Indonesian Journal of Electrical Engineering and Computer Science
Vol. 43, No. 1, July 2026, pp. 259~270
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v43.i1.pp259-270

Evaluating oversampling methods for imbalanced Arabic dialect identification

Maulana Ihsan Ahmad, Aina Musdholifah, Arif Nurwidyantoro
Department of Computer Science and Electronics, Faculty of Mathematics and Natural Sciences,
Universitas Gadjah Mada, Yogyakarta, Indonesia

Article Info

Article history:

Received: Feb 4, 2026
Revised: Jun 2, 2026
Accepted: Jun 27, 2026

Keywords:

Arabic dialect identification
Class imbalance
Feature engineering
Oversampling methods
SMOTE
Text classification

ABSTRACT

This study investigates whether oversampling is a reliable solution for severe class imbalance in Arabic dialect identification. Using the Shami Corpus as a controlled testbed, we demonstrate that conventional oversampling often fails in high-dimensional sparse text spaces, but density-based cluster filtering can effectively resolve this. We conduct a comparative evaluation of SMOTE, clustering-guided variants (ASTRA-SMOTE and SMOTE-RADIANT), and a cost-sensitive ClassWeight approach under an identical 5,644-dimensional feature-engineering pipeline using LightGBM and XGBoost. On the held-out test set, standard SMOTE and class weighting frequently distorted decision boundaries, yielding inconsistent gains across models. In contrast, SMOTE-RADIANT yields a statistically significant macro-F1 improvement for LightGBM (0.8539 vs. 0.8526 on the original data) with a large effect size (r = 0.511), successfully rescuing minority dialects without degrading the majority class. These findings suggest that while oversampling is not universally reliable in sparse text spaces, coupling it with density-based noise neutralization (RADIANT) provides a robust and interpretable alternative to deep learning models. This study provides methodological clarity and reproducible guidance for fair and inclusive Arabic NLP systems.

Corresponding Author:

Arif Nurwidyantoro
Department of Computer Science and Electronics, Faculty of Mathematics and Natural Sciences
Universitas Gadjah Mada,
Yogyakarta, Indonesia
Email: arifn@ugm.ac.id

1. INTRODUCTION

Arabic is spoken by over 400 million people across the Middle East and North Africa, and its regional dialects exhibit substantial variation in vocabulary, morphology, and syntax [1]. Accurate dialect identification is therefore essential for downstream natural language processing (NLP) applications such as speech recognition, machine translation, and sentiment analysis [2]. Several benchmark corpora have been developed for this task, including nuanced Arabic dialect identification (NADI) [3], Multi-Arabic dialect applications and resources (MADAR) [4], and dialectal Arabic resource of tweets (DART) [5]. Among them, the Shami Corpus [6] provides a focused testbed on four Levantine dialects-Jordanian, Lebanese, Palestinian, and Syrian-making it particularly suitable for analyzing class imbalance effects.

Dialect identification is challenging because dialects differ markedly from Modern Standard Arabic (MSA) and are primarily observed in informal social media text [7]-[10]. Twitter data further introduce noise due to short message length, orthographic variation, emojis, hashtags, and code-switching [11]. A major difficulty of the Shami Corpus is its skewed class distribution: Syrian tweets account for

approximately 57% of the data, whereas Jordanian represents only 10.6% [6]. Such imbalance biases classifiers toward the majority class and degrades recognition of minority dialects [12]. However, it remains underexplored how commonly used imbalance-handling strategies-ranging from generative oversampling to cost-sensitive weighting-perform in highly sparse, high-dimensional text feature spaces.

Feature-based representations continue to play a central role in Arabic dialect identification due to their interpretability and computational efficiency. Lexical features effectively capture discriminative vocabulary patterns [12]-[14], while lexicon-based and centroid-similarity features provide complementary dialectal cues [3], [15]-[19]. Although recent studies increasingly rely on transformer-based models such as AraBERT, ARBERT, and MARBERT [20]-[23], these models are computationally expensive and rarely facilitate systematic analysis of imbalance-handling mechanisms. Lightweight feature-engineered pipelines therefore remain valuable for transparent investigation of methodological effects under fixed representations and evaluation protocols [24]-[27].

To mitigate class imbalance, oversampling techniques such as the synthetic minority over-sampling Technique (SMOTE) are widely adopted [28], [29]. However, in high-dimensional spaces (e.g., thousands of extracted lexical and morphological features), standard SMOTE may introduce noisy or overlapping instances that distort class boundaries. Clustering-guided variants, such as KMeans and DBSCAN-based filters, attempt to restrict synthetic generation to minority-dense regions or filter outliers [25], [30]-[34]. Despite their popularity, reported findings on their effectiveness in Arabic dialect identification remain inconsistent, largely due to variations in evaluation protocols and the lack of comparison against non-generative cost-sensitive learning.

To address this gap, the present study provides the first controlled evaluation of oversampling methods for Arabic dialect identification that systematically compares SMOTE and clustering-guided variants (ASTRA-SMOTE and SMOTE-RADIANT) alongside a cost-sensitive ClassWeight comparator under an identical feature-engineering pipeline. By fixing the corpus, preprocessing steps, feature representation, and evaluation protocol, this work effectively disentangles the impact of imbalance handling from learner-specific inductive biases in sparse text spaces.

Specifically, this study contributes by: (1) conducting a unified comparative evaluation in a highly sparse 5,644-dimensional feature space using LightGBM and XGBoost; (2) applying rigorous bootstrapping statistical validation to measure effect sizes on a held-out test set; (3) aggregating feature importance at the block level to interpret the models' decision-making processes; and (4) demonstrating that density-based filtering (RADIANT) significantly outperforms conventional SMOTE and class weighting. Rather than merely proposing a new algorithm, this work provides methodological clarity by identifying when oversampling yields stable minority improvements and when it degrades performance due to learner-feature interactions.

The remainder of this paper is organized as follows. Section 2 describes the dataset, feature representations, rebalancing strategies, learning models, and evaluation protocol. Section 3 presents the experimental results, statistical analysis, and interpretative discussion of findings. Section 4 concludes with key findings, limitations, and directions for future research.

2. METHOD

2.1. Problem setup and data

We investigate multi-class Arabic dialect identification under severe imbalance using the Shami Corpus [6], comprising 66,247 tweets across four Levantine dialects Table 1. Used as released to preserve realistic noise and biases [6], the corpus is heavily skewed: Syrian tweets dominate (57.0%), while Jordanian accounts for only 10.6%. To prevent data leakage, we split the data into a held-out test set (25%) and a training set (75%), subdividing the latter (80/20) for model fitting and validation. Our five-stage pipeline Figure 1 encompasses text preprocessing, high-dimensional feature engineering, imbalance handling (original, data-level oversampling, and algorithm-level weighting), gradient boosting training, and bootstrap statistical evaluation.

2.2. Preprocessing

To reduce orthographic variation, we apply light normalization (removal of diacritics and tatweel) and aggressive normalization (e.g., ه → ة, ي → ى, ا → إ/أ/آ). Non-linguistic artifacts such as URLs, mentions, hashtags, digits, and symbols are removed using regular expressions. Stopword handling is applied selectively using a comprehensive Arabic stopword list combined with topic-specific stopwords (e.g., country and city names like "عمان", "لبنان", "سوريا") to prevent the model from exploiting geographical biases.

2.3. Feature engineering

We engineered a comprehensive, multi-block feature representation comprising 5,644 dimensions. This extreme dimensionality reflects the inherent sparsity of text data, intentionally testing the limits of

standard oversampling algorithms and justifying the need for cluster-based filtering. To manage dimensionality and noise, adaptive feature selection was applied. Lexical word features (TF-IDF and BM25) were filtered using the χ^2 (Chi-square) test, while character-level n-grams were strictly filtered using mutual information (MI). Additionally, out-of-fold (OOF) stacking probabilities from Logistic Regression and LinearSVC were generated using a 5-fold stratified cross-validation. To ensure stability and prevent data leakage, these OOF predictions were averaged across two different random seeds internally within the training set. The complete functional blocks are summarized in Table 2.

Table 1. Dataset distribution in the shami dialect corpus

Dialect	Jordan	Lebanon	Palestine	Syria	Total
Tweets	7,017	10,829	10,642	37,759	66,247
Percentage (%)	10.6	16.3	16.1	57.0	100

Figure 1. System overview of the proposed pipeline

Table 2. Feature blocks summary

Block	Dimension	Description
Lexical (TF-IDF & BM25)	2420	Word and char_wb n-grams, aggressively filtered using χ^2 selection to retain the most discriminative lexical patterns.
Character N-grams (MI)	3,150	Character n-grams (sizes 2-3 and 4-5) strictly filtered using Mutual Information (MI) to capture sub-word dialect markers.
Structural & Stats	9	Document length, token counts, type-token ratio, and stopword statistics.
Morphological Signals	34	Shami-specific prefixes/suffixes (e.g., ما, عم, ب), explicit rules, and morphological word interactions.
Distributional Priors	5	Collocation dominance scores and NB-SVM log-count ratios.
OOF Stacking & Interactions	26	Out-of-fold (OOF) prediction probabilities from Logistic Regression and LinearSVC, combined with their interaction terms (product, diff, max).

2.4. Imbalance handling strategies

Five imbalance-handling scenarios are evaluated Table 3. Resampling is explicitly executed exclusively on the sub-training data prior to classifier training, keeping the validation and test sets completely representative of the real-world distribution.

To prevent excessive noise generation, the sampling_strategy for all generative methods was strictly tuned via grid search, capping synthetic targets to approximately 7,899 samples for Jordanian, and 19,400 for Lebanese and Palestinian. ASTRA-SMOTE introduces a pre-filtering stage, restricting synthesis exclusively to minority clusters exceeding a 50% purity threshold. Conversely, SMOTE-RADIANT acts as a post-filter; it allows standard SMOTE generation but subsequently applies DBSCAN exclusively to the synthetic data (clean_only_synth = True) to neutralize overlapping noise at inter-class decision boundaries. The ClassWeight scenario is included as a non-generative comparator to assess whether performance shifts arise from data augmentation or merely loss reweighting.

Figure 2 illustrates the workflows of the evaluated oversampling methods Figure 2(a) SMOTE, Figure 2(b) ASTRA-SMOTE, and Figure 2(c) SMOTE-RADIANT. While all methods generate a balanced training dataset, ASTRA-SMOTE introduces clustering before oversampling, and SMOTE-RADIANT additionally removes noisy synthetic samples.

Table 3. Imbalance handling strategies evaluated

Scenario	Description	Key parameters
Original	Uses original imbalanced data without resampling.	-
SMOTE	Interpolates minority samples conventionally.	k-neighbors = 5
ASTRA-SMOTE	Applies MiniBatchKMeans clustering; generates synthetic data strictly inside "safe" minority-dense clusters.	n_clusters = 128, safe_threshold = 0.5, alpha_mode = full, k-neighbors = 5
SMOTE-RADIANT	Performs SMOTE followed by density-based filtering (DBSCAN) to remove overlapping synthetic noise at class boundaries.	Metric = cosine, eps = auto, min_samples = auto, clean_only_synth = True
ClassWeight	Cost-sensitive learning using standard inverse-frequency class weights.	class_weight="balanced"

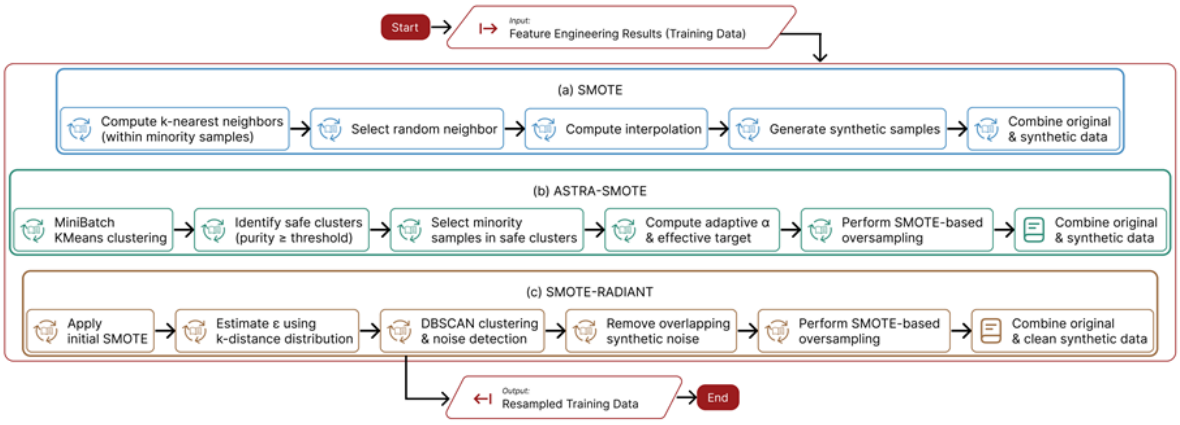

Figure 2. Oversampling strategies (a) SMOTE, (b) ASTRA-SMOTE, and (c) SMOTE-RADIANT

2.5. Models, training, and evaluation protocol

We employ two gradient boosting classifiers: LightGBM (leaf-wise growth) and XGBoost (level-wise growth). Optimal oversampling parameters were selected by maximizing the harmonic mean of both models' validation macro-F1 scores. Classifier hyperparameters were subsequently tuned: LightGBM (num_leaves=25, colsample_bytree=0.4, min_child_samples=50, L1/L2 regularization=1.0) and XGBoost (max_depth=8, colsample_bytree=0.3, min_child_weight=30, L1/L2 regularization=1.5). Both models utilized a 0.01 learning rate and 150 early stopping rounds.

To ensure statistical rigor on the held-out test set (16,562 tweets), we performed 1,000 bootstrap iterations (sampling with replacement) on the predictions. Paired two-sided Wilcoxon signed-rank tests and Cohen's effect sizes (r) were applied to the resulting Macro-F1 distributions. This protocol strictly evaluates whether performance shifts are statistically significant and practically meaningful, ensuring out-of-sample validity and preventing P-hacking.

3. RESULTS AND DISCUSSION

3.1. Performance analysis

Table 4 reports the held-out test-set performance in terms of macro-precision, macro-recall, macro-F1, and accuracy for the evaluated strategies. The models trained on the original data already achieve very high performance due to the robust 5,644-dimensional feature engineering pipeline. LightGBM and XGBoost initially achieved macro-F1 scores of 0.8526 and 0.8517, respectively (bootstrap means: 0.8525 and 0.8516).

Table 4. Test-set performance of LightGBM and XGBoost under five scenarios

Model	Rebalancing strategy	Precision	Recall	Macro-F1	Accuracy
LightGBM	Original	0.8627	0.8439	0.8526	0.8963
LightGBM	SMOTE	0.8584	0.8463	0.8518	0.8961
LightGBM	ASTRA-SMOTE	0.8603	0.8443	0.8517	0.8959
LightGBM	SMOTE-RADIANT	0.8620	0.8469	0.8539	0.8974
LightGBM	ClassWeight	0.8431	0.8578	0.8502	0.8931
XGBoost	Original	0.8620	0.8429	0.8517	0.8960
XGBoost	SMOTE	0.8602	0.8454	0.8524	0.8964
XGBoost	ASTRA-SMOTE	0.8604	0.8430	0.8511	0.8956
XGBoost	SMOTE-RADIANT	0.8618	0.8445	0.8525	0.8964
XGBoost	ClassWeight	0.8427	0.8587	0.8504	0.8934

In this high-dimensional space, the addition of standard SMOTE or clustering-restricted ASTRA-SMOTE generally degraded the macro-F1 performance for LightGBM. Generating synthetic text instances in such a sparse space often introduces overlapping noise that distorts precise decision boundaries. However, SMOTE-RADIANT successfully broke this bottleneck, achieving the highest numerical macro-F1 (0.8539) and accuracy (89.74%) across all LightGBM configurations. By treating synthetic data generation as a two-step process-where DBSCAN acts as a post-filter to neutralize overlapping anomalies-RADIANT allows the model to safely exploit augmented minority regions without boundary degradation.

For XGBoost, the impact of oversampling was more nuanced. While SMOTE and SMOTE-RADIANT both yielded identical marginal gains in macro-F1 (0.8525), the overall performance of XGBoost remained consistently lower than that of LightGBM. This suggests that the level-wise tree growth strategy of XGBoost is generally more sensitive and less optimal at handling synthetic perturbations in highly sparse lexical spaces compared to LightGBM's leaf-wise growth.

Interestingly, the cost-sensitive approach (ClassWeight) exhibited the lowest macro-F1 and accuracy across both learners. While it successfully forced the models to pay attention to minority classes (as evidenced by the highest recall scores), it severely compromised macro-precision. This highlights a critical limitation of non-generative reweighting in highly overlapping lexical spaces: it simply shifts the decision threshold, increasing minority recognition at the direct expense of the majority class.

3.2. Error patterns and per-class performance

To evaluate dialect-level performance shifts, we analyze the confusion matrices Figure 3 alongside the per-class F1-scores Table 5 and Figure 4. In the LightGBM (original) configuration, the model performs robustly but naturally favors the majority Syrian class, yielding lower true positives for the severely underrepresented Jordanian dialect (1,319 out of 1,754). A critical finding emerges when applying the cost-sensitive ClassWeight strategy. While it successfully increases minority recall-Jordanian true positives rise from 1,319 to 1,391-it indiscriminately shifts the global decision boundary. This severely damages the majority class, dropping correct Syrian predictions from 9,059 (original) to 8,830. Consequently, this influx of false positives causes a precision collapse, drastically dropping the Jordanian F1-score from 0.7941 to 0.7850 in LightGBM. This demonstrates that non-generative cost-sensitive learning merely cannibalizes the majority class rather than improving true separability.

In contrast, generative approaches show distinct model-dependent behaviors. For LightGBM, unconstrained standard SMOTE and cluster-restricted ASTRA-SMOTE introduce overlapping synthetic anomalies that slightly degrade the overall per-class balance. However, SMOTE-RADIANT provides a much safer and more effective minority-majority trade-off. By utilizing DBSCAN to filter out overlapping noise, RADIANT provides localized synthetic support. This allows LightGBM to establish tighter decision boundaries, improving Jordanian and Lebanese recognition without aggressively cannibalizing the Syrian majority class.

Figure 3(a) shows the original LightGBM classification pattern. Figure 3(b) highlights the balanced improvement achieved by SMOTE-RADIANT. Figure 3(c) illustrates the trade-off introduced by ClassWeight. Figure 3(d) shows the original XGBoost classification pattern. Figure 3(e) demonstrates improved class balance with SMOTE-RADIANT. Figure 3(f) illustrates the stronger bias introduced by ClassWeight.

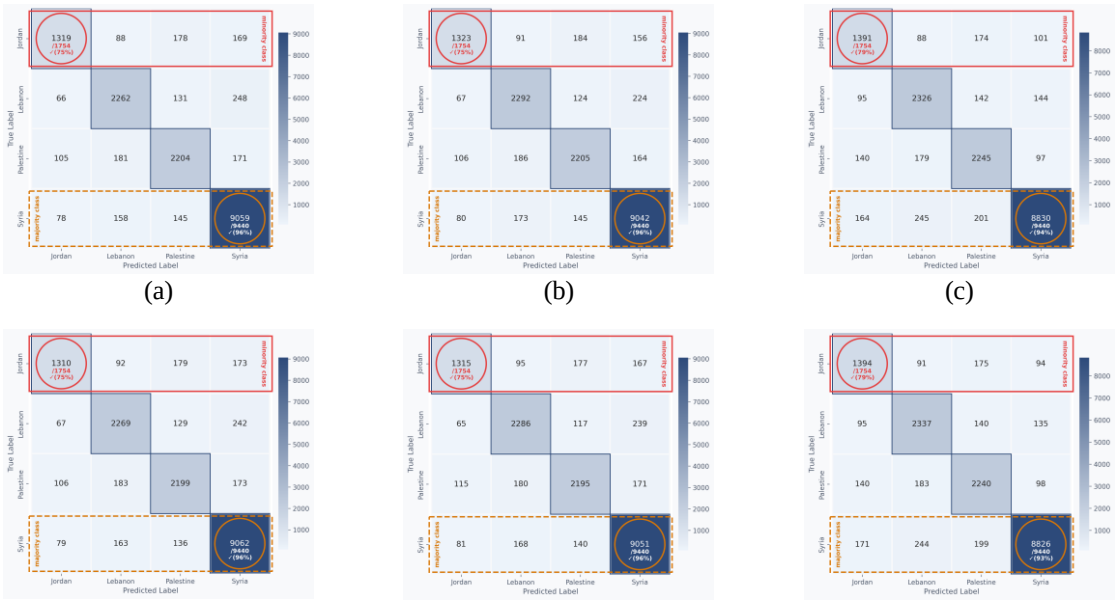

Figure 3. Confusion matrices comparing dialect classification performance under different balancing strategies (a) LightGBM (Original), (b) LightGBM with SMOTE-RADIANT, (c) LightGBM with classweight, (d) XGBoost (Original), (e) XGBoost with SMOTE-RADIANT, and (f) XGBoost with classweight.

For XGBoost, all rebalancing strategies-whether generative or cost-sensitive-struggle to consistently improve the severely imbalanced Jordanian class. Even with RADIANT, XGBoost's level-wise growth shows higher sensitivity to synthetic perturbations, failing to fully disentangle the boundaries in the 5,644-dimensional space.

Across all models and strategies, confusion between Lebanese and Palestinian remains the most frequent bidirectional error. This reflects the strong lexical, morphological, and geographical overlap between these two Levantine dialects, coupled with the limited contextual cues available in short Twitter texts. Overall, this combined analysis confirms that while ClassWeight redistributes misclassification patterns by sacrificing majority-class precision, density-guided oversampling (SMOTE-RADIANT) genuinely refines inter-class decision boundaries for leaf-wise gradient boosting models.

Table 5. Per-class F1-score on the test set (Δ relative to original)

Model	Dialect	Original	SMOTE (Δ)	ASTRA-SMOTE (Δ)	SMOTE-RADIANT (Δ)	ClassWeight (Δ)
LightGBM	Jordan	0.7941	0.7907 (−0.0034)	0.7913 (−0.0028)	0.7946 (+0.0005)	0.7850 (−0.0091)
	Lebanon	0.8384	0.8422 (+0.0038)	0.8391 (+0.0007)	0.8459 (+0.0075)	0.8390 (+0.0006)
	Palestine	0.8287	0.8289 (+0.0002)	0.8265 (−0.0022)	0.8264 (−0.0023)	0.8280 (−0.0007)
	Syria	0.9492	0.9453 (−0.0039)	0.9498 (+0.0006)	0.9485 (−0.0007)	0.9489 (−0.0003)
	Macro-F1	0.8526	0.8518 (−0.0008)	0.8517 (−0.0009)	0.8539 (+0.0013)	0.8502 (−0.0024)
XGBoost	Jordan	0.7907	0.7931 (+0.0024)	0.7895 (−0.0012)	0.7922 (+0.0015)	0.7845 (−0.0062)
	Lebanon	0.8394	0.8408 (+0.0014)	0.8395 (+0.0001)	0.8425 (+0.0031)	0.8403 (+0.0009)
	Palestine	0.8275	0.8276 (+0.0001)	0.8258 (−0.0017)	0.8264 (−0.0011)	0.8273 (−0.0002)
	Syria	0.9492	0.9476 (−0.0016)	0.9495 (+0.0003)	0.9488 (−0.0004)	0.9494 (+0.0002)
	Macro-F1	0.8517	0.8524 (+0.0007)	0.8511 (−0.0006)	0.8525 (+0.0008)	0.8504 (−0.0013)

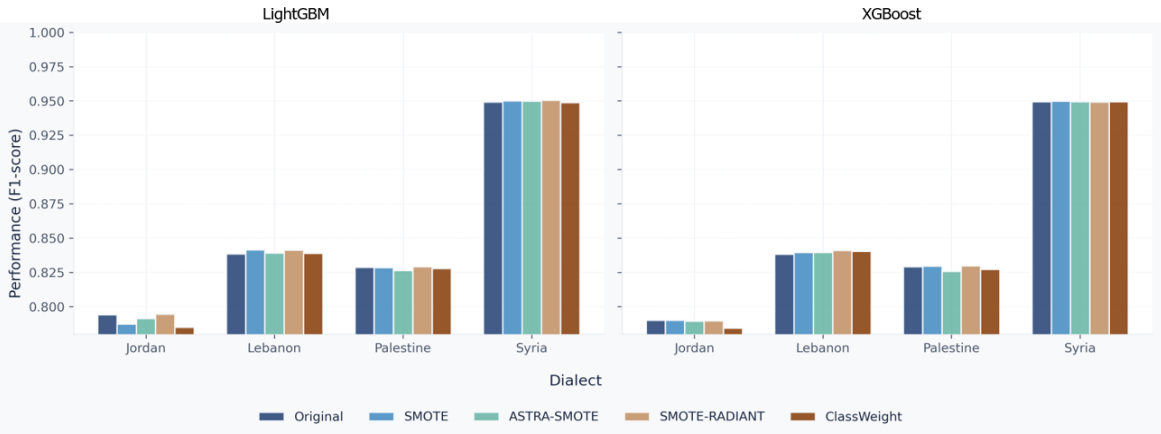

Figure 4. Per-class F1-score of LightGBM and XGBoost under different imbalance-handling strategies. LightGBM exhibits balanced improvements across minority dialects under SMOTE-RADIANT, successfully avoiding the precision collapse seen in ClassWeight and XGBoost struggles with boundary distortion across all generative and cost-sensitive methods

3.3. Block-level feature importance

Given the extreme sparsity of the 5,644-dimensional space, analyzing individual features is uninterpretable. Instead, we aggregated the feature_importances_ (measured by the total number of splits) from the tree models into functional blocks. Figure 5 illustrates these aggregated patterns across representative configurations.

Figure 5(a) shows the original feature hierarchy of LightGBM. Figure 5(b) demonstrates that SMOTE-RADIANT preserves this hierarchy with only minor changes. Figure 5(c) reveals that ClassWeight markedly increases split counts across feature blocks. Figure 5(d) presents the original XGBoost feature distribution, Figure 5(e) shows that SMOTE-RADIANT maintains a similar pattern, and Figure 5(f) indicates that ClassWeight again substantially inflates split counts.

A distinct hierarchy emerges between feature volume and feature efficiency. Across all Original models, the character-level sub-word markers (char_mi23) and sparse word-level representations (lexical) dominate the total importance due to their massive dimensionality (500 and 2,420 columns, respectively). Note that char_mi23 refers specifically to the character 2–3-gram sub-block within the broader 3,150-

dimensional Character N-grams (MI) block reported in Table 2. However, in terms of average importance per column, the Naive Bayes-SVM log-count ratios (nbsvm) and the Out-of-Fold stacking probabilities (oof_svm, oof_lr, oof_interactions) are exceptionally powerful. For instance, in LightGBM (Original), the 4 columns of the nbsvm block average 3,710 splits per column, making them the most structurally critical signals for the first few levels of the decision trees.

When SMOTE-RADIANT is applied, the relative contribution and ranking structure of these feature blocks remain highly stable. In LightGBM RADIANT, char_mi23 (35,470 splits) and lexical (28,440 splits) retain their positions, with a slight reinforcement in longer character sequences (char_mi45 rising to 18,333 splits). This stability is a strong indicator that density-based filtering successfully augments the minority regions without disrupting or confusing the algorithm's reliance on its most trusted linguistic cues.

In sharp contrast, applying cost-sensitive learning (ClassWeight) drastically alters tree growth behavior. In both LightGBM and XGBoost, the total number of splits across almost all blocks nearly doubles. For example, XGBoost's reliance on char_mi23 inflates from 57,604 splits (Original) to 129,946 splits (ClassWeight). This extreme inflation reveals that the heavy penalization weights force the boosting algorithm to over-fragment the decision space in a desperate attempt to separate overlapping classes, directly explaining the precision collapse and boundary distortion observed in Section 3.2.

Figure 5. Aggregated feature importance per block. SMOTE-RADIANT preserves the stable feature hierarchy of the original data, whereas ClassWeight forces aggressive tree fragmentation, visibly inflating total split counts across all feature blocks (a) LightGBM (Original), (b) LightGBM with SMOTE-RADIANT, (c) LightGBM with ClassWeight, (d) XGBoost (Original), (e) XGBoost with SMOTE-RADIANT, and (f) XGBoost with ClassWeight

3.4. Statistical validation

To ensure observed test-set fluctuations are not artifacts of sampling variability, we employed a Bootstrapping procedure. We performed 1,000 bootstrap iterations (sampling with replacement) on predictions to generate highly stable Macro-F1 distributions. Paired two-sided Wilcoxon signed-rank tests and Cohen’s effect sizes (r) were then calculated to measure the practical magnitude of the differences. Using N=1,000 guarantees p-values that are exceptionally robust against random seed variations. Table 6 summarizes these findings.

Table 6. Bootstrap Macro-F1 Means (N=1,000). All comparisons against the Original configuration are highly significant ($p < 0.001$). The effect size (Cohen's r) is denoted in parentheses, alongside the direction of the performance shift (+ for increase, - for decrease)

Model	Original	SMOTE	ASTRA-SMOTE	SMOTE-RADIANT	ClassWeight
LightGBM	0.8525	0.8517 (r=0.36) (-)	0.8515 (r=0.44) (-)	0.8537 (r=0.51) (+)	0.8502 (r=0.58) (-)
XGBoost	0.8516	0.8522 (r=0.29) (+)	0.8509 (r=0.33) (-)	0.8523 (r=0.33) (+)	0.8504 (r=0.38) (-)

The statistical evidence completely refutes the assumption that all oversampling methods provide equivalent or negligible impacts in sparse text spaces. For LightGBM, the application of standard SMOTE, clustering-restricted ASTRA-SMOTE, and ClassWeight all resulted in statistically significant performance degradation (Sig. Decrease, $p < 0.001$) with medium to large effect sizes (r ranging from 0.36 to 0.58). This proves that unconstrained synthetic generation and algorithmic reweighting are actively harmful when dealing with highly overlapping 5,644-dimensional lexical features.

In stark contrast, SMOTE-RADIANT is the only configuration that yields a statistically significant performance increase for LightGBM, accompanied by a large positive effect size ($r = 0.511$). This establishes that the performance gains achieved by applying DBSCAN-based noise neutralization are not due to random chance, but represent a structurally sound and highly reproducible refinement of the decision boundaries.

For XGBoost, while standard SMOTE and SMOTE-RADIANT both registered statistically significant increases, their effect sizes were notably smaller ($r = 0.288$ and $r = 0.327$, respectively) compared to the gains seen in LightGBM. Furthermore, because XGBoost's initial performance was inherently lower than LightGBM's, these minor statistical gains were insufficient to make XGBoost the superior model.

Overall, this rigorous 1,000-iteration bootstrapping validation confirms that density-filtered oversampling (RADIANT) provides a practically meaningful improvement over both generative and cost-sensitive approaches, specifically when paired with leaf-wise gradient boosting architectures.

3.5. Comparison with transformer-based approaches

To contextualize the peak performance of the proposed methodology, Table 7 compares our champion model (LightGBM SMOTE-RADIANT) against previously reported neural and transformer-based approaches on the Shami dataset. For consistency, the per-class F1-scores of the recurrent, convolutional, and transformer models were recomputed as percentages based on the confusion matrices reported [23].

Table 7. Per-class F1-score (%) comparison against deep learning architectures on the Shami dataset

Approaches	Model	Jordan	Lebanon	Palestine	Syria	Macro-F1
Recurrent	GRU	64.07	79.93	73.65	93.23	77.72
Recurrent	LSTM	58.56	77.33	70.66	91.84	74.60
Convolutional	CNN	57.23	78.02	70.26	91.45	74.24
Transformer-based	Base-Arabert	72.14	83.28	80.94	94.82	82.79
Transformer-based	Arabic-XLM-R-Base	74.21	83.55	81.56	94.82	83.53
Transformer-based	Stacking-Transformer	78.73	83.98	82.63	95.13	85.12
Gradient Boosting (this study)	LightGBM (Original)	79.41	83.84	82.87	94.92	85.26
Gradient Boosting (this study)	LightGBM SMOTE-RADIANT	79.46	84.59	82.64	94.85	85.39

As shown in Table 7, recurrent and convolutional architectures struggle significantly, particularly in identifying the severely underrepresented Jordanian dialect. While individual transformer-based models like AraBERT improve overall performance, they still exhibit notable imbalance across dialects, with Jordanian F1 remaining relatively low (≈ 72.14%) [23]. Within our controlled experimental framework, the lightweight LightGBM SMOTE-RADIANT pipeline numerically outperforms heavy individual transformer architectures and is highly competitive with complex Stacking-Transformer models. Most notably, our pipeline achieves substantially higher recognition for the minority Jordanian dialect (79.46%). However, because experimental pipelines differ in preprocessing and optimization strategies, this comparison serves as a contextual reference point rather than a claim of absolute architectural superiority. Ultimately, rather than competing directly with large pretrained language models, this study proves that rigorous feature engineering combined with intelligent density-filtered oversampling offers a computationally efficient, interpretable, and highly competitive alternative for processing skewed Arabic texts.

3.6. Interpretation, dataset bias, and practical implications

The execution logs and cluster metrics provide profound insights into why density-based filtering (SMOTE-RADIANT) succeeds where clustering-guided generation (ASTRA-SMOTE) fails in the sparse,

5,644-dimensional space. ASTRA-SMOTE successfully achieved 100% of its synthetic generation targets by strictly interpolating inside "safe" minority clusters, resulting in exceptionally clean inter-class metrics (e.g., Davies-Bouldin index dropped significantly to 1.64). However, this overly conservative generation failed to improve the classifier. Gradient boosting trees rely on finding optimal splits at the boundaries between classes. Generating data exclusively deep inside safe clusters provides no new structural information to the model's decision boundary.

Conversely, SMOTE-RADIANT allowed standard interpolation across boundaries and subsequently utilized DBSCAN to autonomously detect and filter out overlapping noise. The logs reveal that RADIANT intentionally dropped 5,336 synthetic samples (an 18% drop rate) across the minority classes because they violated density heuristics. By sacrificing target volume to purge boundary-distorting anomalies, RADIANT maintained the necessary structural tension at the decision boundary, directly leading to the statistically significant performance increase observed in LightGBM.

This divergence also highlights that the effectiveness of oversampling is strongly model-dependent. LightGBM's leaf-wise growth enables localized splitting in minority-dense regions, allowing it to selectively exploit the clean synthetic boundary samples provided by RADIANT. Meanwhile, XGBoost's level-wise growth tends to propagate synthetic perturbations more uniformly across the tree, rendering it highly sensitive to boundary distortion even when density filtering is applied. Similar model-specific sensitivities have been reported in prior work [25], [29], [35]. Furthermore, while clustering-guided methods aim to restrict synthesis or filter outliers [30], [32], [33], their advantages do not always translate seamlessly to extremely sparse TF-IDF spaces, where standard density estimation can be unreliable without intelligent heuristic tuning [25], [33], [34].

Feature-level analysis supports this interpretation. Lexical features (TF–IDF and BM25) capture dialectal distinctions effectively [12], [13], while the newly engineered statistical features-such as NB-SVM priors and OOF stacking probabilities-provide powerful, complementary decision cues [17], [19]. These features interact constructively with SMOTE-RADIANT in LightGBM, solidifying the feature hierarchy without amplifying sensitivity to overlapping lexical noise [25], [29].

It is important to acknowledge that lexicon-based and frequency-driven representations may inadvertently encode topical, stance-related, or sentiment-bearing patterns in addition to purely structural dialectal markers. This is particularly true in social media corpora, where dialect usage often co-varies with affective expression and discussion themes [16], [24]. For instance, certain dialects may be disproportionately associated with specific political, cultural, or social topics, introducing latent topic bias into the feature distributions. However, because our 5,644-dimensional feature extraction pipeline and validation splits were strictly fixed across all rebalancing scenarios, any such bias remained constant throughout the experiments. Consequently, the comparative effects observed among the Original, SMOTE, ASTRA-SMOTE, SMOTE-RADIANT, and ClassWeight configurations can be reliably attributed to the imbalance-handling strategies themselves rather than uncontrolled shifts in lexical bias.

From a practical standpoint, achieving a statistically significant improvement in minority dialect recognition without degrading the majority class has profound real-world implications. Reliable dialect identification is a critical upstream component for downstream NLP applications such as sentiment analysis, machine translation, and speech technologies. By successfully lifting the recall of underrepresented dialects (e.g., Jordanian) safely, methodologies like RADIANT ensure that dialectal misclassification errors do not propagate through processing pipelines, thereby preventing AI systems from inadvertently marginalizing specific Arabic-speaking regions. Rather than directly competing with large pretrained models such as AraBERT, ARBERT, and MARBERT [20]-[23], this study provides methodological clarity by characterizing oversampling behavior within a transparent, interpretable feature-based framework [25], [35].

Several limitations remain. The analysis is restricted to the Shami Corpus and focuses on specific gradient-boosting architectures. Future work should extend evaluation to broader Arabic dialect benchmarks (e.g., NADI [3] and MADAR [4]), explore embedding-aware adaptive oversampling methods, and investigate hybrid cost-sensitive transformer-based approaches.

4. CONCLUSION

This study presented a rigorously controlled, leakage-free comparative evaluation of imbalance mitigation strategies for Arabic dialect identification on the Shami Corpus. Operating within a highly sparse 5,644-dimensional text representation, we demonstrated that conventional mitigation techniques-such as unconstrained SMOTE, cluster-restricted ASTRA-SMOTE, and cost-sensitive class weighting-largely failed to refine decision boundaries. These methods frequently sacrificed majority-class precision or introduced boundary-distorting noise in an attempt to artificially inflate minority recall.

However, SMOTE-RADIANT-a strategy that couples generative oversampling with density-based noise neutralization (DBSCAN)-successfully broke this dimensionality bottleneck. By autonomously filtering

out 18% of overlapping synthetic anomalies at the inter-class boundaries, RADIANT provided structurally sound data augmentation. Validated through a robust 1,000-iteration bootstrapping procedure on the held-out test set, SMOTE-RADIANT yielded a statistically significant macro-F1 improvement for LightGBM ($p < 0.001$) accompanied by a large effect size ($r = 0.511$). This provides compelling evidence that while generating synthetic text data in sparse spaces is inherently risky, strategically neutralizing overlapping noise can safely rescue minority dialects without majority-class degradation.

These findings clarify the complex trade-offs of imbalance mitigation, highlighting that model architecture (leaf-wise vs. level-wise), feature design, and oversampling strategies interact in non-trivial ways. Beyond dialect identification, this study underscores the necessity of carefully validated imbalance mitigation for building fair and robust Arabic NLP systems, ensuring that regional inclusivity is maintained in downstream applications.

ACKNOWLEDGMENTS

This work was supported by the Department of Computer Science and Electronics, Universitas Gadjah Mada under the Publication Funding Year 2026.

FUNDING INFORMATION

This work was supported by the Department of Computer Science and Electronics, Universitas Gadjah Mada under the Publication Funding Year 2026.

AUTHOR CONTRIBUTIONS STATEMENT

This study follows the Contributor Roles Taxonomy (CRediT). The contributions of each author are detailed as follows:

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Maulana Ihsan Ahmad	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓			
Aina Musdholifah	✓	✓		✓		✓	✓	✓		✓		✓	✓	
Arif Nurwidyantoro	✓			✓	✓		✓			✓		✓	✓	

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

DATA AVAILABILITY

The data that support the findings of this study are publicly available as the Shami Corpus [6]. To support transparent and reproducible research, the complete source code and experimental pipeline are publicly available at: https://github.com/moel-ihsan/arabic-dialect-identification.

REFERENCES

[1] Y. Matrane, F. Benabbou, and N. Sael, "A systematic literature review of Arabic dialect sentiment analysis," *Journal of King Saud University - Computer and Information Sciences*, vol. 35, no. 6, p. 101570, Jun. 2023, doi: 10.1016/j.jksuci.2023.101570.

[2] O. F. Zaidan and C. Callison-Burch, "Arabic dialect identification," *Computational Linguistics*, vol. 40, no. 1, pp. 171–202, Mar. 2014, doi: 10.1162/COLI_a_00169.

[3] M. Abdul-Mageed, C. Zhang, A. Elmadany, H. Bouamor, and N. Habash, "NADI 2022: The Third Nuanced Arabic dialect identification shared task," in *Proceedings of the Fifth Arabic Natural Language Processing Workshop*, Stroudsburg, PA, USA: Association for Computational Linguistics, 2020, pp. 97–110. doi: 10.18653/v1/2022.wanlp-1.9.

[4] H. Bouamor *et al*., "The madar Arabic dialect corpus and lexicon," in *LREC 2018 - 11th International Conference on Language Resources and Evaluation*, 2018, pp. 3387–3396.

[5] I. Alsarsour, E. Mohamed, R. Suwaileh, and T. Elsayed, "DART: A large dataset of dialectal Arabic tweets," in *LREC 2018 - 11th International Conference on Language Resources and Evaluation*, 2018, pp. 3666–3670.
[6] K. A. Kwaik, M. Saad, S. Chatzikyriakidis, and S. Dobnik, "Shami: a corpus of levantine Arabic dialects," in *LREC 2018 - 11th International Conference on Language Resources and Evaluation*, 2018, pp. 3645–3652.
[7] M. J. Althobaiti, "Automatic Arabic dialect identification systems for written texts: a survey," *International Journal of Computational Linguistics (IJCL)*, vol. 11, no. 3, 2020.
[8] E. Y. Alqulaity, W. M. S. Yafooz, A. Alourani, and A. Jaradat, "Arabic dialect identification in social media: a comparative study of deep learning and transformer approaches," *Intelligent Automation and Soft Computing*, vol. 39, no. 5, pp. 907–928, 2024, doi: 10.32604/iasc.2024.055470.
[9] A. A. Alsuwaylimi, "Arabic dialect identification in social media: A hybrid model with transformer models and BiLSTM," *Heliyon*, vol. 10, no. 17, p. e36280, Sep. 2024, doi: 10.1016/j.heliyon.2024.e36280.
[10] N. Baimukan, H. Bouamor, and N. Habash, "Hierarchical Aggregation of Dialectal Data for Arabic Dialect Identification," in *2022 Language Resources and Evaluation Conference, LREC 2022*, 2022, pp. 4586–4596.
[11] A. Alsudais, W. Alotaibi, and F. Alomary, "Similarities between Arabic dialects: Investigating geographical proximity," *Information Processing and Management*, vol. 59, no. 1, p. 102770, Jan. 2022, doi: 10.1016/j.ipm.2021.102770.
[12] N. N. A. Balaji and B. Bharathi, "Semi-supervised fine-grained approach for Arabic dialect detection task," in *Proceedings of the Fifth Arabic Natural Language Processing Workshop*, 2020, pp. 257–261.
[13] Z. Nassr, F. Benabbou, N. Sael, and T. Hamim, "Improving sentiment analysis performance on imbalanced Moroccan dialect datasets using resample and feature extraction techniques," *Information*, vol. 16, no. 1, p. 39, Jan. 2025, doi: 10.3390/info16010039.
[14] Y. Lv and C. Zhai, "Lower-bounding term frequency normalization," in *International Conference on Information and Knowledge Management, Proceedings*, New York, NY, USA: ACM, Oct. 2011, pp. 7–16. doi: 10.1145/2063576.2063584.
[15] W. Zaghouani, "Critical survey of the freely available Arabic corpora," in *Workshop on Free/Open-Source Arabic Corpora and Corpora Processing Tools*, 2014. doi: 10.13140/RG.2.1.1362.1284.
[16] M. Taboada, J. Brooke, M. Tofiloski, K. Voll, and M. Stede, "Lexicon-basedmethods for sentiment analysis," *Computational Linguistics*, vol. 37, no. 2, pp. 267–307, Jun. 2011, doi: 10.1162/COLI_a_00049.
[17] G. Badaro, R. Baly, H. Hajj, N. Habash, and W. El-Hajj, "A large scale arabic sentiment lexicon for Arabic opinion mining," in *ANLP 2014 - EMNLP 2014 Workshop on Arabic Natural Language Processing, Proceedings*, Stroudsburg, PA, USA: Association for Computational Linguistics, 2014, pp. 165–173. doi: 10.3115/v1/w14-3623.
[18] D. Schönle, C. Reich, and D. O. Abdeslam, "Linguistic driven feature selection for text classification as stop word replacement," *Journal of Advances in Information Technology*, vol. 14, no. 4, pp. 796–802, 2023, doi: 10.12720/jait.14.4.796-802.
[19] Y. Zhou, "A review of text classification based on deep learning," in *ICGDA '20: Proceedings of the 2020 3rd International Conference on Geoinformatics and Data Analysis*, New York, NY, USA: ACM, Apr. 2020, pp. 132–136. doi: 10.1145/3397056.3397082.
[20] W. Antoun, F. Baly, and H. Hajj, "AraBERT: transformer-based model for Arabic language understanding," in *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*, 2022, pp. 9–15. [Online]. Available: https://aclanthology.org/2020.osact-1.2/
[21] M. Abdul-Mageed, A. R. Elmadany, and E. M. B. Nagoudi, "ARBERT & MARBERT: deep bidirectional transformers for Arabic," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, Stroudsburg, PA, USA: Association for Computational Linguistics, 2021, pp. 7088–7105. doi: 10.18653/v1/2021.acl-long.551.
[22] B. AlKhamissi, M. Gabr, M. ElNokrashy, and K. Essam, "Adapting MARBERT for improved arabic dialect identification: submission to the NADI 2021 shared task," in *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, Association for Computational Linguistics, 2021, pp. 260–264. [Online]. Available: https://aclanthology.org/2021.wanlp-1.29/
[23] H. Saleh *et al.*, "Advancing arabic dialect detection with hybrid stacked transformer models," *Frontiers in Human Neuroscience*, vol. 19, Feb. 2025, doi: 10.3389/fnhum.2025.1498297.
[24] S. Wang and C. D. Manning, "Baselines and bigrams: Simple, good sentiment and topic classification," in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2012, pp. 90–94.
[25] M. Mujahid *et al.*, "Data oversampling and imbalanced datasets: an investigation of performance for machine learning and feature engineering," *Journal of Big Data*, vol. 11, no. 1, p. 87, Jun. 2024, doi: 10.1186/s40537-024-00943-4.
[26] H. Kudiabor, "AI's computing gap: academics lack access to powerful chips needed for research," *Nature*, vol. 636, no. 8041, pp. 16–17, Dec. 2024, doi: 10.1038/d41586-024-03792-6.
[27] M. Murat, "Unlocking the Full Potential of AI in the Middle East How Can AI Accelerate the Growth of the Region's Digital Economy?," AWS. Accessed: Feb. 26, 2026. [Online]. Available: https://pages.awscloud.com/rs/112-TZM-766/images/AWS-Whitepaper-Potential-AI-Middle-East.pdf
[28] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, Jun. 2002, doi: 10.1613/jair.953.
[29] C. Bunkhumpornpat, K. Sinapiromsaran, and C. Lursinsap, "Safe-level-SMOTE: Safe-level-synthetic minority over-sampling technique for handling the class imbalanced problem," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2009, pp. 475–482. doi: 10.1007/978-3-642-01307-2_43.
[30] Z. Xu, D. Shen, T. Nie, Y. Kou, N. Yin, and X. Han, "A cluster-based oversampling algorithm combining SMOTE and k-means for imbalanced medical data," *Information Sciences*, vol. 572, pp. 574–589, Sep. 2021, doi: 10.1016/j.ins.2021.02.056.
[31] S. N. Syakiylla Sayed Daud, R. Sudirman, and T. Wee Shing, "Safe-level SMOTE method for handling the class imbalanced problem in electroencephalography dataset of adult anxious state," *Biomedical Signal Processing and Control*, vol. 83, p. 104649, May 2023, doi: 10.1016/j.bspc.2023.104649.
[32] Y. Tao, Y. Zhang, and B. Jiang, "DBCSMOTE: a clustering-based oversampling technique for data-imbalanced warfarin dose prediction," *BMC Medical Genomics*, vol. 13, no. S10, p. 152, Oct. 2020, doi: 10.1186/s12920-020-00781-2.
[33] S. F. Taskiran, B. Turkoglu, E. Kaya, and T. Asuroglu, "A comprehensive evaluation of oversampling techniques for enhancing text classification performance," *Scientific Reports*, vol. 15, no. 1, p. 21631, Jul. 2025, doi: 10.1038/s41598-025-05791-7.
[34] A. Arafa, N. El-Fishawy, M. Badawy, and M. Radad, "RN-SMOTE: Reduced Noise SMOTE based on DBSCAN for enhancing imbalanced data classification," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 8, pp. 5059–5074, Sep. 2022, doi: 10.1016/j.jksuci.2022.06.005.
[35] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 9, pp. 1263–1284, Sep. 2009, doi: 10.1109/TKDE.2008.239.

BIOGRAPHIES OF AUTHORS

Maulana Ihsan Ahmad, S.Kom., S.Hum., M.Hum. received the S.Kom. degree in Computer Science from Universitas Syiah Kuala, Banda Aceh, Indonesia, in 2022, the S.Hum. degree in Humanities from UIN Ar-Raniry, Banda Aceh, Indonesia, in 2022, and the M.Hum. degree in Humanities from UIN Sunan Kalijaga, Yogyakarta, Indonesia, in 2025. He is currently pursuing the M.Cs. degree in Computer Science with the Department of Computer Science and Electronics, Faculty of Mathematics and Natural Sciences, Universitas Gadjah Mada, Yogyakarta, Indonesia. His research interests include Arabic linguistics, Arabic literature, natural language processing, machine learning, and geographic information systems (GIS). He can be contacted at email: maulanaihsanahmad@mail.ugm.ac.id.

Aina Musdholifah, S. Kom., M.Kom., Ph.D. received the S.Kom. degree in Computer Science from Universitas Gadjah Mada, Yogyakarta, Indonesia, in 2003, the M. Kom. degree in Computer Science from Universitas Gadjah Mada, Indonesia, in 2006, and the Ph.D. degree in Computer Science from Universiti Teknologi Malaysia, Johor, Malaysia, in 2013. She is currently an Assistant Professor with the Intelligent Systems Laboratory, Department of Computer Science and Electronics, Faculty of Mathematics and Natural Sciences, Universitas Gadjah Mada, Yogyakarta, Indonesia. Her research interests include genetic algorithms and fuzzy logic. She can be contacted at email: aina_m@ugm.ac.id.

Arif Nurwidyantoro, S.Kom., M.Cs., Ph.D. received the bachelor's degree in Computer Science from Institut Pertanian Bogor, Indonesia, the master's degree in Computer Science from Universitas Gadjah Mada, Yogyakarta, Indonesia, and the Ph.D. degree from Monash University, Melbourne, Australia. He is currently an Assistant Professor with the Software and Data Engineering Laboratory, Department of Computer Science and Electronics, Faculty of Mathematics and Natural Sciences, Universitas Gadjah Mada, Yogyakarta, Indonesia. His research interests include big data, data analytics, and software engineering. He can be contacted at email: arifn@ugm.ac.id.