ananthi [31] applied hybrid CNN-LSTM models to classify student sentiments from learning platform reviews, achieving 97% accuracy. Bousalem et al. [32] developed an AI-based recommendation system for mobile educational APPs using student classification and performance prediction. The above literature synthesis shows that the present study will fill the following five gaps: (1) there is no systematic evaluation consisting of multiple representation comparisons in the mobile games domain; (2) domain adaptation is considered sufficient; (3) class-imbalance correction is overlooked; (4) interpretability is under-utilized; and (5) there is no cross-domain validation to test representation transferability.

## 3 METHODOLOGY

This section describes the research procedures illustrated in Figure 1 that were followed to design, implement, and evaluate the proposed mobile APP recommendation framework. The methodology is structured around four principal components:

(1) collection and characterization of the experimental datasets, (2) systematic construction of 15 text representation schemes, (3) selection and configuration of five classification algorithms, and (4) the experimental protocol and evaluation criteria.

## 3.1 Dataset description

To assess the multi-domain APP of the proposed framework, two separate datasets were created using Google Play Store reviews for mobile games and Ed-Tech Apps. Both datasets were extracted programmatically using the google-play-scraper library [12] and are publicly available on Kaggle for reproducibility. In reference to the labeling specification, the reviews that received ratings (stars) of four and five were assigned the label Recommended (label = 1), while the reviews that received rating (stars) of one and two were assigned the label Not Recommended (label = 0). All three-star reviews were omitted from the datasets because of the ambiguous recommendation signal they provide, according to a labeling scheme [3]. With a total of 10,000 reviews taken during January and February 2026, 2,500 are per APP, from four free-to-play mobile games: Subway Surfers, Candy Crush Saga, Clash of Clans, and PUBG Mobile [11]. After removing neutral reviews, we ended up with 9,664 observations, of which 88.8% were Recommended and 11.2% were Not Recommended and had an average review length of 48 characters [4]. Dataset 2 contains around 15,000 reviews sampled from Duolingo, Khan Academy, and Kahoot. The reviews were collected from February 2022 to February 2026 at 5,000 reviews per APP [26]. After filtering, a total of 14,344 observations were used. Of those, 84.3% are Recommended and 15.7% are Not Recommended. The mean review length is 67 characters [8]. Both the datasets have noticeable class imbalance; we apply SMOTE [13] only on the training splits. Table 1 provides a detailed comparison summary.

**Table 1.** Comparative summary of experimental datasets

| Attribute | Dataset 1: Mobile Games | Dataset 2: Educational Apps |
|---|---|---|
| Data Source | Google Play Store | Google Play Store |
| APPs | Subway Surfers, Candy Crush Saga, Clash of Clans, PUBG Mobile | Duolingo, Khan Academy, Kahoot |
| No. of APPs | 4 | 3 |
| Total Raw Reviews | 10,000 | 15,000 |
| Reviews per APP | 2,500 | 5,000 |
| After Filtering (score ≠ 3) | 9,664 | 14,344 |
| Collection Period | Jan–Feb 2026 | Feb 2022–Feb 2026 |
| Recommended (Label = 1) | 8,577 (88.8%) | 12,095 (84.3%) |
| Not Recommended (Label = 0) | 1,087 (11.2%) | 2,249 (15.7%) |
| Imbalance Handling | SMOTE [13] | SMOTE [13] |
| Avg. Review Length (chars) | 48 | 67 |

Figure 1 illustrates the per-game recommendation rates in Dataset 1, revealing that Subway Surfers yields the highest rate (92.1%) while PUBG Mobile records the lowest (80.9%). Figure 2 depicts the class distribution before and after SMOTE

resampling for Dataset 1, confirming that the synthetic oversampling procedure balances the minority class from 12.2% to 50% in the training partition without modifying the test set distribution. The equivalent distributions for Dataset 2 are presented in Figure 8.

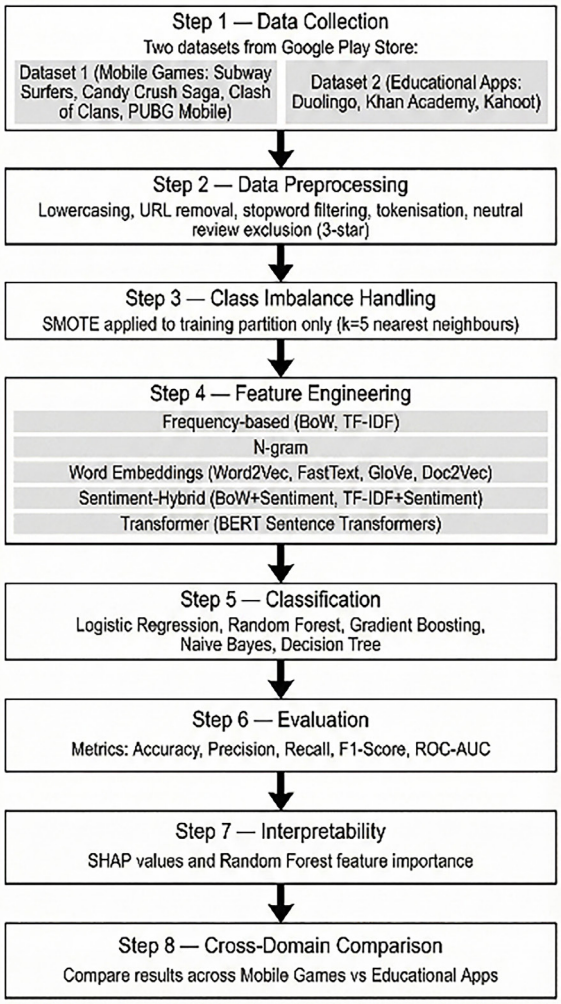


**Fig. 1.** End-to-end methodology framework for mobile APP recommendation prediction using NLP and machine learning

The two datasets suffer from severe class imbalance; the majority class (Recommended) accounts for 88.8% of the observations in Dataset 1 and 84.3% in Dataset 2. If this skew is not corrected, the classifiers will prefer the majority class in a systematic way. Thus, the accuracy estimates will be inflated but will come at a cost of lower sensitivity on the minority class. The minority class is the type of class that has more practical interest for quality monitoring and churn prevention. To reduce this effect, SMOTE [13] was only applied to the training partition of each representation-specific experiment following a stratified train-test split. The SMOTE technique employs k-nearest neighbors (k = 5) to generate synthetic minority-class instances based on existing samples in the feature space. This method balances the training distribution while retaining the original class proportions of the held-out test set and ensuring that the evaluation metrics reported are indicative of real deployment conditions.

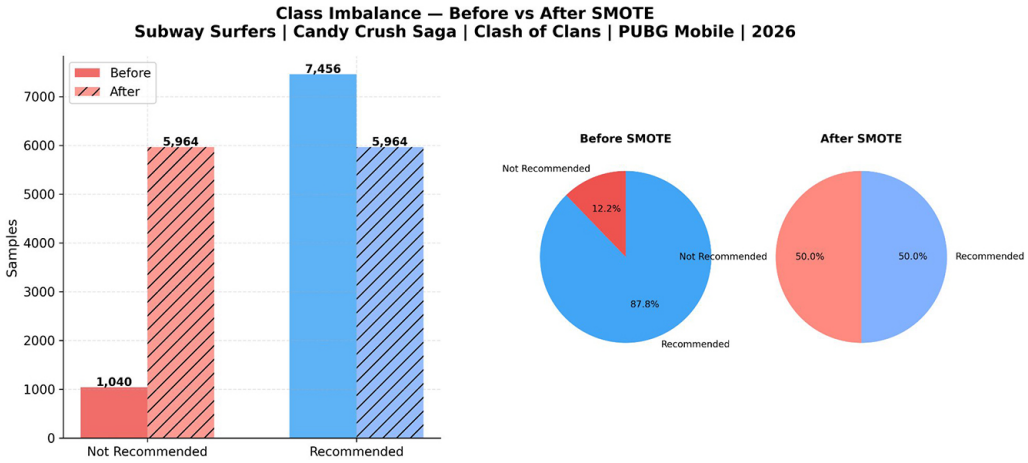


**Fig. 2.** Class imbalance before and after SMOTE dataset 1 (2026)

*Note:* SMOTE resampling increases the minority class from 1,040 samples (12.2%) to 5,964, achieving a balanced 50/50 training distribution while preserving real-world imbalance in the test partition.

## 3.2 Feature engineering framework

A systematic assessment of popular text representation methods for the evaluation of user reviews to recommend predictions was undertaken over 15. The representations cover a wide range of techniques, spanning classical sparse frequency-based methods to dense neural embedding-based schemes and hybrid representations that combine textual features with domain sentiment signals, thus allowing a thorough empirical comparison of current NLP paradigms [6]. Text preprocessing flagged for spam words that contained less than three characters, which were also removed. According to the methods of embeddings (R4, R10–R15), morphological stemming could not be done to preserve the lexical variation dependence of the distributional and subword-level models. To ensure the comparability of methods, all representations were processed by this pipeline [8]. Feature set of sentiment. In the hybrid schemes (R8–R11), each review is assessed for seventeen sentiment features relating to the individual domain, which are joined to the main text representations. The features include: (i) general sentiment intensity scores with compound and positive, negative, and neutral polarity components; (ii) subjectivity and polarity estimates from a lexicon-based analyzer; (iii) five emotion-category scores from a domain-adapted affective lexicon; and (iv) five aspect-based sentiment scores for domain-relevant quality dimensions (gameplay or learning efficacy, graphics or interface design, technical performance, monetization or motivation, multiplayer or content quality) along with (v) surface features: positive word count, negative word count, review character length, total word count, exclamation mark count, question mark count, etc. The domains of gaming and education were each separately calibrated for the aspect lexicons [9]. GloVe portrayal. We used the pre-trained GloVe vectors of 100 dimensions 23 to obtain the representation for R13. The document vectors were formed by taking the mean of the word vectors before L2 normalization and z-score standardization of standardscaler. For R13, a default RF classifier was replaced with an SVM model using an RBF kernel (C = 1.0, gamma = 'scale'). SVM with standardization resolves the near-random ROC-AUC performance when tree-based ensemble methods are used on dense L2-normalized embeddings [15].

Table 2 provides a complete enumeration of all 15 representations, their methodological descriptions, and the classifier assigned to each.

**Table 2.** Summary of text representation techniques (15 evaluated)

| # | Representation | Description | Classifier |
|---|---|---|---|
| 1 | Bag of Words (BoW) | Token frequency counts over the full vocabulary | RF |
| 2 | TF-IDF Unigram | Term frequency–inverse document frequency, unigram tokens [20] | RF |
| 3 | TF-IDF Bigram | TF-IDF extended with unigram and bigram features [20] | RF |
| 4 | Word2Vec | Averaged word embeddings; dim = 100, window = 5 [21] | RF |
| 5 | N-grams | TF-IDF applied over bigram phrases | RF |
| 6 | Char N-grams | Character-level n-gram TF-IDF, n ∈ {2,3,4} | RF |
| 7 | Hashing Vectorizer | Feature hashing with 2,000 buckets | RF |
| 8 | BoW + Sentiment | BoW concatenated with 17 domain sentiment features | RF |
| 9 | TF-IDF + Sentiment | TF-IDF bigram concatenated with 17 sentiment features | RF |
| 10 | Word2Vec + Sentiment | Word2Vec embeddings concatenated with 17 sentiment features | RF |
| 11 | Combined All | TF-IDF bigram + Word2Vec + 17 sentiment features | RF |
| 12 | FastText | Subword skip-gram embeddings; dim = 100, wordNgrams = 2 [22] | RF |
| 13 | GloVe + SVM | L2-normalized GloVe vectors (dim = 100) + StandardScaler [23] | SVM |
| 14 | Doc2Vec | Distributed memory paragraph vectors; dim = 100, dm = 1 [24] | RF |
| 15 | BERT (Sentence Transformers) | all-MiniLM-L6-v2 sentence embeddings; dim = 384 [10] | RF |

## 3.3 Classification algorithms

To evaluate the performance of each text representation, five well-known supervised classifiers were selected; they were linear, probabilistic, ensemble, and tree-based learning approaches. All algorithms utilized scikit-learn 1.3 [19] and fixed random seeds for reproducibility. Logistic regression generates a model using an input feature's linear combination to examine the log-odds of class membership. Even though LR is parametrically simple, it performs well on high-dimensional sparse representations of text and yields interpretable coefficient weights that clearly showcase the directional influence on the outcome of each feature [6]. An L-BFGS solver with a limit of 1,000 iterations was used. RF is a bagging ensemble of decision trees (DT) wherein trees are trained on bootstrap samples of the training set with a random subset of features considered at each split node. RF is strong against unfitting features; RF resists overfitting and natively generates feature importance estimates, the interpretability mechanism adopted in this investigation [25]. An ensemble of 100 trees was generated using the Gini impurity criterion. Gradient Boosting (GB) [16] is an ensemble method, which builds sequentially, that fits shallow DTs to the negative gradient of a differentiable loss function, correcting the residual errors of the predecessor's ensemble stage by stage. GB typically achieves high predictive accuracy and is effective for modeling complex and non-linear feature dependencies. The estimators were set to 100 and the learning rate to 0.1. Under a conditional

independence assumption, the multinomial NB algorithm applies Bayes' theorem. The multinomial NB model counts features, which are multinomially distributed. Although the independence assumption generally does not hold in practice, nonetheless, NB is efficient and competitive on sparse count-based representations of text [6]. Laplace smoothing (alpha = 1.0) was applied to avoid 0 frequencies.

The DT forms binary partitions in the feature space through recursive axis-aligned splits. Also, these splits help to maximize the information gain at each internal node. DT builds simple, interpretable, rule-based models and is a suitable baseline for benchmarking the ensemble techniques. The maximum depth restriction was not set in the use of the Gini criterion. The five algorithms' hyperparameters were kept constant across all 15 representations to separate the effects of feature engineering and algorithmic configuration. In obtaining the GloVe representation (R13), we replaced the default RF classifier with a support vector machine with an RBF kernel (see Section 3.2). Table 3 presents a summary of all the algorithms having fixed configurations.

**Table 3.** Classification algorithms and hyperparameter configurations

| Algorithm | Key Hyperparameters |
|---|---|
| LR | max_iter=1000, solver=lbfgs, random_state=42 |
| RF | n_estimators=100, criterion=gini, random_state=42 |
| GB | n_estimators=100, learning_rate=0.1, random_state=42 |
| NB | alpha=1.0 (Laplace smoothing) |
| DT | criterion=gini, max_depth=None, random_state=42 |

## 3.4 Experimental protocol and evaluation metrics

Using a standard experimental protocol for all representation-classifier combinations and both datasets allows results to be rigorous, reproducible, and directly comparable. Stratified random sampling was used to partition each dataset to an 80% training set and a 20% held-out test set, ensuring the original number of communities was preserved in each set. Stratification protects against optimistically biased estimates of minority-class performance possibly caused by a homogeneous test partition [11]. After the stratified split, the SMOTE [13] took place for the training partition, with k = 5 nearest neighbors; meanwhile, no synthetic samples were added to the test set. This design consists of having the distribution of the classes of the test partition be the same as that of the real world. Thus, all the reported metrics correspond to such realistic deployment and not to some artificially balanced one [17]. Two consecutive trials were performed on each dataset. All 15 text representations were evaluated in the representation comparison experiment using their associated classifiers—RF for R1–R12 and R14–R15, SVM for R13 training on the SMOTE-balanced training partition and evaluation on the original imbalanced test partition. This experiment examines the contribution of the feature engineering stage in isolation and identifies the most effective representation for each APP domain, independently. The comparison of the classifiers used the representation with the highest F1-score from the first experiment, which was fixed. All five classifiers were trained on the SMOTE-balanced training partition, and performance was

evaluated on the held-out test set. This separates how well the classifier works from how it represents things, allowing us to identify the best learning algorithm. The model's effectiveness was assessed with the help of five complementary metrics. This model assessment is determined by the parameters concerning the model's effectiveness, accuracy (Eq. 1), precision (Eq. 2), recall (Eq. 3), F1-score (Eq. 4), and ROC-AUC (Eq. 5) [11]. The trivial majority-class predictor does attain high overall accuracy, but this is not useful for the minority class of practical interest. This indicates that accuracy is not sufficient under class imbalance. To this end, we select the main ranking criteria as the Macro F1-score. The Macro-F1 Score averages F1 across both classes equally. It does not take class prevalence into account. Thus, minority-class performance is not masked. The threshold-independent performance measurement of the classifier's discriminative power across all thresholds is reported by ROC-AUC.

$$\text{Accuracy} = (TP + TN)/(TP + TN + FP + FN) \tag{1}$$

$$\text{Precision} = TP/(TP + FP) \tag{2}$$

$$\text{Recall} = TP/(TP + FN) \tag{3}$$

$$\text{F1-Score} = 2 \times (\text{Precision} \times \text{Recall})/(\text{Precision} + \text{Recall}) \tag{4}$$

$$\text{ROC-AUC} = \int \text{TPR}\, d(\text{FPR}) \tag{5}$$

$$\text{where TPR} = TP/(TP + FN),\ \text{FPR} = FP/(FP + TN)$$

where TP, TN, FP, and FN are true positive, true negative, false positive, and false negative, respectively. The primary ranking criterion throughout the work is macro-F1 (the unweighted average of per-class F1s), because it gives equal weight to both the majority and minority classes, making it less sensitive to class imbalance. We also report Majority-class F1, for reference, along with Minority-class F1, which directly measures accuracy on the "Not Recommended" class. The measure ROC-AUC (Eq. 5) is a threshold-independent measure of a classifier's ability to discriminate across all thresholds. Python 3.10 with scikit-learn 1.3 [19], sentence-transformers [10], and imbalanced-learn were used for all experiments. We fix all random seeds (seed = 42) on all stochastic operations, for instance, splitting of data, synthesis via SMOTE [13], training for Word2Vec [21] and Doc2Vec [24], and classifier initialization, etc., to ensure reproducibility. The Kaggle cloud computing platform hosted the computations, offering GPU-enabled sessions with Sentence-BERT encoding [10] and FastText [22] training.

## 4 RESULTS AND DISCUSSION

This section presents and interprets the experimental results obtained from two datasets, i.e., mobile games (Dataset 1) and educational APPs (Dataset 2). The section continues by sharing the results on representation comparison experiments (see Section 4.1) and classifier comparison experiments (see Section 4.2). According to Macro-F1 metrics, all results were evaluated and ranked on the original imbalanced test partitions after SMOTE-balanced training. The graphical representation of the class distribution before and after the implementation of SMOTE has been shown in Dataset 1 and Dataset 2 through Figures 2 and 8, respectively.

### 4.1 Representation comparison

Table 4 gives the results for all 15 text representations on Dataset 1 (Mobile Games). There is a clear performance gradient, with sentiment-augmented hybrid representations consistently falling in the upper tier and plain bag-of-words methods in the lower tier. When using Macro-F1 as the evaluative metric, the best performance was achieved by (the best performing ensemble) TF-IDF + Sentiment, which attained the (Macro-F1 = 0.7987, Minority-F1 = 0.6452). The second-best performance is by BERT Sentence front-end transforms (which received Macro-F1 = 0.7865). The third-best performance is by BoW + Sentiment, which received Macro-F1 = 0.7743. GloVe + SVM is not covered in our analysis as it did not detect any of the minority classes (Minority-F1 = 0.000, ROC-AUC = 0.500), i.e., it is no better than random chance, so it is not useful. The superiority of representations augmented with sentiment over purely lexical or embedding-based counterparts suggests that the domain-specific sentiment signals are carrying complementary discriminative information that cannot be captured by either surface-form counts or distributed embeddings independently. The BERT method achieved the greatest ROC-AUC (0.8967), while the BoW + Sentiment featured on recall (0.9665), which was the most important for minority class detection. The F1 score of char N-grams (F1 = 0.9366) outperformed the pretrained word-level embeddings (Word2Vec: F1 = 0.9073; GloVe: F1 = 0.9183). This shows that character-level subword patterns learned by the model encode meaningful domain-specific morphological features in short mobile game reviews. N-grams showed a Macro-F1 of 0.6840, which was the lowest. Moreover, plain BoW and Word2Vec are ranked 0.6917, which are also at the lower level, confirming that the raw token frequency counts are not sufficient for reliable recommendation classification in this domain.

**Table 4.** Representation comparison results mobile games dataset (2026)

| Representation | Accuracy | Precision | Recall | Macro-F1* | Minority-F1 | Majority-F1 | ROC-AUC |
|---|---|---|---|---|---|---|---|
| TF-IDF + Sentiment | 0.9159 | 0.9482 | 0.9564 | **0.7987** | 0.6452 | 0.9523 | 0.8706 |
| BERT (Sentence-T) | 0.9100 | 0.9460 | 0.9517 | **0.7865** | 0.6241 | 0.9489 | 0.8967 |
| BoW + Sentiment | 0.9141 | 0.9353 | 0.9692 | **0.7743** | 0.5967 | 0.9519 | 0.8617 |
| Combined All | 0.8806 | 0.9523 | 0.9095 | **0.7551** | 0.5797 | 0.9304 | 0.8624 |
| Word2Vec+Sentiment | 0.8706 | 0.9511 | 0.8988 | **0.7412** | 0.5582 | 0.9242 | 0.8616 |
| Doc2Vec | 0.8882 | 0.9340 | 0.9390 | **0.7354** | 0.5343 | 0.9365 | 0.8484 |
| Char N-grams | 0.8882 | 0.9328 | 0.9403 | **0.7331** | 0.5297 | 0.9366 | 0.8629 |
| Hashing | 0.8559 | 0.9561 | 0.8760 | **0.7307** | 0.5471 | 0.9143 | 0.8730 |
| FastText | 0.8700 | 0.9368 | 0.9135 | **0.7186** | 0.5121 | 0.9250 | 0.8704 |
| TF-IDF (unigram) | 0.8465 | 0.9522 | 0.8686 | **0.7157** | 0.5229 | 0.9085 | 0.8716 |
| TF-IDF (bigram) | 0.8435 | 0.9527 | 0.8646 | **0.7132** | 0.5199 | 0.9065 | 0.8705 |
| Word2Vec | 0.8476 | 0.9351 | 0.8881 | **0.6917** | 0.4725 | 0.9111 | 0.8392 |
| BoW | 0.8335 | 0.9442 | 0.8613 | **0.6917** | 0.4826 | 0.9008 | 0.8291 |
| N-grams | 0.8682 | 0.9210 | 0.9296 | **0.6840** | 0.4428 | 0.9253 | 0.7949 |
| Dummy classifier (majority) | | | | 0.4702 | 0.0000 | ~0.940 | |

*Notes:* *Macro-F1 is the primary evaluation metric; representations are ranked by Macro-F1.

Figures 3 and 4 visualize the Macro-F1 and Minority-F1 rankings, respectively, for Dataset 1, with the dummy classifier baseline indicated by a dashed red line. Figure 5 presents a side-by-side comparison of Majority-F1 and Macro-F1 for each representation, exposing the degree to which the majority-class metric overstates performance under class imbalance. Figure 6 maps the precision-recall trade-off across all 15 representations, and Figure 7 ranks them by ROC-AUC, which captures discriminative capacity independently of the classification threshold. Representation results for Dataset 2 (Educational APPs) are shown in Table 5. BERT achieved both the highest Macro-F1 (0.8181, Minority-F1 = 0.6964) and the highest ROC-AUC (0.9190) in this dataset. This is in stark contrast with the game's dataset, where the sentiment-augmented representations led. This movement displays the greater semantic richness and lengths of educational reviews, which favor sentence-level context representations over bag-of-words or shallow hybrid schemes. The performance of FastText showed the fourth-best performance overall (Macro-F1 = 0.7876), due to the subword-level performance on the pedagogic vocabulary. Sentiment addition also continues to work well as we see BoW + Sentiment also being competitive with F1 = 0.9357. The F1 score of bigrams collapsed to only 0.7073 on educational reviews, which is significantly lower compared to that of the game's dataset. The OOV terms due to the clear difference between gaming and educational bigram vocabularies are the culprits behind this. Frequency-based N-gram models are especially susceptible to this.

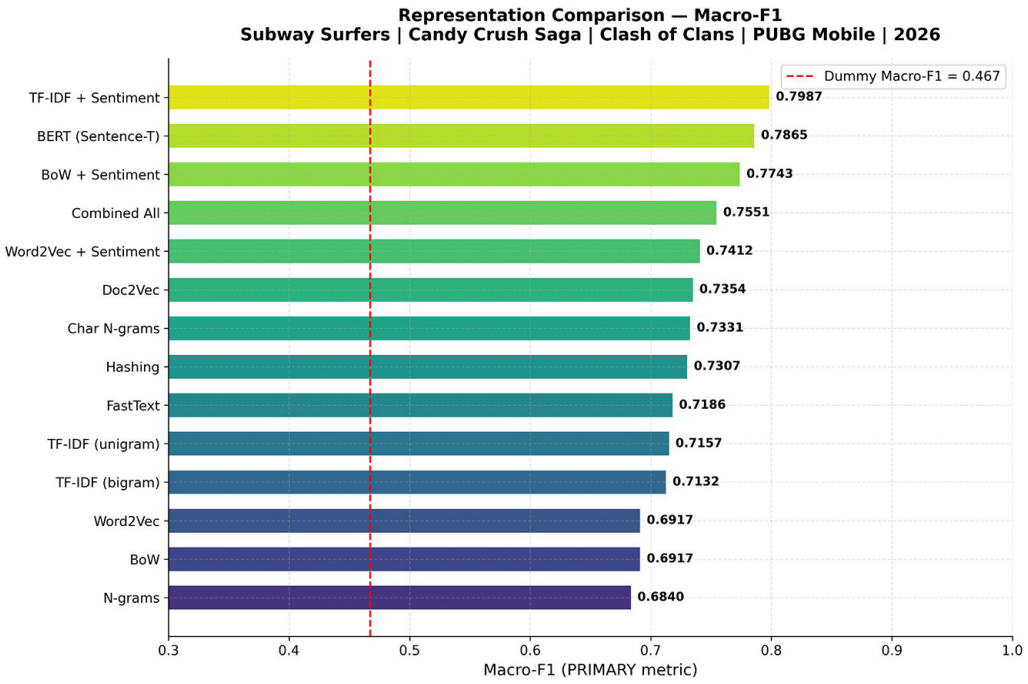


**Fig. 3.** Representation comparison by Macro-F1 dataset 1 (2026)

*Notes:* TF-IDF + sentiment leads with Macro-F1 = 0.7987. The red dashed line marks the dummy classifier baseline (0.467); all 15 representations substantially exceed this threshold, confirming genuine learning beyond majority-class bias.

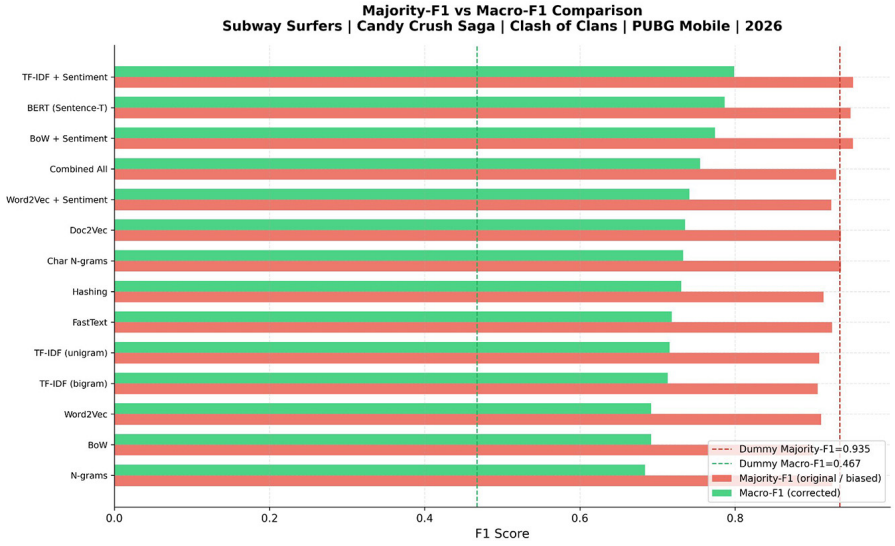


**Fig. 4.** Majority-F1 versus Macro-F1 per representation dataset 1 (2026)

*Notes:* The gap between the two metrics reveals how majority-class F1 systematically overstates model performance. The corrected Macro-F1 (green) is substantially lower than Majority-F1 (red) for every representation, justifying the revised evaluation protocol.

**Table 5.** Representation comparison results educational APPs dataset (2022–2026)

| Representation | Accuracy | Precision | Recall | Macro-F1* | Minority-F1 | Majority-F1 | ROC-AUC |
|---|---|---|---|---|---|---|---|
| BERT (Sentence-T) | 0.8994 | 0.9408 | 0.9387 | **0.8181** | 0.6964 | 0.9397 | 0.9190 |
| TF-IDF + Sentiment | 0.8861 | 0.9358 | 0.9273 | **0.7969** | 0.6622 | 0.9315 | 0.9003 |
| BoW + Sentiment | 0.8903 | 0.9241 | 0.9464 | **0.7903** | 0.6454 | 0.9351 | 0.8937 |
| FastText | 0.8732 | 0.9430 | 0.9028 | **0.7876** | 0.6528 | 0.9225 | 0.9013 |
| Combined All | 0.8554 | 0.9496 | 0.8732 | **0.7726** | 0.6354 | 0.9098 | 0.8834 |
| Word2Vec+Sentiment | 0.8554 | 0.9487 | 0.8741 | **0.7720** | 0.6340 | 0.9099 | 0.8788 |
| TF-IDF (unigram) | 0.8448 | 0.9525 | 0.8569 | **0.7633** | 0.6244 | 0.9022 | 0.9006 |
| Hashing | 0.8569 | 0.9318 | 0.8941 | **0.7593** | 0.6061 | 0.9126 | 0.8763 |
| Doc2Vec | 0.8600 | 0.9264 | 0.9041 | **0.7573** | 0.5993 | 0.9152 | 0.8746 |
| TF-IDF (bigram) | 0.8361 | 0.9510 | 0.8473 | **0.7532** | 0.6101 | 0.8962 | 0.8965 |
| Char N-grams | 0.8676 | 0.9152 | 0.9273 | **0.7526** | 0.5840 | 0.9212 | 0.8700 |
| Word2Vec | 0.8471 | 0.9265 | 0.8873 | **0.7436** | 0.5806 | 0.9065 | 0.8749 |
| BoW | 0.8194 | 0.9328 | 0.8446 | **0.7221** | 0.5576 | 0.8865 | 0.8568 |
| N-grams | 0.6099 | 0.9474 | 0.5643 | **0.5613** | 0.4152 | 0.7073 | 0.7826 |
| Dummy classifier (majority) | | | | 0.4580 | 0.0000 | ~0.915 | |

*Notes:* *Macro-F1 is the primary evaluation metric; representations are ranked by Macro-F1.

Figure 5 presents the Macro-F1 ranking for Dataset 2, with the dummy classifier baseline indicated by a dashed red line. Figure 8 (dataset characterization) is presented in Section 3.

## 4.2 Classifier comparison

The results of comparing the classifier effectiveness for Dataset 1 in Table 6 utilized the best representation, TF-IDF + Sentiment (Macro-F1 = 0.7987). Random Forest, due to

its ensemble diversity and power to manage residual class imbalance in the test partition, achieved the highest Macro-F1 (0.8006) and Minority-F1 (0.6485). The Not Recommended category is arguably the most important classification for mobile APP quality monitoring and churn prevention from an industry point of view. Figure 6, the confusion matrix of RF, shows that it correctly identifies 131 of the 208 "Not Recommended" reviews (recall = 0.630). Thus, 77 negative reviews would remain undetected that would reach recommendation pipelines unfiltered. The recall score for DT is 0.546, and for Logistic Regression is 0.608. Naive Bayes possesses the lowest recall of the minority class (0.387), confirming its unsuitability for imbalanced recommendation tasks. The Logistic Regression model has the highest ROC-AUC (0.9014) score. Practitioners can take advantage of this score to improve the recall of "Not Recommended" by lowering the classification threshold. This trade-off is appropriate in deployment contexts with a focus on quality. Although its Macro-F1 is competitive (0.7430), DT's ROC-AUC is low (0.7217), which means it yields poorly calibrated probabilities, limiting threshold-based optimization. GB, which achieved a Macro-F1 score of 0.7558, secured the third rank, while Naive Bayes, which got a 0.6428 for the macro-F1 score, achieved a fifth rank, mostly because of the high conditional independence violation present in hybrid representations due to sentiment and frequency features being correlated.

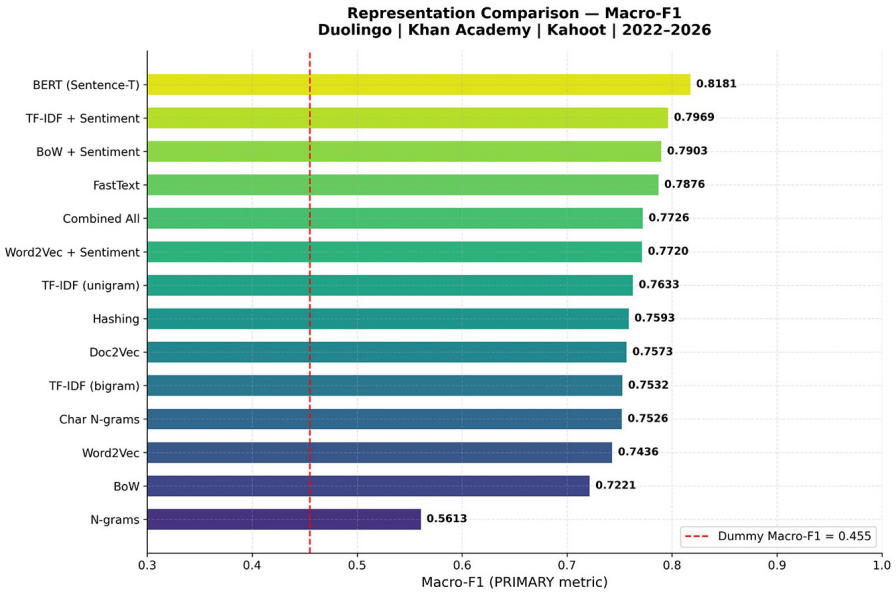


**Fig. 5.** Representation comparison by Macro-F1 dataset 2 (2022–2026)

*Notes:* BERT leads with Macro-F1 = 0.8181. N-grams collapse to 0.5613 due to vocabulary mismatch between gaming and educational bigram distributions. All remaining representations substantially exceed the dummy baseline (0.455).

**Table 6.** Classifier comparison results mobile games dataset (2026)

| Classifier | Accuracy | Precision | Recall | Macro-F1* | Minority-F1 | Majority-F1 | ROC-AUC |
|---|---|---|---|---|---|---|---|
| Random Forest | 0.9165 | 0.9488 | 0.9564 | **0.8006** | 0.6485 | 0.9526 | 0.8730 |
| Logistic Regression | 0.8841 | 0.9609 | 0.9048 | **0.7702** | 0.6084 | 0.9320 | 0.9014 |
| Gradient Boosting | 0.8812 | 0.9523 | 0.9102 | **0.7558** | 0.5809 | 0.9308 | 0.8830 |
| Decision Tree | 0.8935 | 0.9344 | 0.9450 | **0.7430** | 0.5464 | 0.9397 | 0.7217 |
| Naive Bayes | 0.8265 | 0.9194 | 0.8794 | **0.6428** | 0.3867 | 0.8989 | 0.6887 |

*Notes:* *Macro-F1 is the primary evaluation metric; classifiers are ranked by Macro-F1.

Figure 6 provides the confusion matrix for the best-performing classifier (RF), revealing that 131 of 208 minority-class reviews (63.0%) are correctly identified. Figure 7 plots the ROC curves for all classifiers, confirming that logistic regression achieves the highest area under the curve (AUC = 0.9014) despite ranking second on Macro-F1.

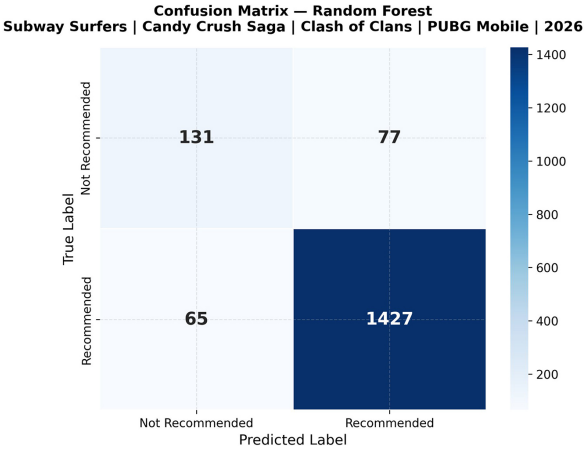


**Fig. 6.** Confusion matrix RF classifier, dataset 1 (2026)

*Notes:* Of 208 true "Not Recommended" reviews, 131 (63.0%) are correctly identified. The 77 false negatives represent undetected negative reviews, consistent with a Minority-F1 of 0.649 under realistic class imbalance.

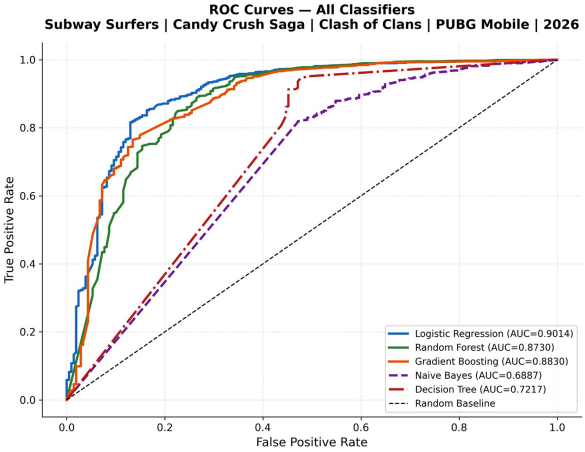


**Fig. 7.** ROC curves all classifiers, Dataset 1 (2026)

*Note:* Logistic regression achieves the highest AUC (0.9014). DT and Naive Bayes show substantially lower discriminative capacity (AUC = 0.7217 and 0.6887, respectively), reflecting the limitations of single-tree models and the independence assumption on hybrid representations.

Table 7 shows the classifier comparison (BERT as a fixed optimal representation for Dataset 2). Ordering of classifiers is quite different from the game's dataset. RF once again attained the peak Macro-F1 (0.8278) and Minority-F1 (0.7113) scores. On the other side, Logistic regression produced both the peak ROC-AUC one (.9357) and the most significant AUC achieved in both datasets and the peak Minority-F1 one (one 7151), which reflects that the spatially regular form of the BERT embeddings is well suited to the linear decision boundary. On the educational dataset Naïve Bayes performed significantly better (ROC-AUC = 0.9253; F1 = 0.9124) than on the game's dataset due to the greater lexical regularity of the educational reviews and their

longer average length, which yields data more amenable to the multinomial independence assumption. Among all classifiers, DT had the worst performance on both datasets (Macro-F1 = 0.6908; ROC-AUC = 0.7214). This reflects the incapacity of non-ensemble tree-based models to achieve reliable minority-class detection, regardless of domain. The educational dataset gives significantly higher "Not Recommended" recall over all classifiers: RF classifies 303 of 434 minority-class reviews correctly (recall = 0.698), while Logistic Regression gives the highest recall of the minority class on Dataset 2 (0.716). Negative complaints that render in education technology contexts particularly important; a missing complaint about the APP's content quality or the interface's usability extracts pedagogical value from learners. The higher recall for Dataset 2 relative to Dataset 1 (Logistic regression—0.716 vs. 0.608) suggests that NLP-based recommendation systems are likely more deployable on educational platforms that receive more detailed reviews with richer semantics. Naive Bayes demonstrates a greater cross-domain enhancement for minority-class performance (Minority-F1: 0.387→0.665) due to the increased lexical regularity of educational reviews that better accords with the multinomial independence assumption.

**Table 7.** Classifier comparison results educational APPs dataset (2022–2026)

| Classifier | Accuracy | Precision | Recall | Macro-F1* | Minority-F1 | Majority-F1 | ROC-AUC |
|---|---|---|---|---|---|---|---|
| RF | 0.9066 | 0.9409 | 0.9478 | **0.8278** | 0.7113 | 0.9443 | 0.9262 |
| LR | 0.8884 | 0.9681 | 0.8960 | **0.8229** | 0.7151 | 0.9306 | 0.9357 |
| GB | 0.8827 | 0.9583 | 0.8987 | **0.8100** | 0.6925 | 0.9276 | 0.9204 |
| NB | 0.8611 | 0.9646 | 0.8655 | **0.7889** | 0.6654 | 0.9124 | 0.9253 |
| DT | 0.8027 | 0.9163 | 0.8405 | **0.6908** | 0.5048 | 0.8768 | 0.7214 |

*Notes:* *Macro-F1 is the primary evaluation metric; classifiers are ranked by Macro-F1.

Figure 8 shows the SMOTE class resampling for Dataset 2. Figures 9 and 10 present the confusion matrix and ROC curves for Dataset 2, respectively. Every classifier achieves a higher ROC-AUC on the educational dataset than on the game's dataset: Random Forest: 0.9262 vs. 0.8730; Logistic Regression: 0.9357 vs. 0.9014; Gradient Boosting: 0.9204 vs. 0.8830 (see Figures 7 and 10), confirming the greater separability of educational review text.

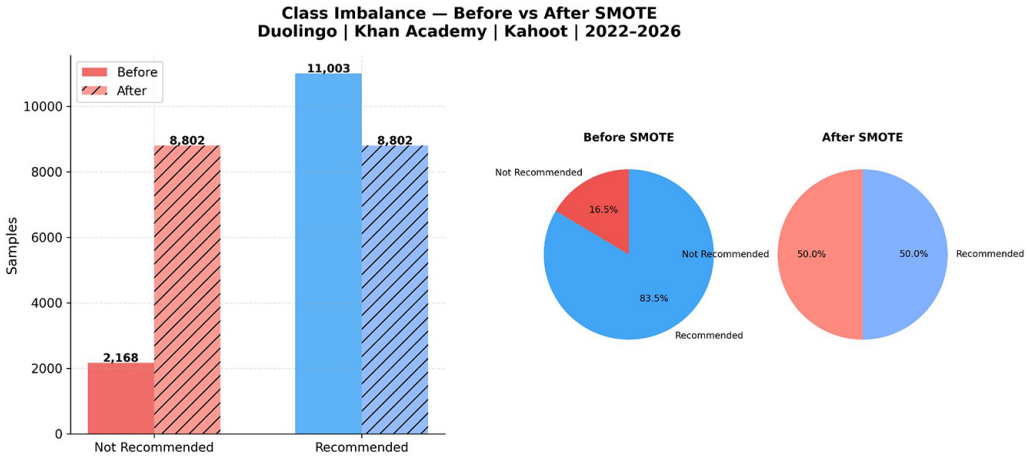


**Fig. 8.** Class imbalance before and after SMOTE dataset 2 (2022–2026)

*Note:* The minority class grows from 2,168 (16.5%) to 8,802 samples after SMOTE resampling, producing a balanced 50/50 training partition while preserving realistic imbalance in the test set.

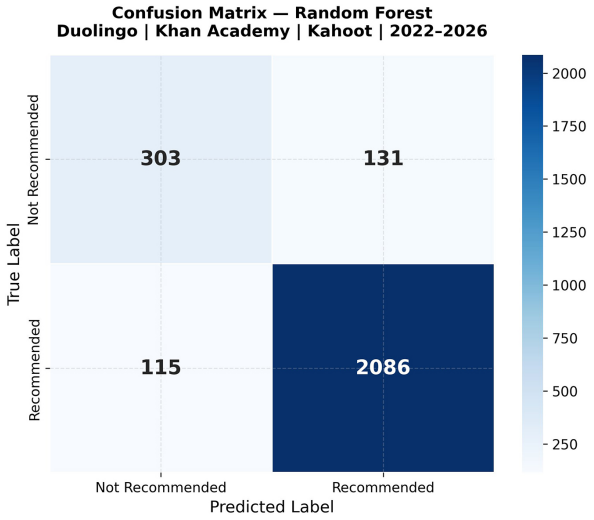


**Fig. 9.** Confusion matrix RF classifier, dataset 2 (2022–2026)

*Note:* Of 434 true "Not Recommended" reviews, 303 (69.8%) are correctly classified, yielding a Minority-F1 = 0.711 substantially higher than dataset 1 (0.649), reflecting the greater separability of educational review text.

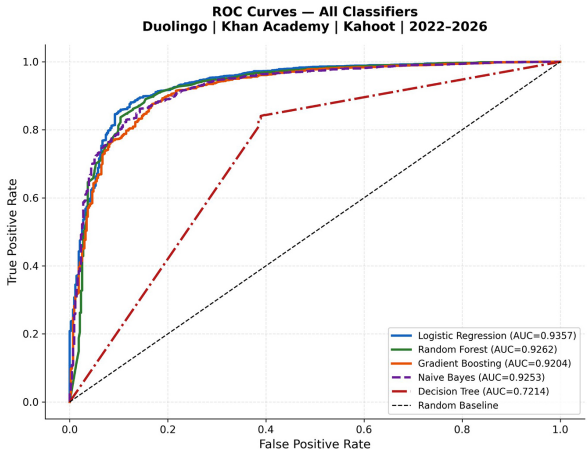


**Fig. 10.** ROC curves all classifiers, dataset 2 (2022–2026)

*Note:* Logistic regression achieves the highest AUC (0.9357), the best across both datasets. DT is the clear outlier (AUC = 0.7214). Naive Bayes, Gradient Boosting, and RF all cluster above AUC = 0.92, reflecting the geometrically regular structure of BERT embeddings.

Finding similar patterns from two datasets helps detect cross-domain findings. To begin with, the educational dataset produced a higher ROC-AUC than the games dataset for every classifier. Random Forest: 0.9262 vs. 0.8604; Logistic Regression: 0.9357 vs. 0.8589; Gradient Boosting: 0.9204 vs. 0.8927; see Figures 7 and 10. This suggests that the review text is more easily separable for recommendation prediction in educational APPs. This is likely due to the structured vocabulary and longer, more informative nature of educational reviews. Secondly, the optimal representation varies across domains: BoW + Sentiment performs best for games, while BERT excels for educational reviews. No representation may be universally superior, which requires influencing the selection with domain expertise. Third, it was discovered that applying SMOTE to training partitions consistently maintained high recall values (≥ 0.88 across all leading models in both datasets). This emphasizes the importance of class-imbalance correction for reliable minority-class detection.

The findings suggest that 15 representations with systematic evaluation of the classifier along with SMOTE balance are an effective and transferable baseline for mobile app recommendation prediction across the review domain.

## 4.3 Cost-sensitive learning versus SMOTE

To validate the choice of SMOTE as the primary imbalance-correction strategy, this section compares SMOTE-based resampling against Cost-Sensitive Learning (class_weight='balanced' applied to RF and Logistic Regression without synthetic oversampling). Cost-Sensitive Learning penalizes misclassification of the minority class through inverse-frequency class weighting rather than synthetic sample generation. As present in Table 8.

**Table 8.** SMOTE versus cost-sensitive learning: Macro-F1, Minority-F1, and minority-class recall comparison

| Method | Macro-F1* | Minority-F1 | Min-Recall | Majority-F1 | ROC-AUC |
|---|---|---|---|---|---|
| **Dataset 1: Mobile Games (TF-IDF + Sentiment, 2026)** | | | | | |
| SMOTE + Random Forest | **0.8006** | 0.6485 | | 0.9526 | 0.8730 |
| SMOTE + Logistic Regression | 0.7702 | 0.6083 | | 0.9320 | 0.9014 |
| Cost-Sensitive + Random Forest | 0.7511 | 0.5579 | 0.5096 | 0.9444 | 0.8554 |
| Cost-Sensitive + Logistic Reg. | 0.7678 | 0.6055 | 0.7452 | 0.9301 | 0.9047 |
| **Dataset 2: Educational Apps (BERT Sentence-T, 2022–2026)** | | | | | |
| SMOTE + Random Forest | **0.8278** | **0.7113** | | 0.9443 | 0.9262 |
| SMOTE + Logistic Regression | **0.8229** | **0.7151** | | 0.9306 | 0.9357 |
| Cost-Sensitive + Random Forest | 0.7138 | 0.4952 | 0.3548 | 0.9324 | 0.9210 |
| Cost-Sensitive + Logistic Reg. | **0.8166** | **0.7073** | 0.8687 | 0.9258 | 0.9392 |

*Notes:* *Macro-F1 is the primary ranking metric. Bold values = best per column per dataset.
Min-Recall = recall of "Not Recommended" class. CS = Cost-Sensitive Learning (class_weight='balanced', trained on original imbalanced data without SMOTE).

On Dataset 1, the Macro-F1 (0.8006) and Minority-F1 (0.6485) attained by RF-based SMOTE are significantly higher than those of the cost-sensitive RF (Macro-F1 = 0.7511, Minority-F1 = 0.5579), which are 0.049 Macro-F1 points lower. Cost-Sensitive Logistic Regression achieved the best "Not Recommended" recall of any configuration on Dataset 1 (0.745). The same configuration is SMOTE-based Logistic Regression, with significantly lower precision (0.51) and a marginal drop in Macro-F1 (0.7678 vs. 0.7702). The same trend manifests even more evidently on Dataset 2. The RF method based on SMOTE retains the highest scores on Macro-F1 (0.8278 vs. 0.7138) and Minority-F1 (0.7113 vs. 0.4952). Thus, Cost-Sensitive weighting alone is not found to be enough for high-dimensional BERT embeddings under serious class imbalance. Cost-Sensitive Logistic Regression achieves the highest minority-class recall across both Dataset 1 and Dataset 2, achieving 0.869, and the highest ROC-AUC on Dataset 2 with 0.9392. We note this as a practical deployment trade-off: where maximizing detection of every single negative review outweighs balanced performance, Cost-Sensitive Logistic Regression offers the highest sensitivity threshold. All in all, these results confirm that SMOTE provides better balanced classification performance (Macro-F1 and Minority-F1), thus representing the right

primary strategy of this framework. When maximum recall on negative reviews is the top operational priority (real-time quality monitoring systems), cost-sensitive logistic regression is a viable alternative where losing business due to false negatives can be a high cost.

The present study has several noted limitations. The datasets were obtained solely from the Google Play Store, and thus the results may not generalize to other distribution channels (e.g., the Apple App Store), which attracts a different user population and has a different convention for reviews. The binary labeling protocol that codes four- and five-star reviews as Recommended and one- and two-star reviews as Not Recommended necessarily loses information about nuance: a three-star review entails a conditional recommendation, but the protocol's exclusion of this information loses it. The collection of Dataset 1 took two months during the early part of 2026. Thus, there is a time bias that can occur if game updates, promotions, or the platform changes game sentiment. While SMOTE tackles class imbalance at the feature level, the algorithm suffers from a limitation because it relies on linear interpolation in the feature space. As a result, SMOTE cannot recreate the true distribution of minority-class reviews. This limitation becomes more profound in high-dimensional transformer embeddings. In sum, we take RF importance scores and SHAP values from the single best-performing configuration as the basis of our interpretation analysis, which may be unrepresentative of the full range of representation–classifier combinations examined in the present study.

## 5 CONCLUSION

This study proposes a systematic evaluation framework for performing mobile APP recommendation prediction using natural language processing and machine learning. An experimental protocol was developed to benchmark 15 representations and five classifiers under SMOTE-based class-imbalance correction on training partitions and SHAP-based interpretability analysis. The experiments were conducted on two Google Play Store datasets created independently on mobile game reviews and education app reviews. The outcomes indicate distinct patterns based on domain. According to the results found in the mobile game dataset, the sentiment-augmented hybrid representations performed the best. Specifically, the TF-IDF + Sentiment combination achieved the best Macro-F1 score (0.7987, Minority-F1 = 0.6452) under the corrected evaluation protocol, showing the discriminative power of domain-specific sentiment signals. On the educational app dataset, it led with a Macro-F1 of 0.8181 (Minority-F1 = 0.6964) and ROC-AUC of 0.9190. Educational reviews are semantically more complex and longer than retail app reviews. The classifier with the highest F1-score in both domains was Random Forest. GB and Logistic Regression showed the highest ROC-AUC for the games and educational datasets, respectively. DT was the weakest classifier in both cases. Across the leading models in both datasets, SMOTE maintained the Minority F1 above 0.60, thus confirming the effectiveness of the approach for minority-class detection. In summary, the results show that no representation or classifier is best across mobile APP domains. Moreover, reliable recommendation prediction should be rigorously evaluated in an adapted domain. Future work on the framework will be extended to multilingual review corpora. Moreover, aspect-level sentiment features will also be used. Additionally, fine-tuned domain-specific transformer architectures will be explored. In the end, minority-class detection will improve across a variety of APP categories.

## 5.1 Acknowledgment

The authors thank the School of Computer Sciences, Universiti Sains Malaysia (USM), and the Faculty of Information Technology, Applied Science Private University (ASU), for their support in conducting this study.

## 6 REFERENCES

[1] Newzoo, "Global Games Market Report 2023," Newzoo BV, Amsterdam, Netherlands, 2023. [Online]. Available: https://newzoo.com/resources/rankings/top-10-countries-by-game-revenues

[2] Y. Yu, D.-T. Dinh, B.-H. Nguyen, F. Yu, and V.-N. Huynh, "Mining insights from esports game reviews with an aspect-based sentiment analysis framework," *IEEE Access*, vol. 11, pp. 61161–61172, 2023. https://doi.org/10.1109/ACCESS.2023.3285864

[3] P. Jaiswal, H. Setia, P. Raghuwanshi, and P. Randhawa, "A natural language processing model for predicting five-star ratings of video games on short-text reviews," *Engineering Proceedings*, vol. 59, no. 1, p. 58, 2023. https://doi.org/10.3390/engproc2023059058

[4] Y. Zhang, X. Li, Z. Wang, and J. Chen, "Review classification based on machine learning: Classifying game user reviews," *IEEE Access*, vol. 12, pp. 12801–12818, 2024. https://doi.org/10.1109/ACCESS.2023.3342294

[5] E. Hasan, M. Rahman, C. Ding, J. X. Huang, and S. Raza, "Review-based recommender systems: A survey of approaches, challenges and future perspectives," *ACM Computing Surveys*, vol. 58, no. 1, Art. no. 25, 2025. https://doi.org/10.1145/3742421

[6] M. Wankhade, A. C. S. Rao, and C. Kulkarni, "A survey on sentiment analysis methods, applications, and challenges," *Artificial Intelligence Review*, vol. 55, pp. 5731–5780, 2022. https://doi.org/10.1007/s10462-022-10144-1

[7] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. 2019 Conf. North American Chapter of the Association for Computational Linguistics (NAACL-HLT)*, Minneapolis, MN, USA, 2019, pp. 4171–4186. https://doi.org/10.18653/v1/N19-1423

[8] N. Alturayeif, H. Aljamaan, and J. Hassine, "An automated approach to aspect-based sentiment analysis of apps reviews using machine and deep learning," *Automated Software Engineering*, vol. 30, p. 30, 2023. https://doi.org/10.1007/s10515-023-00397-7

[9] J. Hamari, K. Alha, S. Järvelä, J. M. Kivikangas, J. Koivisto, and J. Paavilainen, "Why do players buy in-game content? An empirical study on concrete purchase motivations," *Computers in Human Behavior*, vol. 68, pp. 538–546, 2017. https://doi.org/10.1016/j.chb.2016.11.045

[10] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using Siamese BERT-networks," in *Proc. 2019 Conf. Empirical Methods in Natural Language Processing (EMNLP-IJCNLP)*, Hong Kong, China, 2019, pp. 3982–3992. https://doi.org/10.18653/v1/D19-1410

[11] T. Hasib, N. A. Towhid, K. O. Faruk, J. Al Mahmud, and M. Mridha, "Data oversampling and imbalanced datasets: An investigation of performance for machine learning and feature engineering," *Journal of Big Data*, vol. 11, no. 1, pp. 1–35, 2024. https://doi.org/10.1186/s40537-024-00943-4

[12] A. Al Tawil, "Game reviews dataset," *Kaggle*, 2026. [Online]. Available: https://www.kaggle.com/datasets/araraltawil/game-reviews [Accessed: Feb. 2026].

[13] D. Dablain, B. Krawczyk, and N. V. Chawla, "DeepSMOTE: Fusing deep learning and SMOTE for imbalanced data," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 9, pp. 6390–6404, 2023. https://doi.org/10.1109/TNNLS.2021.3136503

[14] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 30, 2017, pp. 4765–4774. [Online]. Available: https://proceedings.neurips.cc/paper/2017/hash/8a20a8621978632d76c43dfd28b67767-Abstract.html

[15] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001. https://doi.org/10.1023/A:1010933404324

[16] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *Annals of Statistics*, vol. 29, no. 5, pp. 1189–1232, 2001. https://doi.org/10.1214/aos/1013203451

[17] J. L. Leevy, T. M. Khoshgoftaar, and A. Abdallah, "An investigation of SMOTE based methods for imbalanced datasets with data complexity analysis," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 12, pp. 12660–12672, 2023. https://doi.org/10.1109/TKDE.2022.3179381

[18] H. Khalid, E. Shihab, M. Nagappan, and A. E. Hassan, "What do mobile app users complain about?" *IEEE Software*, vol. 32, no. 3, pp. 70–77, 2015. https://doi.org/10.1109/MS.2014.50

[19] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011. [Online]. Available: https://jmlr.org/papers/v12/pedregosa11a.html

[20] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," *Information Processing & Management*, vol. 24, no. 5, pp. 513–523, 1988. https://doi.org/10.1016/0306-4573(88)90021-0

[21] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 26, 2013, pp. 3111–3119. [Online]. Available: https://arxiv.org/abs/1310.4546

[22] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, "Bag of tricks for efficient text classification," in *Proc. 15th Conf. European Chapter of the Association for Computational Linguistics (EACL)*, Valencia, Spain, 2017, pp. 427–431. https://doi.org/10.18653/v1/E17-2068

[23] J. Pennington, R. Socher, and C. D. Manning, "GloVe: Global vectors for word representation," in *Proc. 2014 Conf. Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar, 2014, pp. 1532–1543. https://doi.org/10.3115/v1/D14-1162

[24] Q. V. Le and T. Mikolov, "Distributed representations of sentences and documents," in *Proc. 31st Int. Conf. Machine Learning (ICML)*, Beijing, China, 2014, pp. 1188–1196. [Online]. Available: https://arxiv.org/abs/1405.4053

[25] M. Aria, C. Cuccurullo, and A. Gnasso, "A comparison among interpretative proposals for Random Forests," *Machine Learning with Applications*, vol. 6, p. 100094, 2021. https://doi.org/10.1016/j.mlwa.2021.100094

[26] A. Al Tawil, "Educational App reviews dataset," *Kaggle*, 2026. [Online]. Available: https://www.kaggle.com/datasets/araraltawil/educationreview-games [Accessed: Feb. 2026].

[27] A. D. Samala, S. Papadakis, and S. Rawas, "Global insights into mobile learning in higher education: A PRISMA-guided bibliometric analysis from 2007 to 2023," *International Journal of Educational Reform*, 2025. https://doi.org/10.1177/10567879251341869

[28] S. Papadakis and T. Karakose, "Gamification and student achievement: Potential benefits, limitations, and effective use in educational environments," *Educational Process: International Journal*, 2025. https://doi.org/10.22521/edupij.2025.19.529

[29] F. C. Blumberg *et al.*, "Current state of play: Children's learning in the context of digital games," *Journal of Children and Media*, vol. 18, no. 2, pp. 293–299, 2024. https://doi.org/10.1080/17482798.2024.2335725

[30] A. K. Tegegnie, "Multi-task mining of Ethiopian mobile app reviews using machine learning and deep learning approaches," *Int. J. Interact. Mobile Technol.*, vol. 20, no. 1, pp. 137–159, 2026. https://doi.org/10.3991/ijim.v20i01.58867

[31] T. Ananthi Claral Mary, "Hybrid deep learning model to predict students' sentiments in higher educational institutions," *Int. J. Interact. Mobile Technol.*, vol. 19, no. 1, pp. 46–61, 2025. https://doi.org/10.3991/ijim.v19i01.50883

[32] Z. Bousalem, A. Qazdar, I. El Guabassi, and A. Haj, "A recommendation system based on early academic performance prediction and student classification: Utilizing artificial intelligence and mobile-based application," *Int. J. Interact. Mobile Technol.*, vol. 18, no. 15, pp. 169–189, 2024. https://doi.org/10.3991/ijim.v18i15.47135

## 7 AUTHORS

**Arar Al Tawil** received the B.Sc. in Computer Science from Al-Hussein Bin Talal University, Jordan, in 2018, and the M.Sc. from the University of Jordan in 2021. He is currently pursuing a Ph.D. at the School of Computer Sciences, Universiti Sains Malaysia (USM), and serves as a Lecturer and Developer at the Computer Sciences Department, Faculty of Information Technology, Applied Science Private University (ASU), Amman, Jordan. His research interests include virtual reality, reinforcement learning, optimization algorithms, machine learning, deep learning, and NLP. He has authored several papers in IEEE, Springer, and IGI Global venues, with projects spanning VR-based rehabilitation and adaptive gamified learning systems.

**Siti Hazyanti Mohd Hashim** obtained her B.Sc., M.Sc., and Ph.D. in Computer Science from Universiti Teknologi MARA (UiTM), Malaysia. She is currently a Lecturer at the School of Computer Sciences, Universiti Sains Malaysia (USM), Pulau Pinang, Malaysia. Her research interests include computer technology, multimedia, game and mobile applications, virtual reality, and software engineering. She actively contributes to academic supervision and collaborative research in immersive technologies and educational innovation (E-mail: sitihazyanti@usm.my).