CEPAT Journal of Computer Engineering: Progress, Application and Technology
Vol. 5, No. 2, November 2026, pp. 180~193
ISSN: 2963-6728, DOI: https://doi.org/10.25124/cepat.v5i02.12438

Backbone-Dependent Gains from Recurrent Classification Heads on IndoBERT for Mental Health Help Indication in Indonesian Tweets

Rivaldo Jonathan Paath[1], Oktavian A. Lantang[1], Pujo H. Saputro[1]
[1] Department of Informatics Engineering, Faculty of Engineering, Sam Ratulangi University, Indonesia

Article Info

Article history:

Received October 10, 2026
Revised November 09, 2026
Accepted November 09, 2026

Keywords:

Class Weighting
Threshold Tuning
Ablation Study
Mental Health Help Indication
Indonesian Tweets

ABSTRACT

Whether recurrent classification heads benefit pre-trained Transformers equally across backbones is unclear for informal Indonesian mental health text. We curated 33,641 tweets from a public, automatically labeled corpus into needs_help (need for professional help) and non_needs_help classes, using slang normalization without stopword removal or stemming. A three-seed 2×4 ablation compared IndoBERT-Base and IndoBERT-Lite with four heads: standard first-token, bidirectional long short-term memory (BiLSTM), BiLSTM with additive attention, and an extended head adding layer normalization, residual projection, and multi-sample dropout. On IndoBERT-Lite, BiLSTM-based heads raised threshold-tuned Macro-F1 by about 1.5 percentage points (pp) and area under the receiver operating characteristic curve (AUC) by about 1 pp. On IndoBERT-Base, AUC differed by under 0.2 pp across heads, and tuned Macro-F1 differences were small and inconclusive. Attention added no clear gain beyond the BiLSTM head, and the extended head did not reduce the dropout-free training–validation gap. Square-root class weighting shifted default operating points but did not measurably change AUC or tuned Macro-F1 in four configurations. Reference-style stopword-removal runs scored lower, but the training recipe also differed, so this cannot be attributed to stopword removal. Because labels were generated by rules, results may reflect rule learnability rather than clinical validity.

This is an open access article under the CC BY-SA license.

Corresponding Author:

Rivaldo Jonathan Paath
Department of Informatics Engineering
Faculty of Engineering, Sam Ratulangi University
Manado, Indonesia
Email: rivaldopaath026@student.unsrat.ac.id

1. INTRODUCTION

Depression contributes substantially to the global disability burden, yet many affected individuals do not access professional care [1], [2]. In Indonesia, a 2022 national adolescent survey reported that one in three adolescents experienced a mental health problem [3]. However, only 2.6% of those affected adolescents accessed professional mental health services [3]. Parallel to this service gap, platform X recorded 24.69 million active users in Indonesia in 2024 [4]. Young adults often use social media, including X, for emotional self-disclosure [5], [6]. Consequently, natural language processing (NLP) offers a scalable approach for mental health help indication (identifying needs for professional help) [7].

Classifying mental health help indication in Indonesian tweets involves distinct linguistic challenges. Indonesian social media posts contain informal syntax, phonetic abbreviations, colloquial slang, and code-

mixing [8], [9]. Psycholinguistic studies associate distress expressions with first-person singular pronouns, absolutist words, and explicit negative emotion terms [10], [11]. Early Indonesian studies used Support Vector Machine, Naïve Bayes, and Random Forest classifiers [8], [12], [13], [14]. These classifiers relied on term-frequency features, which do not model word order [8], [12], [13]. Furthermore, conventional stopword removal deletes pronouns and negations, while root stemming alters sub-word boundaries in Transformer tokenizers [9], [15].

Bidirectional Encoder Representations from Transformers (BERT) encodes context with bidirectional self-attention [16]. Its WordPiece tokenizer splits rare words into frequent sub-words [16]. Available Indonesian resources include IndoNLU and its IndoBERT models [17], IndoLEM and a separate IndoBERT model [18], and the Twitter-specific IndoBERTweet [19]. We use the IndoNLU models because they offer both a full-capacity and a compressed variant [17]. The compressed IndoBERT-Lite variant uses A Lite BERT (ALBERT) with cross-layer parameter sharing [17], [20].

In mental health NLP, researchers have explored domain-specific encoders such as MentalBERT [21] alongside hybrid classification heads and auxiliary features [22], [23], [24], [25]. Several architectures stack Bidirectional Long Short-Term Memory (BiLSTM) and attention layers above the first classification token ([CLS]) [22], [23]. Indonesian mental health studies differ in dataset size, annotation procedure, and preprocessing, which limits direct comparison [26], [27], [28], [29], [30]. Extended heads may bundle Layer Normalization [31] and Multi-Sample Dropout (MSD), which averages predictions across multiple dropout masks [32]. Consequently, their individual effects are hard to separate.

Pratama et al. [30] analyzed 40,000 Indonesian tweets labeled as needs_help (indication of a need for professional mental health help) and non_needs_help. Their study evaluated hyperparameter settings using English bert-base-uncased with stopword removal [30]. It remains unclear how progressive classification heads behave across full-capacity and compressed Indonesian backbones under controlled multi-seed evaluation. Similarly, the effects of algorithm-level class weighting and validation threshold tuning [33], [34] on this corpus require empirical verification. Accordingly, this study addresses three research questions: (1) Does the effect of recurrent and attentive classification heads depend on backbone capacity between IndoBERT-Base and IndoBERT-Lite? (2) Which head components BiLSTM, additive attention, or bundled Layer Normalization, residual projection, and MSD are associated with classification changes and training–validation gaps? (3) Does square-root class weighting alter ranking capability (AUC) or threshold-tuned classification performance compared to unweighted training?

To answer these questions, this study conducts a three-seed 2×4 ablation on indobenchmark/indobert-base-p1 and indobenchmark/indobert-lite-base-p1 [17]. First, we curate 33,641 Indonesian tweets from the public corpus using length filtering and exact deduplication. Non-Indonesian posts are filtered using FastText, a lightweight library for text classification and language identification [35]. We then apply lexicon-based slang normalization without stopword removal or stemming. Second, we evaluate four progressive heads with class-weighting and threshold-tuning ablations, measuring training–validation gaps without active dropout. Third, we compare reference-style stopword-removal setups against our stopword-retaining pipeline on the same data split. These comparisons are descriptive and do not isolate stopword removal nor replicate [30].

2. METHOD

This study evaluates four progressive classification heads across two pre-trained IndoBERT backbones for mental health help indication in Indonesian tweets. Figure 1 illustrates the experimental workflow, covering three-stage curation, slang normalization, stratified splitting, end-to-end multi-seed ablation training, and validation threshold tuning.

2.1. Dataset

This study uses the public Mental Health Help Indication Dataset Tweet X [36] constructed by Pratama et al. [30]. The raw repository contains 40,000 Indonesian tweets collected from X between July 2023 and July 2025 [30]. In the source study [30], tweets were categorized into two classes reflecting indications of a need for professional mental health help. These classes are labeled in the repository as non_needs_help (not help-indicated, Class 0, 73.00%) and needs_help (help-indicated, Class 1, 27.00%) [30], [36]. The labels were generated automatically using Term Frequency–Inverse Document Frequency (TF-IDF) scoring and rule-based linguistic cues [30]. No human clinical annotation or inter-annotator agreement coefficients were reported in the public release [30], [36]. Consequently, the labels represent rule-based linguistic indications rather than clinical diagnoses.

Before modeling, a three-stage curation audit was conducted to remove noisy, duplicate, and foreign-language entries, as summarized in Table 1. First, 3,760 overly short tweets containing three or fewer words were removed due to limited semantic context. Second, cross-row label consistency was audited across identical texts, revealing zero conflicting labels, followed by removing 287 exact duplicates. Third, tweets of

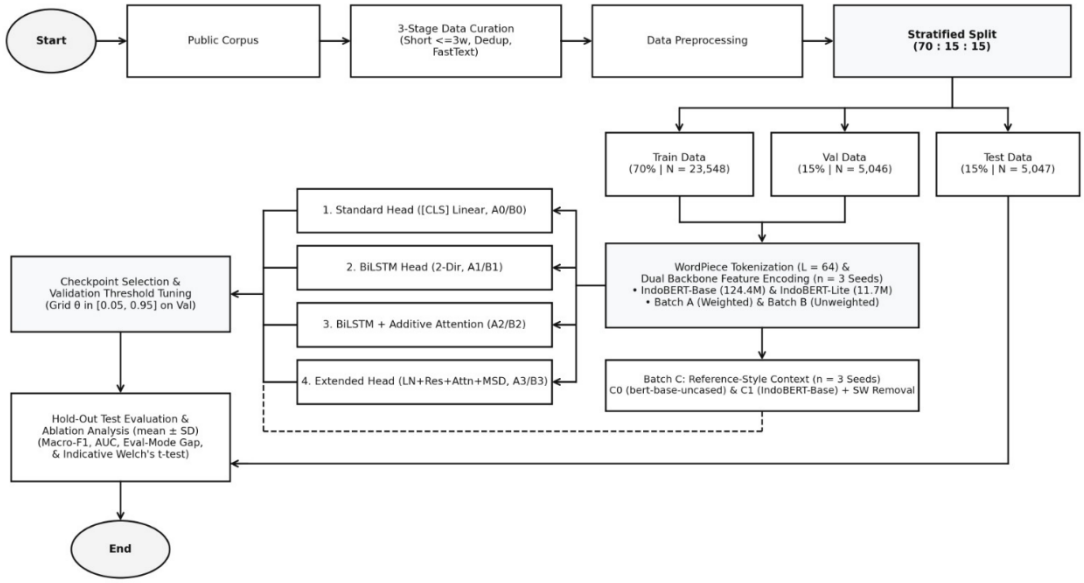

Figure 1. Experimental workflow of the multi-seed 2×4 classification head and class-weighting ablation study.

at least four words classified as non-Indonesian with confidence ≥ 0.90 by FastText (lid.176.ftz) [35] were removed (2,312 tweets). Posts identified as Indonesian (id), Malay (ms), or Javanese (jv) were retained to preserve colloquial and regional expressions. In total, 6,359 tweets (15.90%) were excluded, leaving 33,641 unique Indonesian tweets (84.10%).

Table 1. Three-stage dataset curation and filtering summary

Curation Stage	Remaining Tweets	Removed Tweets	Percentage Retained (%)
Initial public repository	40,000	-	100.00%
Short text removal (≤ 3 words)	36,240	3,760	90.60%
Label conflict audit and exact deduplication	35,953	287	89.88%
Foreign-language filtering via FastText (lid.176.ftz)	33,641	2,312	84.10%
Final Curated Corpus	**33,641**	**6,359 (15.90%)**	**84.10%**

The curated dataset contains 24,020 non_needs_help (not help-indicated, 71.40%) and 9,621 needs_help (help-indicated, 28.60%) tweets. This 2.50:1 ratio constitutes a mild class imbalance, as the minority class accounts for 28.60% of the corpus [37]. Because the public repository omits user identifiers for anonymity, the dataset was partitioned at the tweet level via stratified splitting (70:15:15, seed = 42). Consequently, potential user-level overlap across partitions cannot be ruled out. The split yielded 23,548 training (16,813 Class 0; 6,735 Class 1) and 5,046 validation (3,603 Class 0; 1,443 Class 1) tweets. The isolated hold-out test set contains 5,047 tweets (3,604 Class 0; 1,443 Class 1). All analyzed data are secondary, publicly available, and de-identified, and all example tweets shown in this paper are synthetic paraphrases.

2.2. Data Preprocessing

In the source repository [36], the released cleaned text column had already undergone removal of URLs, user mentions, hashtags, and non-ASCII symbols. Although [30] describes stopword removal during preprocessing, the released text column retains stopwords, so the exact text used by the labeling rules cannot be verified. We applied lowercase conversion, expanded alphanumeric reduplications, and removed remaining punctuation, standalone digits, and extra whitespace. Informal tokens were then standardized using a curated lexicon of 4,327 slang-to-standard word pairs. This lexicon normalization standardized colloquial tokens across 23,830 tweets (70.84% of the curated corpus).

Our main preprocessing pipeline omitted morphological stemming and stopword removal to retain personal pronouns, negations, and absolutist qualifiers [9], [15]. We audited the 33,641 post-normalization tweets (526,694 total tokens) using the default 126-entry PySastrawi stopword list. This audit showed that 32,024 normalized tweets (95.19%) contained Sastrawi stopwords, accounting for 32.81% of all corpus tokens. To examine whether removing these tokens affects performance, we compare our stopword-retaining pipeline against a reference-style stopword-removal setup in Section 3.6. Stopword removal may contribute to performance differences by stripping pronouns and negations, although this mechanism remains a hypothesis because training recipes also differed. Table 2 illustrates the sequential preprocessing stages on a synthetic example tweet alongside the actual PySastrawi stopword removal output.

Table 2. Illustration of the preprocessing pipeline on a synthetic example tweet compared with PySastrawi stopword removal

Preprocessing Stage	Resulting Text Representation
Synthetic Raw Tweet	*Aku udh pengen mati di tikam tangan sendiri. Kau menahan ku dgn tangan kiri. Udh berkali2 kau nangis tp gak ingin memutus nadi. Aku yg gak pernah nangis malah pengen bunuh diri. https://t.co/27n4cqouRW*
Repository cleaning	*aku udh pengen mati tikam tangan sendiri kau menahan ku dgn tangan kiri udh berkali2 kau nangis tp gak ingin memutus nadi aku yg gak pernah nangis malah pengen bunuh diri*
Slang normalization	*aku sudah ingin mati tikam tangan sendiri kau menahan ku dengan tangan kiri sudah berkali-kali kau menangis tapi tidak ingin memutus nadi aku yang tidak pernah menangis malah ingin bunuh diri (Restores standard pronouns, conjunctions, and negations from slang)*
IndoBERT WordPiece tokenization	*['[CLS]', 'aku', 'sudah', 'ingin', 'mati', 'tika', '##m', 'tangan', 'sendiri', 'kau', 'menahan', 'ku', 'dengan', 'tangan', 'kiri', 'sudah', 'berkali', '-', 'kali', 'kau', 'menangis', 'tapi', 'tidak', 'ingin', 'memutus', 'nadi', 'aku', 'yang', 'tidak', 'pernah', 'menangis', 'malah', 'ingin', 'bunuh', 'diri', '[SEP]']*
Comparison: PySastrawi stopword removal (StopWordRemoverFactory, 126-entry default list)	*mati tikam tangan kau menahan ku tangan kiri kau menangis memutus nadi menangis bunuh*

Across the 33,641 normalized tweets, word counts ranged from 4 to 68 words (mean = 15.7, median = 12.0, 99th percentile = 43.0). When segmented by the IndoBERT WordPiece tokenizer, sub-word sequences averaged 20.8 tokens. Because the 99th percentile was 55 tokens, the maximum length was set to L = 64 tokens, covering 99.64% of tweets without truncation.

2.3. Classification Head Variants

Each mini-batch of size B = 32 and length L = 64 is encoded by an IndoNLU backbone [17]. The first backbone is IndoBERT-Base (A0–A3), a 12-layer bidirectional Transformer encoder with D = 768 and 124.4 million parameters. The second is IndoBERT-Lite (B0–B3), a 12-layer compressed encoder (11.7 million parameters) based on A Lite BERT (ALBERT) [20]. ALBERT reduces parameter count via factorized embedding parameterization and cross-layer parameter sharing [20]. Given the final encoder hidden states $H_{\text{BERT}} \in R^{B\times L\times D}$, four progressive classification heads were constructed on both backbones.

Head 1: Standard Head (A0, B0). All recurrent, attention, and extra normalization layers are omitted. The head extracts the first classification token ([CLS]) vector $h_1 \in R^{B\times D}$, applies dropout (p = 0.30), and projects logits via a linear layer [16].

Head 2: BiLSTM Head (A1, B1). The full sequence H_{BERT} is fed into a single-layer Bidirectional Long Short-Term Memory (BiLSTM) network. The BiLSTM uses hidden size d = 128 per direction (2d = 256 total). Dynamic sequence packing is applied so that padding tokens do not update the recurrent states. The final forward and backward hidden states are concatenated into $h_{\text{concat}} = [h_n; \overleftarrow{h_n}] \in R^{B\times 256}$ followed by dropout (p = 0.30) and a linear classifier.

Head 3: BiLSTM with Additive Attention Head (A2, B2). All unpacked BiLSTM hidden states $H_{\text{LSTM}} = [h_1, \ldots, h_L] \in R^{B\times L\times 256}$ enter an additive soft attention module [38]. For each token position t, the attention energy score e_t and normalized weight α_t are computed as in (1) and (2):

$$e_t = v^T \tanh(W_h h_t + b_h) \quad (1)$$

$$\alpha_t = \frac{\exp(e_t)}{\sum_{k=1}^{L} \exp(e_k)} \quad (2)$$

Where $W_h \in R^{128\times 256}$, $b_h \in R^{128}$, and $v \in R^{128}$ are learnable parameters, and padding positions are masked with -10^9 before softmax. The weighted context vector $C = \sum_{t=1}^{L} \alpha_t h_t \in R^{B\times 256}$ passes through a dropout layer (p = 0.30) and a linear classifier.

Head 4: Extended Head (A3, B3). This stage adds a bundled stabilization block around the BiLSTM-attention architecture. First, H_{BERT} is normalized via pre-recurrent Layer Normalization (LayerNorm) [31] over D = 768 to obtain X_{norm}. LayerNorm standardizes activations across features, and is followed here by a pre-BiLSTM dropout layer (p = 0.30). Second, the BiLSTM output is summed with a linear residual projection of X_{norm} and regularized by intermediate dropout (p = 0.30), as in (3):

$$H_{\text{res}} = \text{Dropout}_{\text{mid}}\left(\text{BiLSTM}\left(\text{Dropout}_{\text{pre}}(X_{\text{norm}})\right) + (X_{\text{norm}} W_{\text{res}} + b_{\text{res}})\right) \quad (3)$$

In (3), $W_{\text{res}} \in R^{768\times 256}$ and $b_{\text{res}} \in R^{256}$ project the normalized encoder states into the 256-dimensional BiLSTM space. Third, after computing the attention context vector C, post-attention LayerNorm is applied over 2d = 256 to obtain Z_{post}. Finally, Z_{post} enters a five-branch Multi-Sample Dropout (MSD) module [32] with rates from 0.20 to 0.40 sharing one linear classifier, as in (4):

$$\text{Logits} = \frac{1}{5}\sum_{m=1}^{5}\left(W_{\text{cls}}\,\text{Dropout}_m\left(Z_{\text{post}}\right) + b_{\text{cls}}\right) \quad (4)$$

Because dual LayerNorm, residual projection, and MSD are bundled into one stage, their individual sub-component effects are not isolated in this study.

2.4. Experimental Design and Training Protocol

All experiments were conducted on the fixed 70:15:15 stratified split across three random seeds (42, 100, and 2026), as summarized in Table 3. All backbone and classification head parameters were fine-tuned end-to-end for 5 epochs (batch size B = 32) using the AdamW optimizer [39]. Optimization used decoupled weight decay ($\lambda = 0.03$) and gradient clipping (maximum norm = 1.0). We applied differential learning rates: $\eta_{\text{BERT}} = 6.5 \times 10^{-6}$ for the pre-trained backbone and $\eta_{\text{head}} = 2.0 \times 10^{-4}$ for the classification head. A Cosine Annealing schedule with a 20% linear warm-up was used.

To examine class weighting and preprocessing context, three experimental batches totaling 42 training runs were executed:

Batch A (Weighted 2×4 Ablation, 24 runs) evaluates A0–A3 and B0–B3 using square-root smoothed class weights from the training split, as in (5):

$$w'_c = \sqrt{\frac{N_{\text{train}}}{K \cdot N_{c,\text{train}}}} \quad (5)$$

With $N_{\text{train}} = 23{,}548$, K = 2, $N_{0,\text{train}} = 16{,}813$, and $N_{1,\text{train}} = 6{,}735$, (5) yields $w'_0 = 0.8368$ and $w'_1 = 1.3222$. These weights were applied inside a cross-entropy loss with label smoothing $\epsilon = 0.05$ [40], which softens target distributions to limit extreme logit growth.

Batch B (Unweighted Ablation, 12 runs) evaluates A0, A3, B0, and B2 using unweighted cross-entropy loss ($w'_0 = w'_1 = 1.0$). Label smoothing ($\epsilon = 0.05$) and all other hyperparameters remained identical to Batch A to isolate the effect of class weighting.

Batch C (6 runs) evaluates C0 (bert-base-uncased) and C1 (indobert-base-p1) with standard [CLS] heads over three seeds. Both applied Sastrawi stopword removal after slang normalization, with unweighted loss, no label smoothing, and the uniform learning rate of 5×10^{-5} reported in [30]. Like Batches A and B, C0 and C1 used AdamW ($\lambda = 0.03$, gradient clipping 1.0), Cosine Annealing (20% warm-up). Unlike [30]

(three epochs), we trained five epochs, selected checkpoints via validation $F1_1$, and evaluated at $\theta = 0.50$ and θ_{opt}. C1 therefore differs from unweighted A0 (Batch B) in stopword removal, learning-rate scheme, and label smoothing. Batch C provides context, not a replication of [30].

Table 3. Hyperparameter, experimental batch, and hardware configuration

Parameter	Specification / Value
Random seed	3 seeds: 42, 100, and 2026 (n = 3 runs per configuration; 42 total runs)
Software environment	Linux Pop!_OS 22.04, Python 3.10.21, PyTorch 2.2, HuggingFace Transformers 4.38, Scikit-learn 1.4, PySastrawi 1.2
Hardware environment	NVIDIA GeForce RTX 4050 GPU (6 GB GDDR6), Intel Core i7-14650HX, 16 GB DDR5 RAM
Pre-trained backbones	indobenchmark/indobert-base-p1 (124.4M) and indobert-lite-base-p1 (11.7M)
Maximum sequence length (L)	64 WordPiece sub-word tokens including [CLS] and [SEP] (99.64% coverage; $P_{99} = 55$)
Dataset split (Train : Val : Test)	70 : 15 : 15 stratified at seed = 42 (23,548 : 5,046 : 5,047 tweets; fixed across all 42 runs)
Batch size (B) / Training epochs	32 / 5 epochs across Batches A, B, and C (best checkpoint selected via validation $F1_1$)
Optimizer and regularization	AdamW, decoupled weight decay $\lambda = 0.03$, gradient clipping maximum norm = 1.0
Learning rates (Batches A & B)	Differential: $\eta_{\text{BERT}} = 6.5 \times 10^{-6}$ (Backbone); $\eta_{\text{head}} = 2.0 \times 10^{-4}$ (Head)
Learning rate schedule	Cosine Annealing with linear warm-up over the first 20% of training steps
Dropout rates (Batches A & B)	Head 1–3 dropout: 0.30; Head 4 pre/mid dropout: 0.30; Head 4 MSD (5 branches): $\{0.20, 0.25, 0.30, 0.35, 0.40\}$
Loss & class weights (Batch A)	Smoothed weights $w_0' = 0.8368$, $w_1' = 1.3222$ with label smoothing $\epsilon = 0.05$ (A0–A3, B0–B3)
Loss & class weights (Batch B)	Unweighted ($w_0' = w_1' = 1.0$) with label smoothing $\epsilon = 0.05$ (A0, A3, B0, B2)
Reference-style setup (Batch C)	CO (bert-base-uncased) & C1 (indobert-base-p1): uniform LR 5×10^{-5}, Sastrawi stopword removal, unweighted CE, $\epsilon = 0.0$, 5 epochs, evaluated at $\theta = 0.50$ and θ_{opt}
Threshold moving range (θ_{opt})	Grid search $\theta \in [0.05, 0.95]$ with step size 0.05 on the validation set

2.5. Evaluation Metrics and Statistical Analysis

For each run across all three batches, the checkpoint achieving the highest validation positive-class F1-score ($F1_1$) over 5 epochs was selected. On that best checkpoint, the probability threshold θ for predicting Class 1 was swept across $[0.05, 0.95]$ (step 0.05) on the validation set. The optimal threshold θ_{opt} was selected by maximizing validation $F1_1$. Both the default threshold ($\theta = 0.50$) and θ_{opt} were then locked and evaluated once on the test set ($N_{\text{test}} = 5{,}047$)

Following standard classification evaluation [41], [42], performance is derived from True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN). From these four counts, Accuracy, Precision, Recall, $F1_1$, and Macro-F1 are computed as in (6)–(10):

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{6}$$

$$Precision_1 = \frac{TP}{TP + FP} \tag{7}$$

$$Recall_1 = \frac{TP}{TP + FN} \tag{8}$$

$$F1_1 = \frac{2 \times \text{Precision}_1 \times \text{Recall}_1}{\text{Precision}_1 + \text{Recall}_1} \tag{9}$$

$$\text{Macro F1} = \frac{F1_0 + F1_1}{2} \tag{10}$$

Macro-F1 and AUC serve as primary metrics. Positive-class precision and recall at θ_{opt} are treated as exploratory because threshold variation across seeds introduces larger standard deviations.

To evaluate overfitting without dropout distortion, we compute train-mode and evaluation-mode Macro-F1 gaps at $\theta = 0.50$ on the selected checkpoint. The train-mode gap subtracts validation Macro-F1 ($N_{\text{val}} = 5{,}046$) from full-training-set Macro-F1 ($N_{\text{train}} = 23{,}548$) recorded with active dropout. Conversely, the evaluation-mode gap recomputes full-training-set Macro-F1 deterministically with dropout disabled before subtracting validation Macro-F1. Because the validation set is used to select the best checkpoint, validation scores are slightly optimistic relative to the hold-out test set.

All metrics are reported as mean $\pm$ sample standard deviation across n = 3 seeds. Metric differences between configurations are expressed in percentage points (pp). Pairwise comparisons use two-sided Welch's t-tests [43] without multiple-comparison correction. Given the small sample size (n = 3 seeds), all p-values are interpreted strictly as indicative, and large p-values do not imply equivalence.

3. RESULTS

3.1. Overall Classification Results

Table 4 reports the test Macro-F1 and AUC scores across all eight weighted configurations (Batch A) over three random seeds. Results are presented at both the default probability threshold ($\theta = 0.50$) and the validation-tuned threshold (θ_{opt}). Across all four head variants, IndoBERT-Base configurations achieved higher Macro-F1 and AUC scores than their IndoBERT-Lite counterparts. Because validation threshold tuning shifts operating points across seeds, positive-class precision and recall at θ_{opt} show higher variance and are treated as exploratory.

Table 4. Test performance across the eight weighted configurations (Batch A, mean $\pm$ SD, n = 3 seeds, $N_{\text{test}} = 5{,}047$)

Code	Acc (%) ($\theta = 0.50$)	Prec$_1$ (%) ($\theta = 0.50$)	Rec$_1$ (%) ($\theta = 0.50$)	F1$_1$ (%) ($\theta = 0.50$)	Macro-F1 (%) ($\theta = 0.50$)	AUC-ROC (%)	θ_{opt} (Val)	Acc (%) (θ_{opt})	F1$_1$ (%) (θ_{opt})	Macro-F1 (%) (θ_{opt})
A0	89.74 ± 0.29	78.01 ± 1.79	89.42 ± 2.19	83.29 ± 0.11	87.95 ± 0.18	96.66 ± 0.19	0.62 ± 0.10	90.49 ± 0.48	83.91 ± 0.59	88.58 ± 0.48
A1	90.28 ± 0.35	80.04 ± 2.61	88.17 ± 3.12	83.84 ± 0.18	88.45 ± 0.20	96.75 ± 0.24	0.63 ± 0.13	91.16 ± 0.46	84.73 ± 0.83	89.25 ± 0.57
A2	90.32 ± 0.31	79.01 ± 1.12	90.09 ± 0.79	84.18 ± 0.33	88.60 ± 0.29	96.79 ± 0.13	0.70 ± 0.00	91.40 ± 0.18	85.19 ± 0.33	89.57 ± 0.21
A3	90.17 ± 0.16	78.85 ± 0.26	89.65 ± 0.79	83.90 ± 0.32	88.41 ± 0.21	96.80 ± 0.04	0.68 ± 0.03	91.05 ± 0.27	84.51 ± 0.23	89.11 ± 0.23
B0	86.98 ± 0.35	72.94 ± 1.17	86.63 ± 1.51	79.18 ± 0.38	84.85 ± 0.32	94.63 ± 0.16	0.70 ± 0.05	88.44 ± 0.38	79.64 ± 0.05	85.78 ± 0.18
B1	89.10 ± 0.20	78.37 ± 0.14	85.47 ± 0.80	81.77 ± 0.40	87.00 ± 0.26	95.52 ± 0.09	0.60 ± 0.05	89.70 ± 0.13	81.95 ± 0.28	87.37 ± 0.11
B2	89.07 ± 0.18	78.28 ± 0.55	85.49 ± 0.21	81.73 ± 0.22	86.96 ± 0.18	95.65 ± 0.10	0.58 ± 0.03	89.57 ± 0.22	81.87 ± 0.46	87.27 ± 0.30
B3	88.20 ± 0.41	76.95 ± 1.64	83.95 ± 1.92	80.27 ± 0.54	85.93 ± 0.40	95.11 ± 0.22	0.60 ± 0.05	88.92 ± 0.53	80.61 ± 0.69	86.43 ± 0.53

3.2. Head Effects Across Backbones

To address RQ1, Table 5 and Figure 2 compare each modified head against the standard head (A0 or B0). On the compressed IndoBERT-Lite backbone, adding a BiLSTM head produced clear, indicative improvements across both primary metrics. Relative to B0, B1 (+BiLSTM) increased threshold-tuned Macro-F1 by $+1.59$ pp (p = 0.001) and AUC by $+0.89$ pp (p = 0.003). Similarly, B2 (+BiLSTM+Attention) increased threshold-tuned Macro-F1 by $+1.49$ pp (p = 0.004) and AUC by $+1.02$ pp (p = 0.002). At $\theta = 0.50$, B1 and B2 improved Macro-F1 over B0 by $+2.15$ pp (p = 0.001) and $+2.11$ pp (p = 0.002), respectively. This head gain persisted without class weighting (Batch B), where unweighted B2 exceeded unweighted B0 by $+1.77$ pp in tuned Macro-F1 (p = 0.008). Unweighted B2 also improved AUC over unweighted B0 by $+1.17$ pp (p = 0.002). Overall, adding a recurrent head indicatively closed roughly half of the baseline Lite–Base gap (53.2%–56.8% on tuned Macro-F1; 43.8%–50.2% on AUC).

In contrast, on IndoBERT-Base, AUC differences across all four heads were at most 0.14 pp and were not statistically indicative ($p \geq 0.329$). At $\theta = 0.50$, Base heads differed from A0 by $+0.46$ to $+0.65$ pp in Macro-F1 (p = 0.033–0.046, uncorrected). However, these differences weakened after validation threshold tuning, with A2 showing the largest difference ($+0.99$ pp, p = 0.053, uncorrected). Across the 12 pairwise comparisons in Table 5 (Bonferroni threshold ≈ 0.004), only the head gains on IndoBERT-Lite approached that threshold. Without class weighting (Batch B), A3 differed from A0 by $+0.01$ pp in tuned Macro-F1 (p = 0.966) and $+0.05$ pp in AUC.

3.3. Component Ablation: BiLSTM, Attention, and Bundled Stabilization

To address RQ2, Table 5 compares consecutive head stages within each backbone to isolate individual module contributions. Adding additive soft attention above the BiLSTM layer did not produce an isolable Macro-F1 gain on either backbone. In tuned Macro-F1, B2 differed from B1 by -0.10 pp (p = 0.632), while A2 differed from A1 by $+0.32$ pp (p = 0.440). Similarly, AUC differences between

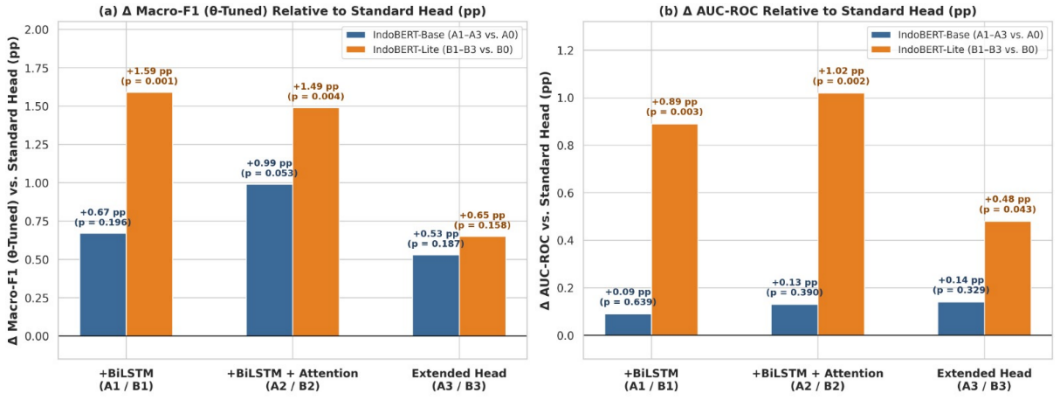

Figure 2. Incremental classification head gains (Δ pp) relative to the standard head (A0/B0) across IndoBERT-Base and IndoBERT-Lite: (a) threshold-tuned Macro-F1 and (b) AUC-ROC (n = 3 seeds).

Table 5. Pairwise classification head comparisons and indicative two-sided Welch's t-test results (n = 3 seeds)

Comparison Pair	Loss Condition	Δ Macro-F1 (θ = 0.50, pp)	Indicative p-value (θ = 0.50)	Δ Macro-F1 (θ_{opt}, pp)	Indicative p-value (θ_{opt})	Δ AUC-ROC (pp)	Indicative p-value (AUC)
A1 (+BiLSTM) vs. A0 (Std)	Weighted	+0.50	0.033	+0.67	0.196	+0.09	0.639
A2 (+BiLSTM+Attn) vs. A0 (Std)	Weighted	+0.65	0.039	+0.99	0.053	+0.13	0.390
A3 (Extended) vs. A0 (Std)	Weighted	+0.46	0.046	+0.53	0.187	+0.14	0.329
A2 (+Attn) vs. A1 (+BiLSTM)	Weighted	+0.15	0.507	+0.32	0.440	+0.04	0.816
A3 (Extended) vs. A2 (+Attn)	Weighted	-0.19	0.415	-0.46	0.063	+0.01	0.909
A3 (Extended) vs. A0 (Std)	Unweighted	+0.28	0.494	+0.01	0.966	+0.05	0.474
B1 (+BiLSTM) vs. B0 (Std)	Weighted	+2.15	0.001	+1.59	0.001	+0.89	0.003
B2 (+BiLSTM+Attn) vs. B0 (Std)	Weighted	+2.11	0.002	+1.49	0.004	+1.02	0.002
B3 (Extended) vs. B0 (Std)	Weighted	+1.08	0.024	+0.65	0.158	+0.48	0.043
B2 (+Attn) vs. B1 (+BiLSTM)	Weighted	-0.04	0.839	-0.10	0.632	+0.13	0.170
B3 (Extended) vs. B2 (+Attn)	Weighted	-1.03	0.031	-0.84	0.092	-0.54	0.035
B2 (+BiLSTM+Attn) vs. B0 (Std)	Unweighted	+1.76	0.004	+1.77	0.008	+1.17	0.002

BiLSTM+Attention and BiLSTM alone were small on Lite (+0.13 pp, p = 0.170) and Base (+0.04 pp, p = 0.816). Furthermore, the bundled extended head (A3 and B3) added no performance benefit over the BiLSTM-attention head. On IndoBERT-Base, A3 scored 0.46 pp lower than A2 in tuned Macro-F1 (p = 0.063) with similar AUC (+0.01 pp, p = 0.909). On IndoBERT-Lite, B3 scored lower than B2 by −1.03 pp in default Macro-F1 (p = 0.031). On Lite, B3 also trailed B2 in tuned Macro-F1 (−0.84 pp, p = 0.092) and AUC (−0.54 pp, p = 0.035). We hypothesize that stacking pre/mid dropout and five-branch MSD onto shared-parameter encoder layers over-regularizes the compressed 11.7M-parameter backbone.

3.4. Effect of Class Weighting

To address RQ3, Table 6 presents a 2×2 comparison (weighting by threshold) across the four configurations tested in Batch B (A0, A3, B0, and B2). Across these four configurations, square-root class weighting did not measurably change threshold-independent AUC ($|\Delta| \leq 0.09$ pp, $p \geq 0.305$). Likewise, after validation threshold tuning, Macro-F1 showed no indicative change ($|\Delta| \leq 0.46$ pp, largest on A0 at p = 0.233). Instead, class weighting shifted the default operating point at $\theta = 0.50$ by increasing positive-class recall while lowering positive-class precision. Consequently, at $\theta = 0.50$, weighting lowered Macro-F1 on A0 (−0.51 pp, p = 0.038) and A3 (−0.33 pp, p = 0.428). Similarly, at $\theta = 0.50$, weighting lowered Macro-F1 on B0 (−0.67 pp, p = 0.078) and B2 (−0.32 pp, p = 0.244). Three of the four unweighted models already favored recall over precision at $\theta = 0.50$ (e.g., 88.27% vs. 79.89% on A0). The fourth unweighted model (B2) was nearly balanced at $\theta = 0.50$ (81.40% recall vs. 82.20% precision). Thus, unweighted training on this corpus did not exhibit prediction collapse toward the majority class.

Table 6. Comparison of square-root class weighting (Batch A) versus unweighted training (Batch B) across four configurations (mean ± SD, n = 3 seeds)

Code	Loss Weighting	Prec$_1$ (%) (θ = 0.50)	Rec$_1$ (%) (θ = 0.50)	Macro-F1 (%) (θ = 0.50)	Δ Macro-F1 (θ = 0.50, W − NoW)	θ_{opt}	Macro-F1 (%) (θ_{opt})	Δ Macro-F1 (θ_{opt}, W − NoW)	AUC-ROC (%)	Δ AUC (W − NoW)
A0	Unweighted	79.89 ± 0.67	88.27 ± 0.24	88.46 ± 0.22	–	0.65 ± 0.05	89.04 ± 0.17	–	96.70 ± 0.09	–
A0	Weighted (w_c')	78.01 ± 1.79	89.42 ± 2.19	87.95 ± 0.18	-0.51 pp (p = 0.038)	0.62 ± 0.10	88.58 ± 0.48	-0.46 pp (p = 0.233)	96.66 ± 0.19.	-0.04 pp (p = 0.764)
A3	Unweighted	80.81 ± 2.26	87.96 ± 1.29	88.74 ± 0.57	–	0.67 ± 0.03	89.05 ± 0.33	–	96.75 ± 0.06	–
A3	Weighted (w_c')	78.85 ± 0.26	89.65 ± 0.79	88.41 ± 0.21	-0.33 pp (p = 0.428)	0.68 ± 0.03	89.11 ± 0.23	+0.06 pp (p = 0.810)	96.80 ± 0.04	+0.05 pp (p = 0.305)
B0	Unweighted	77.08 ± 1.10	82.33 ± 1.14	85.52 ± 0.37	–	0.55 ± 0.05	85.60 ± 0.40	–	94.54 ± 0.19	–
B0	Weighted (w_c')	72.94 ± 1.17	86.63 ± 1.51	84.85 ± 0.32	-0.67 pp (p = 0.078)	0.70 ± 0.05	85.78 ± 0.18	+0.18 pp (p = 0.532)	94.63 ± 0.16	+0.09 pp (p = 0.565)
B2	Unweighted	82.20 ± 0.31	81.40 ± 0.94	87.28 ± 0.34	–	0.52 ± 0.03	87.37 ± 0.46	–	95.71 ± 0.13	–
B2	Weighted (w_c')	78.28 ± 0.55	85.49 ± 0.21	86.96 ± 0.18	-0.32 pp (p = 0.244)	0.58 ± 0.03	87.27 ± 0.30	-0.10 pp (p = 0.771)	95.65 ± 0.10	-0.06 pp (p = 0.563)

3.5. Training–Validation Overfitting Gaps

Table 7 and Figure 3 compare training–validation Macro-F1 gaps measured in train-mode (dropout active) versus evaluation-mode (clean, dropout disabled). In evaluation-mode, the clean Macro-F1 gaps across A0–A3 were +3.76, +4.30, +5.31, and +3.95 pp, respectively. On IndoBERT-Lite, the corresponding evaluation-mode Macro-F1 gaps across B0–B3 were +0.93, +1.72, +2.27, and +0.93 pp. Active dropout in train-mode lowers training scores, which is consistent with the negative gaps on B0 (−0.58 pp) and B3 (−0.38 pp). When measured deterministically in evaluation-mode, the extended head did not reduce the training–validation gap on either backbone. Specifically, the clean Macro-F1 gap was +3.95 pp on A3 versus +3.76 pp on A0 ($p = 0.599$). On IndoBERT-Lite, the clean Macro-F1 gap was +0.93 pp on both B3 and B0 ($p = 0.988$).

Adding BiLSTM and attention tended to increase the clean Macro-F1 gap on Base (A2 vs. A0: +1.55 pp, $p = 0.031$). A similar clean-gap increase occurred on Lite (B2 vs. B0: +1.34 pp, $p < 0.001$). Unweighted runs showed the same pattern: A3 vs. A0 (−0.31 pp, $p = 0.335$) and B2 vs. B0 (+1.73 pp, $p = 0.014$). However, on IndoBERT-Lite, larger clean gaps on B1 and B2 coincided with higher validation and test scores. Across three seeds, the train-mode positive-class F1 gap was larger on A3 than on A0 (+4.09 vs. +2.52 pp, $p = 0.374$). Because validation data were used to select the best checkpoint and θ_{opt}, validation scores remain slightly optimistic relative to the test set.

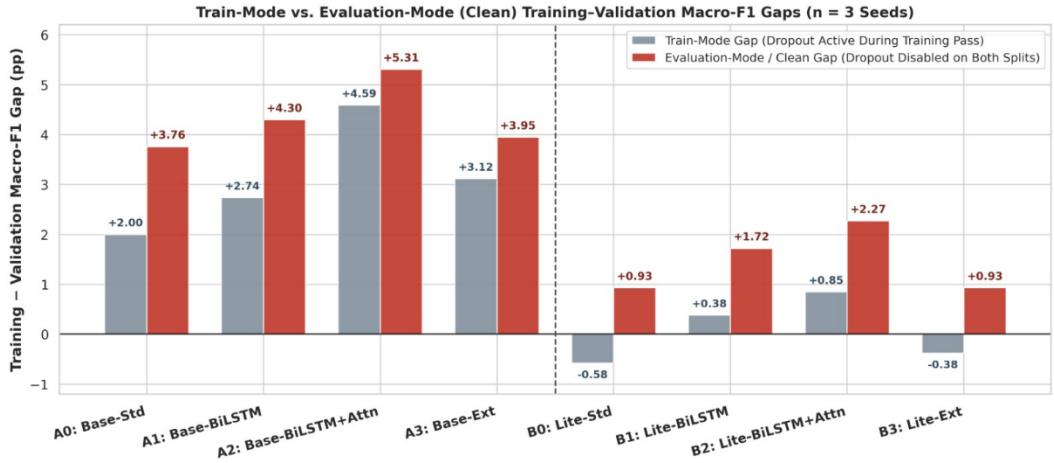

Figure 3. Comparison of train-mode (dropout active) versus evaluation-mode (clean, dropout disabled) training–validation Macro-F1 gaps across all eight configurations (mean ± SD, n = 3 seeds).

Table 7. Training–validation gaps in train-mode versus evaluation-mode (clean, no-dropout) across Batch A (mean ±SD, n = 3 seeds)

Code	Train Acc (Train-Mode, %)	Train Acc Clean (Eval-Mode, %)	Val Acc (Best, %)	Gap Acc Train-Mode (pp)	Gap Acc Clean (pp)	Gap $F1_1$ Train-Mode (pp)	Gap $F1_1$ Clean (pp)	Gap Macro-F1 Train-Mode (pp)	Gap Macro-F1 Clean (pp)
A0	92.43 ± 1.23	93.72 ± 0.09	90.51 ± 0.52	+1.91 ± 1.67	+3.21 ± 0.51	+2.52 ± 2.40	+5.19 ± 0.65	+2.00 ± 1.83	+3.76 ± 0.53
A0	93.62 ± 1.69	94.83 ± 0.45	91.19 ± 0.36	+2.43 ± 1.92	+3.64 ± 0.48	+3.67 ± 2.77	+5.97 ± 0.56	+2.74 ± 2.11	+4.30 ± 0.47
A3	94.99 ± 0.11	95.48 ± 0.30	90.92 ± 0.23	+4.07 ± 0.29	+4.56 ± 0.07	+6.16 ± 0.34	+7.31 ± 0.21	+4.59 ± 0.29	+5.31 ± 0.12
A3	93.76 ± 0.04	94.31 ± 0.16	90.92 ± 0.26	+2.84 ± 0.27	+3.39 ± 0.18	+4.09 ± 0.39	+5.44 ± 0.31	+3.12 ± 0.30	+3.95 ± 0.22
B0	88.20 ± 0.53	88.95 ± 0.52	88.14 ± 0.52	+0.06 ± 0.49	+0.80 ± 0.05	-1.48 ± 0.82	+1.27 ± 0.07	-0.58 ± 0.57	+0.93 ± 0.05
B0	90.57 ± 0.36	91.51 ± 0.16	90.07 ± 0.14	+0.50 ± 0.33	+1.44 ± 0.07	+0.32 ± 0.54	+2.42 ± 0.10	+0.38 ± 0.39	+1.72 ± 0.08
B2	91.24 ± 0.38	92.28 ± 0.02	90.38 ± 0.05	+0.86 ± 0.41	+1.90 ± 0.03	+1.02 ± 0.68	+3.20 ± 0.03	+0.85 ± 0.49	+2.27 ± 0.03
B2	89.68 ± 0.40	90.54 ± 0.10	89.74 ± 0.06	-0.06 ± 0.45	+0.80 ± 0.13	-0.85 ± 0.77	+1.27 ± 0.25	-0.38 ± 0.54	+0.93 ± 0.17

3.6. Contextual Comparison with Reference-Style Setups

To place our pipeline in context with Pratama et al. [30], we examine why direct comparisons with their published scores are not on equal terms. First, our three-stage curation removed 15.90% of the raw corpus and used a 70:15:15 split rather than the 70:20:10 split in [30]. Second, several training hyperparameters in [30] (such as scheduler warm-up, weight decay, and dropout) were not fully specified. Accordingly, we executed two reference-style configurations (C0 and C1, Batch C) on our curated split across three seeds, as summarized in Table 8.

At $\theta = 0.50$, C0 (bert-base-uncased with Sastrawi stopword removal) achieved 82.37 ± 0.17% Macro-F1 and 91.70 ± 0.14% AUC. This Macro-F1 score lies near the published figure in [30] (80.17%), although the underlying datasets and test splits differ. However, positive recall in [30] was 58.33% (630/1,080, computed from the confusion matrix in [30]), whereas C0 achieved 78.29%, indicating different operating points. Replacing the English backbone with IndoBERT-Base under stopword removal (C1) increased Macro-

F1 by +1.82 pp to 84.19 ± 0.13% ($p < 0.001$). Comparing C1 against unweighted A0 (Batch B) produced the largest stepwise increase of +4.27 pp in Macro-F1 (to 88.46 ± 0.22%, $p < 0.001$). This transition from C1 to unweighted A0 also raised AUC by +3.63 pp (to 96.70%, $p < 0.001$) and positive recall by +9.91 pp. Batch C uses our curated split, so these runs do not isolate the effect of curation. The largest step (C1 to A0) combined stopword removal, the learning-rate scheme, and label smoothing, whereas class weighting and head modification contributed little. The gap to the scores in [30] therefore cannot be attributed to the head, and its sources cannot be separated here.

Table 8. Stepwise contextual comparison on the identical curated test split ($N_{\text{test}} = 5{,}047$, $\theta = 0.50$, mean ± SD, n = 3 seeds)

Code	Backbone	Preprocessing & Training Setup	Acc (%) ($\theta = 0.50$)	Prec_1 (%) ($\theta = 0.50$)	Rec_1 (%) ($\theta = 0.50$)	Macro-F1 (%) ($\theta = 0.50$)	ROC-AUC	Stepwise Δ Macro-F1 / Δ AUC
C0	bert-base-uncased	Sastrawi stopword removal; Unweighted CE; Uniform LR 5×10^{-5}; $\epsilon = 0.0$	85.26 ± 0.16	72.40 ± 0.43	78.29 ± 0.38	82.37 ± 0.17	91.70 ± 0.14	Reference-style baseline context
C1	indobert-base-p1	Sastrawi stopword removal; Unweighted CE; Uniform LR 5×10^{-5}; $\epsilon = 0.0$	87.01 ± 0.06	76.71 ± 0.26	78.36 ± 0.73	84.19 ± 0.13	93.07 ± 0.03	+1.82 pp Macro-F1 / +1.37 pp AUC (vs. C0: Backbone)
A0 (Batch B)	indobert-base-p1	Stopwords retained; Unweighted CE; Differential LR; $\epsilon = 0.05$	90.29 ± 0.21	79.89 ± 0.67	88.27 ± 0.24	88.46 ± 0.22	96.70 ± 0.09	+4.27 pp Macro-F1 / +3.63 pp AUC (vs. C1: Pipeline & recipe)
A0 (Batch A)	indobert-base-p1	Stopwords retained; Smoothed weight w_c'; Differential LR; $\epsilon = 0.05$	89.74 ± 0.29	78.01 ± 1.79	89.42 ± 2.19	87.95 ± 0.18	96.66 ± 0.19	-0.51 pp Macro-F1 / -0.04 pp AUC (vs. A0 Batch B: Weighting)
A2 (Batch A)	indobert-base-p1	+BiLSTM +Attention head; Smoothed weight w_c'; Differential LR; $\epsilon = 0.05$	90.32 ± 0.31	79.01 ± 1.12	90.09 ± 0.79	88.60 ± 0.29	96.79 ± 0.13	+0.65 pp Macro-F1 / +0.13 pp AUC (vs. A0 Batch A: Head)

3.7. Error Analysis

Across three seeds on the test set ($N_{\text{test}} = 5{,}047$), false-positive (FP) and false-negative (FN) counts reflected threshold and operating-point shifts. At $\theta = 0.50$, C0 and C1 recorded 313.3 ± 5.5 and 312.3 ± 10.6 FN, alongside 430.7 ± 10.4 and 343.3 ± 8.0 FP. In contrast, weighted IndoBERT-Base configurations (A0–A3) at $\theta = 0.50$ averaged 143.0 to 170.7 FN and 317.3 to 363.7 FP. After validation threshold tuning, mean FP on A0–A3 decreased to 239.3–287.6, while mean FN increased to 191.6–211.4. On IndoBERT-Lite at $\theta = 0.50$, B0 averaged 463.8 FP and 192.9 FN (86.63% positive recall). Adding recurrent heads on B1 and B2 reduced mean FP to 340.4 and 342.3, while increasing mean FN to 209.7 and 209.4. Thus, on IndoBERT-Lite, the recurrent heads traded a modest increase in missed positives ($\sim$ 16.6 FN) for a larger reduction in false alarms ($\sim$ 122.5 FP).

In a small manual inspection of 20 misclassified test tweets, false positives included hyperbolic situational complaints containing personal pronouns and negative words. A synthetic example of a false positive is: tugas kuliah menumpuk rasanya aku capek sekali ingin menyerah saja. Conversely, inspected false negatives included indirect or metaphorical expressions that omit explicit distress terms. A synthetic example of a false negative is: semuanya terasa hening di kamar ini dan saya sudah tidak berharap apa pun lagi.

4. DISCUSSION

Regarding RQ1, the effect of recurrent heads depended on the backbone. On IndoBERT-Lite, BiLSTM-based heads improved both AUC and threshold-tuned Macro-F1 across seeds, and the gain persisted without class weighting. On IndoBERT-Base, the same heads gave small and inconclusive differences. This contrast may help explain why earlier studies report gains from BiLSTM-attention heads [22] or hybrid heads with auxiliary features [23], [24], [25] , although those studies used other encoders and datasets. The two backbones also differ in more than size. IndoBERT-Lite shares parameters across layers [20] and starts from a lower baseline, which leaves more room for improvement. We therefore cannot attribute the contrast to capacity alone.

Regarding RQ2, attention showed no isolable benefit. BiLSTM and BiLSTM-attention heads performed similarly on both backbones, so the gains are associated with the BiLSTM head. This head also replaces [CLS] pooling with sequence-level pooling, and we did not separate the two changes. Attention weights were not analyzed, so we offer no evidence about which tokens the heads emphasize. The extended head added no benefit over BiLSTM-attention and tended to score lower on IndoBERT-Lite. Over-regularization of the shared-parameter encoder is one possible explanation, but we did not test it. The head also bundles layer normalization [31], residual projection, and multi-sample dropout [32], so their effects cannot be separated.

Regarding RQ3, class weighting shifted the default operating point toward recall. It did not measurably change AUC or threshold-tuned Macro-F1 in the four configurations tested. Threshold tuning appears to absorb most of the effect of weighting on this mildly imbalanced corpus, although only four

configurations were examined. Both class weighting and threshold moving are established responses to imbalance [33], [34].

In the reference-style runs, the largest difference arose when stopword removal, the learning-rate scheme, and label smoothing changed together. Preprocessing can affect Transformer performance [9], [15], but these factors are confounded here, so the result does not show that stopword removal alone caused the drop. The gap to the scores reported in [30] likewise cannot be attributed to the classification head, because our curation, split ratio, and training recipe differ from theirs.

Taken together, the evidence supports a modest message. For a compact backbone, a BiLSTM head is worth trying. For a full-capacity backbone, head design adds little once the threshold is tuned. We do not claim that this generalizes beyond this corpus.

Several limitations concern the data and labels. All experiments used one Indonesian dataset and one fixed split (5,047 test tweets), which limits generalization to other corpora or languages. User identifiers were unavailable [36], so the split was made at the tweet level, and cross-split user overlap cannot be excluded. The source labels [30], [36] were generated automatically with TF-IDF scoring and lexical rules, without human annotation or inter-annotator agreement. High AUC (around 96.7% on IndoBERT-Base) may therefore reflect learnability of the labeling rules rather than clinical validity. The labeling cue list is not reported in [30]. Although [30] describes stopword removal before labeling, the released text retains stopwords, so the exact text seen by the labeling rules cannot be verified.

Other limitations concern the design and statistics. Only two backbones were compared, which prevents conclusions about model scale. All p-values come from three seeds with uncorrected two-sided Welch's t-tests [43] and are strictly indicative. Because the split was fixed, seed variance does not capture split variance. Shared hyperparameters were tuned once on validation data with seed 42, which was also one of the three evaluation seeds. Validation data were also used to select the checkpoint and θ_{opt}, so validation gaps are slightly optimistic. θ_{opt} varied widely across seeds, which inflates variance in tuned precision and recall. Class weighting was tested on only four configurations. The error analysis covered 20 tweets without human re-annotation, and because the labels are rule-based, some errors may reflect label noise rather than model error. Finally, the reference-style runs used our curated split rather than replicating [30], and C1 versus unweighted A0 confounds stopword removal with learning-rate and label-smoothing differences.

5. CONCLUSION

This three-seed ablation examined whether progressive classification heads and class weighting improve IndoBERT models for mental health help indication in Indonesian tweets. First, head effects depended on the backbone. BiLSTM-based heads improved IndoBERT-Lite by about 1.5 pp in threshold-tuned Macro-F1, whereas differences on IndoBERT-Base were small and inconclusive once the threshold was tuned. Second, additive attention gave no isolable gain beyond the BiLSTM head, which also replaces first-token pooling with sequence-level pooling. The extended head neither improved classification nor reduced the dropout-free training–validation gap. Third, square-root class weighting moved default operating points toward recall but did not measurably change AUC or tuned Macro-F1, so threshold tuning appears sufficient for this mildly imbalanced corpus. Fourth, comparisons with reference-style setups serve as context only, because stopword removal and the training recipe changed together. Within this corpus, these results suggest adding a BiLSTM head when fine-tuning a compact backbone such as IndoBERT-Lite, and tuning the decision threshold on validation data. Our evidence does not support attention or bundled stabilization modules. Future work should isolate stopword removal under identical recipes, separate the BiLSTM from sequence-level pooling, test domain-specific backbones such as IndoBERTweet across more seeds, and use human-annotated corpora.

ACKNOWLEDGEMENTS

The authors thank the Department of Informatics Engineering, Faculty of Engineering, Universitas Sam Ratulangi, for supporting this research and providing the necessary computational facilities. The authors also acknowledge Pratama from Telkom University and the IndoNLU team for making the source dataset and pre-trained IndoBERT models publicly accessible for academic research.

REFERENCES

[1] World Health Organization, "Depressive disorder (depression)." Accessed: Nov. 08, 2026. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/depression

[2] World Health Organization, "World Mental Health Report: Transforming Mental Health for All," Geneva, Switzerland, 2022. Accessed: Nov. 08, 2026. [Online]. Available: https://www.who.int/publications/i/item/9789240049338

[3] Center for Reproductive Health, University of Queensland, and Johns Hopkins Bloomberg School of Public Health, "Indonesia – National Adolescent Mental Health Survey (I-NAMHS): Laporan

Penelitian," Yogyakarta, Indonesia, 2022. Accessed: Nov. 08, 2026. [Online]. Available: https://qcmhr.org/outputs/reports/12-i-namhs-report-bahasa-indonesia

[4] DataReportal, "Digital 2024: Indonesia." Accessed: Nov. 08, 2026. [Online]. Available: https://datareportal.com/reports/digital-2024-indonesia

[5] M. N. Asari and T. Mukhoyyaroh, "The impact of loneliness and anonymity on self-disclosure among social media X users," *Psikologia: Jurnal Pemikiran dan Penelitian Psikologi*, vol. 19, no. 1, pp. 32–41, May 2024, doi: 10.32734/psikologia.v19i1.15271.

[6] J. T. V. Harefa, K. A. Manik, Z. L. Habehan, Juniarta, and S. L. Cicilia, "Stress, anxiety, depression and social media in generation Z: A scoping review," *JOURNAL OF Mental Health Concerns*, vol. 4, no. 3, pp. 174–187, Aug. 2025, doi: 10.56922/mhc.v4i3.1362.

[7] R. Salas-Zárate, G. Alor-Hernández, M. A. Paredes-Valverde, M. del P. Salas-Zárate, M. Bustos-López, and J. L. Sánchez-Cervantes, "Mental-Health: An NLP-Based System for Detecting Depression Levels through User Comments on Twitter (X)," *Mathematics*, vol. 12, no. 13, Jul. 2024, doi: 10.3390/math12131926.

[8] M. Y. Dhinora and E. Mailoa, "Analisa Tweet Mahasiswa untuk Deteksi Gejala Depresi dengan Penerapan Natural Language Processing," *Jurnal Indonesia: Manajemen Informatika dan Komunikasi (JIMIK)*, vol. 6, no. 2, pp. 1193–1211, 2025, doi: 10.63447/jimik.v6i2.1405.

[9] U. Khairani, V. Mutiawani, and H. Ahmadian, "Pengaruh Tahapan Preprocessing Terhadap Model Indobert Dan Indobertweet Untuk Mendeteksi Emosi Pada Komentar Akun Berita Instagram," *Jurnal Teknologi Informasi dan Ilmu Komputer*, vol. 11, no. 4, pp. 887–894, Aug. 2024, doi: 10.25126/jtiik.1148315.

[10] G. Coppersmith, M. Dredze, and C. Harman, "Quantifying Mental Health Signals in Twitter," in *Proceedings of the workshop on computational linguistics and clinical psychology: From linguistic signal to clinical reality*, 2014, pp. 51–60.

[11] S. Bouktif, A. M. U. D. Khanday, and A. Ouni, "Explainable Predictive Model for Suicidal Ideation During COVID-19: Social Media Discourse Study," *J. Med. Internet Res.*, vol. 27, 2025, doi: 10.2196/65434.

[12] F. Darmawan, M. Joe, Y. I. Kurniawan, and L. Afuan, "Analisis Sentimen Kemungkinan Depresi dan Kecemasan pada Twitter Menggunakan Support Vector Machine," *Jurnal Eksplora Informatika*, vol. 13, no. 1, pp. 24–36, Sep. 2023, doi: 10.30864/eksplora.v13i1.854.

[13] M. Fachriza and Munawar, "Analisis Sentimen Kalimat Depresi Pada Pengguna Twitter Dengan Naive Bayes, Support Vector Machine, Random Forest," *KOMPUTEK: Jurnal Teknik Universitas Muhammadiyah Ponorogo*, vol. 7, no. 2, pp. 49–58, 2023, doi: 10.24269/jkt.v7i2.2218.

[14] E. E. Pratiwi, A. R. Aisy, Rahmaddeni, and N. Ananta, "Klasifikasi Kesehatan Mental Pada Usia Remaja Menggunakan Metode SVM," *Jurnal Informatika dan Teknik Elektro Terapan (JITET)*, vol. 13, no. 2, pp. 445–453, 2025, doi: 10.23960/jitet.v13i2.6232.

[15] M. Siino, I. Tinnirello, and M. La Cascia, "Is text preprocessing still worth the time? A comparative survey on the influence of popular preprocessing methods on Transformers and traditional classifiers," *Inf. Syst.*, vol. 121, Mar. 2024, doi: 10.1016/j.is.2023.102342.

[16] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Pre-training of Deep Bidirectional Transformers for Language Understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technology.*, 2019, pp. 4171–4186. doi: 10.18653/v1/N19-1423.

[17] B. Wilie *et al.*, "IndoNLU: Benchmark and Resources for Evaluating Indonesian Natural Language Understanding," in *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, Association for Computational Linguistics, 2020, pp. 843–857.

[18] F. Koto, A. Rahimi, J. H. Lau, and T. Baldwin, "IndoLEM and IndoBERT: A Benchmark Dataset and Pre-trained Language Model for Indonesian NLP," in *Proceedings of the 28th International Conference on Computational Linguistics*, International Committee on Computational Linguistics, 2020, pp. 757–770. doi: 10.18653/v1/2020.coling-main.66.

[19] F. Koto, J. H. Lau, and T. Baldwin, "INDOBERTWEET: A Pretrained Language Model for Indonesian Twitter with Effective Domain-Specific Vocabulary Initialization," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, Eds., Association for Computational Linguistics, Nov. 2021, pp. 10660–10668. doi: 10.18653/v1/2021.emnlp-main.833.

[20] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, "ALBERT: A Lite BERT for Self-supervised Learning of Language Representations," in *Proceedings of the 8th International*

Conference on Learning Representations (ICLR), Feb. 2020. [Online]. Available: http://arxiv.org/abs/1909.11942

[21] S. Ji, T. Zhang, L. Ansari, J. Fu, P. Tiwari, and E. Cambria, "MentalBERT: Publicly Available Pretrained Language Models for Mental Healthcare," in *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, Marseille, France: European Language Resources Association, Jun. 2022, pp. 7184–7190.

[22] J. P. Thekkekara and S. Yongchareon, "An Attention-Based BERT–CNN–BiLSTM Model for Depression Detection from Emojis in Social Media Text," *Big Data and Cognitive Computing*, vol. 9, no. 12, Dec. 2025, doi: 10.3390/bdcc9120310.

[23] F. S. Alhussen, "Linguistic Feature-based Depression Prediction Using Hierarchical Transformer-attentive Model for Mental Disabilities Diagnosis," *Journal of Disability Research*, vol. 4, no. 5, Oct. 2025, doi: 10.57197/JDR-2025-0578.

[24] M. Kerasiotis, L. Ilias, and D. Askounis, "Depression detection in social media posts using transformer-based models and auxiliary features," Sep. 2024, [Online]. Available: http://arxiv.org/abs/2409.20048

[25] S. Yeasmin, N. Afrin, K. Saif, and M. Huq, "Development of a Text Classification Framework using Transformer-based Embeddings," International Conference on Data Technologies and Applications, Jul. 2022, pp. 74–82. doi: 10.5220/0011268000003269.

[26] I. R. Hidayat and W. Maharani, "General Depression Detection Analysis Using IndoBERT Method," *International Journal on Information and Communication Technology (IJoICT)*, vol. 8, no. 1, pp. 41–51, 2022, doi: 10.21108/ijoict.v8i1.634.

[27] F. Apriliani and W. Maharani, "DEPRESSION DETECTION ON SOCIAL MEDIA TWITTER USING XLNET METHOD," *JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)*, vol. 8, no. 1, pp. 172–180, Feb. 2023, doi: 10.29100/jipi.v8i1.3345.

[28] G. F. Situmorang and R. Purba, "Deteksi Potensi Depresi dari Unggahan Media Sosial X Menggunakan IndoBERT," *Building of Informatics, Technology and Science (BITS)*, vol. 6, no. 2, pp. 649–661, Sep. 2024, doi: 10.47065/bits.v6i2.5496.

[29] M. S. M. Faza and Moch. Taufik, "Penerapan Model IndoBERT untuk Deteksi Potensi Sumber Stres dalam Teks Media Sosial," *Jurnal Rekayasa Sistem Informasi dan Teknologi (JRSIT)*, vol. 3, no. 1, pp. 271–283, 2025.

[30] F. Pratama, R. Adhitama, and M. Afrad, "The Effect of Hyperparameter Settings on BERT-Based Detection of Mental Health Support Needs on X (Twitter)," in *International Conference of Information and Communication Technology*, 2026. doi: 10.1109/ICoICT71118.2026.11664807.

[31] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer Normalization," Jul. 2016, [Online]. Available: http://arxiv.org/abs/1607.06450

[32] H. Inoue, "Multi-Sample Dropout for Accelerated Training and Better Generalization," 2019, Accessed: Nov. 09, 2026. [Online]. Available: https://arxiv.org/abs/1905.09788

[33] J. M. Johnson and T. M. Khoshgoftaar, "Survey on deep learning with class imbalance," *J. Big Data*, vol. 6, no. 1, Dec. 2019, doi: 10.1186/s40537-019-0192-5.

[34] J. L. Leevy, J. M. Johnson, J. Hancock, and T. M. Khoshgoftaar, "Threshold optimization and random undersampling for imbalanced credit card data," *J. Big Data*, vol. 10, no. 1, Dec. 2023, doi: 10.1186/s40537-023-00738-z.

[35] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, "Bag of Tricks for Efficient Text Classification," in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2*, Valencia, Spain: Association for Computational Linguistics, Apr. 2017, pp. 427–431.

[36] F. Pratama, "Mental Health Help Indication Dataset (Tweet X) [Dataset]," Kaggle. Accessed: Nov. 09, 2026. [Online]. Available: [https://www.kaggle.com/datasets/fauziadityapratama/mental-health-help-indication-dataset-tweet-x]

[37] Google Developers, "Classification: Imbalanced Data," Machine Learning Crash Course. Accessed: Nov. 09, 2026. [Online]. Available: [https://developers.google.com/machine-learning/crash-course/overfitting/imbalanced-datasets]

[38] D. Bahdanau, K. Cho, and Y. Bengio, "Neural Machine Translation by Jointly Learning to Align and Translate," in *International Conference on Learning Representations*, May 2016.

[39] I. Loshchilov and F. Hutter, "Decoupled Weight Decay Regularization," in *International Conference on Learning Representations*, Jan. 2019.

[40] R. Müller, S. Kornblith, and G. E. Hinton, "When Does Label Smoothing Help?," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2019, pp. 4694–4703.

[41] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognit. Lett.*, vol. 27, no. 8, pp. 861–874, Jun. 2006, doi: 10.1016/j.patrec.2005.10.010.

[42] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Inf. Process. Manag.*, vol. 45, no. 4, pp. 427–437, Jul. 2009, doi: 10.1016/j.ipm.2009.03.002.
[43] B. L. Welch, "The generalization of 'Student's' problem when several different population variances are involved," *Biometrika*, vol. 34, pp. 28–35, Jan. 1947, doi: 10.1093/biomet/34.1-2.28.