TELKOMNIKA Telecommunication Computing Electronics and Control
Vol. 24, No. 2, April 2026, pp. 549~563
ISSN: 1693-6930, DOI: 10.12928/TELKOMNIKA.v24i2.27332

Improving multilabel classification of hate speech and abusive language in Indonesian using MAML

Jasman Pardede, Ghixandra Julyaneu Irawadi, Rizka Milandga Milenio
Department of Informatics, Faculty of Industrial Technology, Institut Teknologi Nasional Bandung, Bandung, Indonesia

Article Info

Article history:

Received Jun 30, 2025
Revised Dec 17, 2025
Accepted Jan 30, 2026

Keywords:

Hate speech detection
IndoBERTweet-BiGRU
Meta-learning
Model-agnostic meta-learning
Multilabel classification

ABSTRACT

This study investigates automated multi-label detection of hate speech and abusive language (HSAL) in Indonesian social media, addressing challenges of data imbalance, especially in minority labels. Two training approaches are compared: standard supervised learning and meta-learning using the model-agnostic meta-learning (MAML) algorithm. IndoBERTweet-BiGRU is adopted as the baseline model, while MAML is leveraged to enhance generalization and adaptability with limited training data. Both models are trained on a multilabel dataset with 13 HSAL categories exhibiting highly imbalanced distributions. The best supervised model achieved an F1-Micro of 84.02% and an F1-macro of 77.97%, whereas the best MAML-trained model reached 84.12% and 76.85%, respectively. Although the overall gap is small, MAML demonstrates notable improvements on minority classes such as hate speech (HS) physical, gender, and race, shown through higher F1-score and area under the receiver operating characteristic curve (AUROC) values. These results highlight its strength in low-resource classification settings. This study is limited to Indonesian language and YouTube transcript contexts, and MAML incurs higher training complexity. Cultural and linguistic nuances also present potential bias in real-world use. Despite these constraints, the proposed system offers practical benefits by enabling fine-grained HSAL classification and supporting earlier detection of harmful online content.

Corresponding Author:

Jasman Pardede
Department of Informatics, Faculty of Industrial Technology, Institut Teknologi Nasional Bandung
Jl. K.H.P. Hasan Mustopa No. 23, Neglasari, Cibeunying Kaler, Bandung 40124, West Java, Indonesia
Email: jasman@itenas.ac.id

1. INTRODUCTION

The rapid dissemination of harmful and aggressive content on the internet, including hate speech (HS) and abusive behavior, poses significant risks to online communities, particularly in digital environments that lack emotional regulation and are prone to miscommunication [1]-[3]. To support safer interactions, automated HS and abusive language (HSAL) detection systems are urgently required. This study aims to improve multi-label HSAL detection in Indonesian online media by addressing practical challenges in real-world deployment, especially under data imbalance and low-resource conditions.

Previous studies have applied machine learning and deep neural models, including support vector machines (SVM), convolutional neural networks (CNN), long short-term memory (LSTM), bidirectional encoder representations from transformers (BERT), and RoBERTa, to HSAL detection [4], [5]. IndoBERTweet, trained on informal Indonesian social media text, effectively handles code-switching and informal vocabulary [6]-[10], yet most research focuses only on Twitter. Conversely, HSAL detection in video-based environments, especially for Indonesian-language content, remains significantly underexplored [5].

Existing Indonesian HSAL datasets also exhibit extreme class imbalance, especially in multilabel scenarios where multiple harmful categories may co-occur [11]-[14]. Moreover, current deep learning models struggle with transcript-style inputs, computational efficiency, and generalization in low-resource settings [12].

To address these limitations, this study proposes an adaptive HSAL detection approach integrating IndoBERTweet and bidirectional gated recurrent unit (BiGRU) with the model-agnostic meta-learning (MAML) algorithm, enabling improved generalization and faster adaptation to minority classes. To the best of our knowledge, this is the first study applying a meta-learning framework for fine-grained multi-label HSAL detection on Indonesian YouTube transcripts, using a model pre-trained on social media text data. The proposed system is designed as an early-stage moderation tool to assist proactive monitoring rather than fully automated content removal, supporting the practical need to reduce exposure to harmful online material in Indonesian digital platforms. This study demonstrates the potential of meta-learning to enhance HSAL detection robustness in real deployment settings with limited data availability.

2. METHOD

2.1. Dataset

This study uses an Indonesian multilabel HSAL dataset released by Darmawan *et al.* [15], containing 13,169 tweets. The dataset was developed in collaboration with the Cyber Crime Directorate of Bareskrim Polri through a focus group discussion (FGD) with relevant stakeholders, resulting in 12 HSAL labels and one non-harmful label. The HSAL labels include: HS, Abusive, HS_Individual, HS_Group, HS_Religion, HS_Race, HS_Physical, HS_Gender, HS_Other, HS_Weak, HS_Moderate, and HS_Strong. The additional label PS (Positive/Neutral) marks tweets without harmful content. Each label represents a different target or severity level of offensive language, allowing for fine-grained classification. The label distribution is shown in Figure 1.

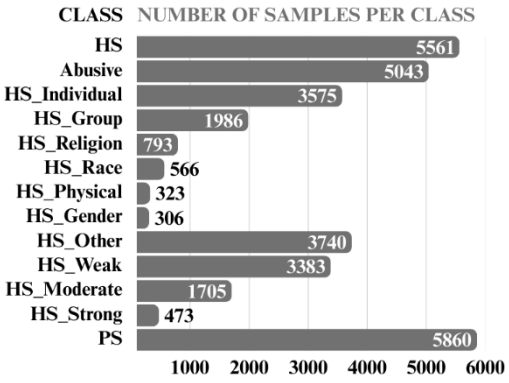


Figure 1. Dataset distribution

These labels categorize tweets based on the type and intensity of HS or abusive language. However, the label distribution is highly imbalanced, as shown in Figure 1. For instance, 5,561 tweets are labeled as HS, while only 306 tweets fall under HS_Gender. Other minority classes include HS_Strong, HS_Moderate, HS_Physical, HS_Race, HS_Religion, and HS_Group.

Class imbalance in multi-label classification is commonly addressed using techniques such as random oversampling, under sampling, cost-sensitive learning, focal loss, or class-balanced loss. In this study, synthetic minority oversampling technique (SMOTE) ($k = 3$) was selected because it generates synthetic minority samples without duplicating instances, provides more stable minority coverage, and does not require modifications to the loss function or model architecture, an important consideration for maintaining stable MAML inner-loop optimization. Preliminary experiments showed that SMOTE yielded more consistent improvements than focal loss and class-balanced loss, which tended to produce unstable gradients when combined with meta-learning. Cost-sensitive learning also showed sensitivity to weight scaling under highly skewed label distributions, resulting in fluctuating updates during meta-training. These observations indicate that SMOTE offers a more reliable imbalance-handling strategy for both conventional and MAML-based training in this setting.

2.1.1. Preprocessing dataset

A preprocessing pipeline was applied before model training. It consists of five stages: case folding, filtering, tokenization, conversion to tensor format, and dataset construction. First, all text is converted to lowercase to standardize word representation. Filtering then removes irrelevant characters, such as symbols, numbers, and extra whitespace. Next, tokenization is performed using the IndoBERTweet tokenizer, which applies a WordPiece-based subword approach to handle informal and out-of-vocabulary terms. The tokenizer generates three outputs: input_ids which represent the tokenized text as numerical indices based on the model's vocabulary, attention_mask which indicates meaningful tokens with values of 1 and padding tokens with values of 0, and token_type_ids which indicate whether the input consists of a single sentence (value 0) or a pair of sentences (value 1). Table 1 shows the final structured format of the preprocessed data.

Table 1. Example of preprocessing results

Real text	Preprocessed text	Token ID	Attention mask	Token type ids
41. Sometimes I wonder why I still believe in God even though I keep falling over and over again. Sometimes I feel that God has abandoned me. When my parents planned to separate, when my sibling chose to become a Christian, when I was a child	sometimes i wonder why i still believe in god even though i keep falling over and over again sometimes i feel that god has abandoned me when my parents planned to separate when my sibling chose to become a Christian, when i was a child	[3, 3851, 2254, 10990, 5396, 2254, 2261, …, 0]	[1, 1, 1, 1, 1, 1, 1, …, 0]	[0, 0, 0, 0, 0, 0, 0, …, 0]

After tokenization, the data were converted into tensor format, as required by the model. The dataset was then divided into three subsets: training, validation, and test. The split proportions were 80% for training and 10% each for validation and testing. Table 2 shows the label distribution across these subsets.

Table 2. Label distribution on subset dataset

Label	Amount of data	Amount of data		
		Training	Validation	Testing
HS	5561	4438	543	580
Abusive	5043	4010	502	531
HS_Individual	3575	2872	341	362
HS_Group	1986	1566	202	218
HS_Religion	793	633	74	86
HS_Race	566	454	58	54
HS_Physical	323	276	24	23
HS_Gender	306	250	26	30
HS_Other	3740	2960	374	406
HS_Weak	3383	2714	322	347
HS_Moderate	1705	1350	167	188
HS_Strong	473	374	54	45
PS	5860	4698	592	570
Total amount data	13.169	10535	1317	1317

2.2. Base line model

The primary model in this study combines IndoBERTweet and BiGRU in a collaborative architecture. Initially, IndoBERTweet acts as a feature extractor, converting input text into dense vector representations that capture the contextual meaning of words. Its output is a three-dimensional tensor of shape [batch_size, sequence_length, hidden_size], where each token is represented by a vector encoding semantic and contextual information in high-dimensional space. This representation comes from pretrained BERT and reflects relationships between words in context. The overall architecture of the proposed model is shown in Figure 2.

The tensor is then passed to a BiGRU layer, which captures sequential dependencies in both directions. The bidirectional BiGRU produces a tensor of shape [batch_size, sequence_length, hidden_size × 2], combining information from forward and backward passes. Finally, the output is fed into a fully connected layer that serves as the classification head. It predicts 13 labels per sample, producing 13 logit values. These logits are converted to probabilities between 0 and 1 using a sigmoid activation. Labels with probabilities above 0.5 are considered positive.

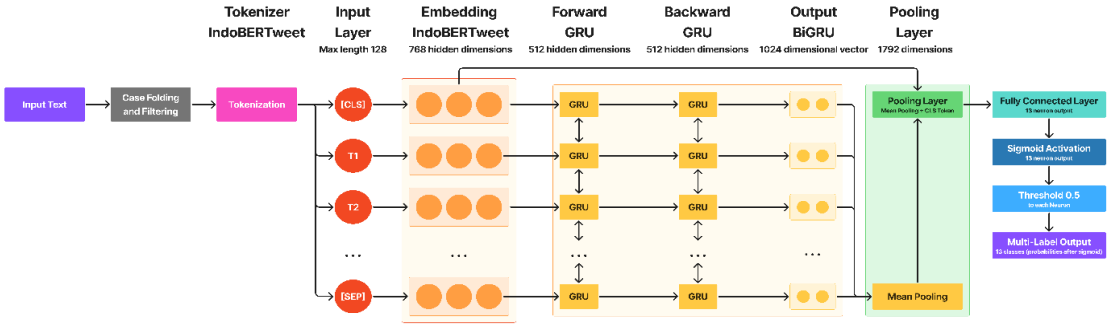


Figure 2. Model architecture

2.2.1. IndoBERTweet

BERT is a deep learning model based on the attention mechanism [16]. It is widely used in natural language processing due to its effectiveness in transfer learning and its ability to model contextual relationships between words [17]. BERT operates within the Transformer encoder framework, comprising 12 layers and 12 attention heads, with 768-dimensional embeddings and hidden states [12], [18]. Tokenization is performed using WordPiece, with additional [CLS] and [SEP] tokens to structure input [19].

IndoBERTweet is a domain-adapted BERT variant, pretrained specifically on Indonesian tweets using approximately 409-490 million tokens, nearly twice that of IndoBERT [6], [9], [15]. Its vocabulary and corpus are optimized for informal social media text, making it more robust in handling linguistic variations than IndoBERT or mBERT [6].

$$NSP = softmax(W \cdot H\_[CLS] \ + b) \tag{1}$$

$$Attention(Q, K, V) = softmax\left(\frac{Q \cdot K^T}{\sqrt{d_k}} \cdot V\right) \tag{2}$$

During pretraining, BERT uses two main objectives: masked language modeling (MLM), which predicts masked tokens to learn contextual word representations, and next sentence prediction (NSP), which evaluates sentence coherence using the [CLS] token and a softmax layer (1) [6]. BERT also employs scaled dot-product self-attention (2) to compute inter-token dependencies via learned query (Q), key (K), and value (V) matrices, enabling rich semantic encoding across sequences [16].

2.2.2. BiGRU

The BiGRU extends the standard gated recurrent unit (GRU) by processing sequential data in both forward and backward directions. This bidirectional design captures contextual information from past and future tokens simultaneously, improving understanding of word relationships within a sentence [19], [20].

$$z_T = \sigma(W_z i_T + U_z h_{T-1} + b_z) \tag{3}$$

$$r_T = \sigma(W_r i_T + U_r h_{T-1} + b_r) \tag{4}$$

$$h_T^{\sim} = tanh(W_h i_T + U_h(r_T \odot h_{T-1}) + b_h) \tag{5}$$

$$h_T = (1 - z_T) \odot h_{T-1} + z_T \odot h_T^{\sim} \tag{6}$$

$$h_T = Concat(h_T^f, h_T^b) \tag{7}$$

BiGRU simplifies traditional recurrent neural networks (RNNs) by combining the input and forget gates into a single update gate and adding a reset gate to control information flow. This makes it more efficient at handling long sequences [19], [21]. The model receives embeddings from IndoBERTweet and computes hidden states through both forward and backward GRU layers. These layers process the sequence from start to end and vice versa, capturing bidirectional context [20]. The GRU mechanism computes the update gate (3), reset gate (4), candidate hidden state (5), and current hidden state (6) [21]. The final contextual representation is formed by concatenating hidden states from both directions (7), yielding a richer and more comprehensive encoding of the input sequence.

2.3. Training method

2.3.1. Conventional supervised learning training

In conventional training, the model is trained iteratively for a predefined number of epochs, each consisting of a training and validation phase. This study adopts a supervised multi-label classification approach, where human annotations are required [21]. To meet this requirement, label development was conducted through a FGD with the Criminal Investigation Agency of the Indonesian National Police (*Bareskrim Polri*). The objective during training is to minimize loss and improve performance metrics, including F1-score, AUROC, and exact match accuracy.

During training, the model operates in training mode to allow gradient updates. Input data are grouped into mini-batches using a data loader. For each batch, the model generates probability outputs through a sigmoid activation function for every label. Loss is then calculated using binary cross-entropy loss (BCELoss). Backpropagation updates the model by computing gradients from the loss function, followed by parameter updates using the optimizer and gradient resetting with optimizer.zero_grad().

In the validation phase, the model switches to evaluation mode with gradients disabled using torch.no_grad(), reducing memory usage and preventing parameter updates. The predicted probabilities are converted into binary labels through a predefined threshold. Performance is evaluated using micro and macro F1-scores, exact match accuracy, and micro-macro-averaged AUROC. Two model checkpoints are stored: the most recent model and the best-performing model based on macro F1-score. A learning-rate scheduler adjusts the training dynamics, and tqdm is used to display progress during the training process.

2.3.2. MAML

MAML is a meta-learning framework designed to help models quickly adapt to new tasks using minimal training data [22], [23]. Being architecture-agnostic, MAML can be applied to any gradient-based model for tasks such as classification, regression, or reinforcement learning [10], [24]. The flowchart of the MAML process is illustrated in Figure 3.

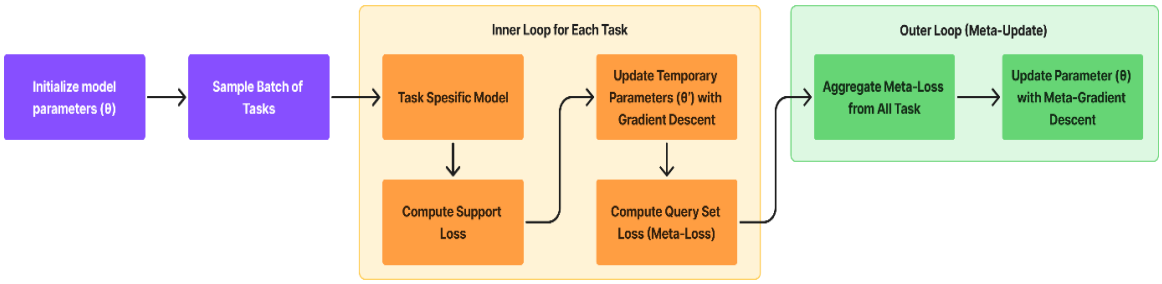


Figure 3. MAML flowchart

The core idea of MAML is to learn an optimal initialization of model parameters. With this initialization, only a few gradient steps are needed to achieve high performance on unseen tasks [22]. Its training consists of two loops: the inner loop updates parameters using the support set of a given task (8), while the outer loop optimizes meta-parameters by minimizing the meta-loss across tasks, as formalized in (9) [22], [24].

$$\theta_i' \rightarrow \theta - \beta \nabla L_{T_i}(f_\theta) \tag{8}$$

$$\theta \leftarrow \theta - \beta \nabla_0 \sum_{T_i \sim p(T)} L_{T_i}(f_{\theta_i'}) \tag{9}$$

By repeatedly adapting to multiple tasks, the outer loop enhances generalization. As a result, MAML-trained models perform efficiently and effectively, especially in low-resource scenarios.

2.3.3. Evaluation metrics

This study employs three primary evaluation metrics, namely F1-score, AUROC, and exact match accuracy. The F1-score is particularly important for imbalanced datasets as it balances precision and recall, and is reported in both micro and macro forms. Micro F1 aggregates true/false positives and negatives across all labels, while macro F1 averages the scores per class, assigning equal weight to each label. AUROC measures the model's ability to distinguish between positive and negative labels without relying on a fixed

threshold, by analyzing sensitivity true positive rate (TPR)) and specificity (false positive rate (FPR)) [12], which is crucial for imbalanced multi-label tasks. Exact Match Accuracy measures whether the predicted label set fully matches the ground truth [25], making it a strict indicator of predictive performance. All hyperparameter combinations were evaluated on the held-out test set using these metrics to ensure comprehensive assessment. Table 3 summarizes the baseline model performance comparison. Furthermore, Table 4 presents the results of the three best-performing models from each training method, including these metrics and the lowest training and validation losses.

3. RESULTS AND DISCUSSION

3.1. Model creation

The experiments were carried out in two phases. The first optimized the IndoBERTweet-BiGRU baseline via conventional supervised learning, evaluating 185 configurations that varied in learning rates, optimizers, loss functions, pooling strategies, batch sizes, and hidden sizes. This step established a stable baseline for comparison with meta-learning.

The second phase applied the same architecture using MAML, testing variations in outer and inner learning rates, number of inner steps, and meta-batch sizes to achieve optimal few-shot adaptation. Additional hidden-size and batch-size settings were also explored. The three top models per method were then compared. All experiments were conducted on Kaggle's Tesla T4 graphics processing unit (GPU) for consistent evaluation.

3.1.1. Metric evaluation results on testing data

To justify the choice of IndoBERTweet-BiGRU as the core architecture, baseline comparisons were performed using BiGRU alone, IndoBERTweet alone, and their combination, trained under both conventional and MAML settings. The combined model produced the most balanced performance, supporting its use in subsequent evaluations of meta-learning. The results of these experiments are presented in Table 3. This experiment was conducted using the configuration presented in Table 4, referred to as Conventional_1.

Table 3. Comparison of baseline model performance

Model	Best loss		Exact match	AUROC		F1-score	
	Train	Val		Micro	Macro	Micro	Macro
BiGRU	0.1557	0.3255	0.4973	0.8816	0.8464	0.6565	0.4059
IndoBERTweet	0.0021	0.1712	0.7517	0.9612	0.946	0.8339	0.7618
IndoBERTweet+BiGRU	**0.0015**	**0.1639**	**0.7570**	**0.9641**	**0.9526**	**0.8402**	**0.7797**
BiGRU + MAML	0.2681	0.3154	0.3675	0.8936	0.8301	0.5869	0.3031
IndoBERTweet + MAML	0.0285	0.1739	**0.7449**	0.9689	0.9546	0.8358	0.7623
IndoBERTweet+BiGRU + MAML	0.0237	0.1766	0.7441	**0.9694**	**0.9552**	**0.8398**	**0.7701**

Based on the results presented in Table 3, the IndoBERTweet-BiGRU model was selected for further hyperparameter tuning under both conventional and MAML setups. The corresponding hyperparameter configurations of the top three models for each training method are summarized in Table 4, all trained for 30 epochs.

Table 4. Hyperparameter configuration

Model	Scheduler	Optimizer	Loss function	Seed	Learning rate		Inner step	Meta batch	Hidden size	Batch size
					Outer	Inner				
Conventional_1	CosineAnnealingLR	AdamW	BCELoss	99	4e-5	-	-	-	256	16
MAML_1	CosineAnnealingLR	AdamW	BCELoss	99	4e-5	2e-3	9	4	256	16
Conventional_2	CosineAnnealingLR	AdamW	BCELoss	99	4e-5	-	-	-	512	16
MAML_2	CosineAnnealingLR	AdamW	BCELoss	99	4e-5	1e-4	10	4	512	16
Conventional_3	CosineAnnealingLR	AdamW	BCELoss	99	4e-5	-	-	-	512	32
MAML_3	CosineAnnealingLR	AdamW	BCELoss	99	4e-5	5e-4	5	4	512	32
Conventional_4	CosineAnnealingLR	Adam	BCELoss	99	4e-5	-	-	-	256	16
MAML_4	CosineAnnealingLR	Adam	BCELoss	99	4e-5	2e-3	9	4	256	16
Conventional_5	CosineAnnealingLR	Adam	BCELoss	99	5e-5	-	-	-	512	32
MAML_5	CosineAnnealingLR	Adam	BCELoss	99	5e-5	2e-3	3	4	512	32
Conventional_6	CosineAnnealingLR	Adam	BCELoss	99	4e-5	-	-	-	512	32
MAML_6	CosineAnnealingLR	Adam	BCELoss	99	4e-5	5e-4	5	4	512	32

The chosen hyperparameters reflected a balance between adaptation speed and stability. Outer learning rates of 4e-5 and 5e-5, effective in conventional training, were also used in MAML for better generalization. Inner learning rates (2e-3, 5e-4, 1e-4) were tested to evaluate their impact on quick adaptation versus optimization stability.

The study varied inner-loop steps (3, 5, 9, 10) to gauge adaptation depth and used a meta-batch size of 4 to balance diversity and cost. Hidden sizes of 256 and 512 were tested for efficiency versus representational strength, along with batch sizes of 16 and 32 for stable gradient updates. A sequence length of 128 matched the dataset's token distribution. Table 5 reports the performance across configurations.

Table 5. Comparison of model performance in two training methods

Model	Best loss		Exact	AUROC		F1-score		Training	Memory	Model size
	Train	Val	match	Micro	Macro	Micro	Macro	time	usage (MB)	(MB)
Conventional_1	**0.0015**	**0.1639**	**0.7570**	0.9641	0.9526	**0.8402**	**0.7797**	02:05:15	3661	427.829
MAML_1	0.0246	0.1734	0.7418	**0.9687**	**0.9565**	0.833	0.7627	06:05:04	1437.49	427.83
Conventional_2	**0.0013**	**0.1679**	**0.7585**	0.9633	0.9499	**0.8375**	**0.7710**	02:20:22	3715	436.866
MAML_2	0.0294	0.1818	0.7267	**0.9647**	**0.9478**	0.8196	0.7453	07:54:19	1506.12	436.87
Conventional_3	**0.0018**	**0.1650**	**0.7525**	0.9662	0.9540	0.8384	0.7662	02:16:14	4765	436.866
MAML_3	0.0254	0.1749	0.7449	**0.9694**	**0.9552**	**0.8385**	**0.7693**	04:36:20	1541.89	436.87
Conventional_4	**0.0017**	**0.1722**	0.7411	0.9664	0.9573	0.8273	0.7636	02:05:32	3661	427.829
MAML_4	0.0250	0.1773	**0.7464**	**0.9703**	**0.957**	**0.8412**	**0.7685**	06:27:31	1424.87	427.83
Conventional_5	**0.0019**	**0.1621**	**0.7487**	0.9629	0.9526	0.8305	0.7629	02:24:07	4765	427.829
MAML_5	0.0237	0.1766	0.7441	**0.9694**	**0.9552**	**0.8398**	**0.7701**	03:46:48	1538.14	436.87
Conventional_6	**0.0018**	**0.1696**	**0.7525**	0.9650	0.9538	0.8359	**0.7721**	01:55:05	4765	436.866
MAML_6	0.0271	0.1770	0.7472	**0.9698**	**0.9567**	**0.8393**	0.7674	04:40:58	1539.62	436.87

Across six configuration pairs, conventional training showed lower training/validation loss but a larger gap, indicating stronger overfitting. The overfitting comparison showed that MAML exhibited more stable generalization despite higher training loss, which is reinforced by research suggesting that the bi-level optimization framework of MAML can reduce overfitting compared to conventional methods [26]. Performance differences were small, with four configurations favoring MAML and two favoring conventional training. MAML required 2-3.5 x longer training but used only 30-40% of the GPU memory, while inference time differed only by a few milliseconds, making both approaches practically similar for deployment.

MAML hyperparameter sensitivity has been noted in previous studies [27] and was further examined from an optimization perspective here. MAML performance depended heavily on the ratio between inner-loop and outer-loop learning rates. Meta-batch size also played a key role: too small led to unstable gradients, while too large reduced task diversity. Experiments indicated that a meta-batch size of four produced the most stable outcomes and was adopted for all reported settings. The number of inner steps influenced results as well, but its effect was closely tied to learning rate choices. With these settings, per-label performance was evaluated, with detailed results shown in Table 6.

Table 6. Evaluation metric for each label

Label	F1-score		AUROC	
	MAML	Conventional	MAML	Conventional
HS	**0.8904**	0.8776	**0.9599**	0.9504
Abusive	**0.926**	0.9244	**0.9816**	0.9774
HS_Individual	**0.8062**	0.7945	**0.9413**	0.9374
HS_Group	**0.7319**	0.7036	**0.9267**	0.9165
HS_Religion	0.75	**0.7602**	**0.9839**	0.9756
HS_Race	**0.8155**	0.7961	**0.9914**	0.9883
HS_Physical	0.5	**0.5581**	0.966	**0.9775**
HS_Gender	**0.6667**	0.6	0.9215	**0.9468**
HS_Other	**0.834**	0.8263	**0.9466**	0.9371
HS_Weak	**0.7825**	0.7776	**0.9372**	0.9328
HS_Moderate	**0.6648**	0.6232	**0.9136**	0.8996
HS_Strong	0.7312	**0.7708**	0.972	**0.973**
PS	**0.9124**	0.9056	**0.9765**	0.9708

For the fifth model pair, MAML improved F1-scores and AUROC for several minority labels, including HS_Group, HS_Gender, and HS_Moderate. Despite the significant data imbalance, MAML proved stable and effective [28]. MAML increased average F1-Macro by 5.19% and improved minority-label F1 by

up to 1.00%. AUROC rose by an average of 0.28%, confirming that MAML leads to a better generalized model initialization for fast adaptation [29] and generalizes better under data imbalance. Table 7 provides an extended comparison that includes normalization and dropout to further support this finding.

With normalization and dropout, conventional models still achieved higher F1 and Exact Match scores but continued to exhibit overfitting. MAML maintained higher AUROC, indicating better discriminative ability under imbalance, while showing more stable validation performance. The application of regularization strategies within the meta-learning framework [26] confirms a trade-off: conventional models optimize accuracy on dominant classes, whereas MAML yields stronger generalization. Layer normalization and dropout contribute to training stability, but the inherent methodological differences remain.

Table 7. Comparison of model performance in two training methods with normalization and dropout

Model	Best loss		Exact	AUROC		F1-score	
	Train	Val	match	Micro	Macro	Micro	Macro
Conventional_1	**0.0016**	**0.1736**	**0.757**	0.9636	0.9505	**0.839**	0.7699
MAML_1	0.0265	0.1785	0.7418	**0.9708**	**0.9591**	0.8313	**0.7714**
Conventional_2	**0.0013**	**0.1661**	**0.7509**	0.9629	0.9519	0.8337	**0.7655**
MAML_2	0.0214	0.1804	0.7335	**0.9709**	**0.9605**	**0.8363**	0.7574
Conventional_3	**0.0019**	**0.1616**	**0.7426**	0.9629	0.9522	**0.8328**	**0.7653**
MAML_3	0.0240	0.1803	0.7365	**0.9692**	**0.9582**	0.8326	0.7606
Conventional_4	**0.0018**	**0.1693**	**0.7616**	0.965	0.9548	**0.8437**	**0.7682**
MAML_4	0.0242	0.1793	0.7365	**0.9692**	**0.9566**	0.828	0.7544
Conventional_5	0.0015	0.1681	**0.7517**	0.9634	0.952	**0.8404**	**0.7624**
MAML_5	0.0266	0.1794	0.7289	**0.9689**	**0.957**	0.8311	0.7567
Conventional_6	**0.0019**	**0.1661**	**0.7593**	0.9627	0.952	**0.8431**	**0.7818**
MAML_6	0.0246	0.1767	0.7365	**0.9695**	**0.9571**	0.8373	0.7534

To more clearly quantify the observed performance trade-offs, the results show that MAML achieves an average F1-macro improvement of 0.91% across the six model pairs in Table 5, with gains of up to 1.00% on minority labels such as HS_Gender and HS_Moderate Table 6. In terms of AUROC, MAML exceeds conventional training by an average margin of 0.28%, indicating significant adaptation benefits [29] and improved discriminative capability across imbalanced classes. When normalization and dropout are applied Table 7, the conventional models retain a slight advantage in F1-scores, but MAML consistently yields higher AUROC, up to + 0.75% on some configurations, demonstrating stronger generalization to unseen distributions. These quantitative results reinforce that MAML offers a more balanced trade-off between accuracy and robustness, particularly under low-resource and imbalanced conditions. To extend the analysis, an additional data-balancing method, SMOTE with $k = 3$, was applied. The oversampled dataset was trained using both conventional supervised learning and MAML for only two epochs due to runtime limits in Kaggle. These models follow the fifth configuration in Table 4, and their performance is summarized in Table 8.

Table 8. Performance comparison of models on the oversampled dataset using two training methods

Model	Best loss		Exact	AUROC		F1-score		Training	Memory usage	Model size
	Train	Val	match	Micro	Macro	Micro	Macro	time	(MB)	(MB)
Conventional	**0.0027**	**0.0029**	**0.9951**	**1.0**	**0.9999**	**0.9987**	**0.9942**	03:27:16	5403	436.866
MAML	0.0189	0.0214	0.9725	0.9996	0.9987	0.9921	0.9729	05:57:13	1538.98	436.87

On the SMOTE-oversampled dataset (two epochs due to runtime limits), conventional training achieved slightly better F1 and validation loss than MAML. However, the short training duration likely prevented MAML from fully optimizing. Longer training would be needed to fairly assess MAML's performance under oversampling.

To quantitatively compare conventional and MAML-based training, paired Wilcoxon signed-rank tests were conducted using per-sample macro-F1 and exact-match scores for each configuration (Table 9), based on the models summarized in Table 5. The mean per-sample scores are reported along with the corresponding Wilcoxon p-values, with statistically significant results ($p < 0.05$) shown in bold.

The statistical analysis shows that only configuration 2 presents a significant performance difference, with the conventional model outperforming MAML in both macro-F1 and exact-match accuracy. All other configurations exhibit no significant differences, although configuration 4 indicates a slight

tendency in favor of MAML. These findings suggest that MAML's generalization capability is highly sensitive to hyperparameter choices, a challenge recognized in studies on meta-optimization for low-resource scenarios [27], yielding competitive or superior results only under appropriately aligned settings. Despite this variability, the overall comparable macro-F1 scores demonstrate that MAML can reliably match conventional training while offering advantages in rapid adaptation. Furthermore, alternative meta-learning strategies, such as Reptile and ProtoMAML, which simplify inner-loop optimization or leverage metric-based representations [10], [15], may provide enhanced stability in low-resource or imbalanced scenarios. Positioning MAML within this broader meta-learning landscape underscores both its potential and the opportunities for future work to refine HSAL detection under multilingual and imbalanced conditions. Additionally, imbalance-oriented strategies such as class-balanced loss or focal loss could also be explored in combination with meta-learning, offering alternative ways to mitigate minority-label sparsity beyond synthetic oversampling.

Table 9. Statistical comparison between conventional and MAML-based models across six configurations

Configuration	n	F1-macro			Exact match accuracy		
		Conventional	MAML	p	Conventional	MAML	p
1	1317	0.906	0.905	0.543	0.757	0.742	0.107
2	1317	0.906	0.897	0.026	0.759	0.727	0.001
3	1317	0.907	0.907	0.884	0.752	0.745	0.365
4	1317	0.899	0.906	0.061	0.741	0.743	0.863
5	1317	0.903	0.908	0.203	0.738	0.741	0.298
6	1317	0.905	0.907	0.779	0.744	0.746	0.386

3.1.2. Loss and F1-score accuracy graph

To provide a deeper analysis of the findings, graphs are presented to illustrate the dynamics of loss and F1-score (both micro and macro) during training and validation processes. These visualizations are based on the fifth model pair in Tables 5 and 7. Figure 4 shows the detailed progression of these metrics under different training conditions, comparing conventional and MAML-based models with and without normalization and dropout.

Figure 4 compares the training dynamics of the conventional and MAML models, both with and without layer normalization and a 0.2 dropout rate. In the conventional model without regularization Figure 4(a), severe overfitting occurs. Training loss quickly approaches zero, while validation loss continues to rise, and validation F1-scores show minimal progress with occasional declines. MAML without regularization Figure 4(b) exhibits more stable learning behavior. Both training and validation loss decrease, although validation progresses more gradually, and micro and macro validation F1-scores increase consistently from early epochs, reflecting better generalization. When normalization and dropout are added to the conventional model Figure 4(c), overfitting is reduced Validation loss becomes more stable, and validation F1-scores improve slightly; however, a clear gap between training and validation loss remains, indicating limited generalization to minority classes. With regularization applied to MAML Figure 4(d), performance improves further. This configuration produces the smallest train-validation loss gap, while validation F1-scores rise smoothly and plateau at strong levels.

Overall, the conventional model learns rapidly but is highly prone to overfitting. MAML offers a more balanced learning process by optimizing parameters for adaptability across tasks, which is a key benefit of the meta-learning optimization strategy [29]. Regularization further enhances training stability and reduces performance fluctuations. Thus, MAML with normalization and dropout provides the most reliable performance among the tested configurations.

Finally, although MAML improves generalization in low-resource settings, ethical risks must be considered when deploying HSAL detection in real platforms. Indonesian linguistic nuances, such as sarcasm, regional expressions, and dialects, may affect predictions and introduce unintended bias. This concern is particularly relevant for sensitive categories like HS_Gender, HS_Group, and HS_Religion, where misclassification could unfairly impact marginalized communities. Therefore, any real-world deployment should include fairness assessments, user feedback loops, and ongoing monitoring to prevent discriminatory outcomes.

3.2. Web application development

The best-performing model from this comparative study was then used as the basis for developing a web-based application and subsequently tested using transcription data from YouTube videos. The model used at this stage is the fifth model pair from Table 5. The results of this evaluation are presented in Figure 5. Figure 5(a) illustrates the overall analysis output of the proposed web-based content moderation application

after processing a YouTube video transcript. Figure 5(b) presents the detailed sentence-level analysis produced by the classification model.

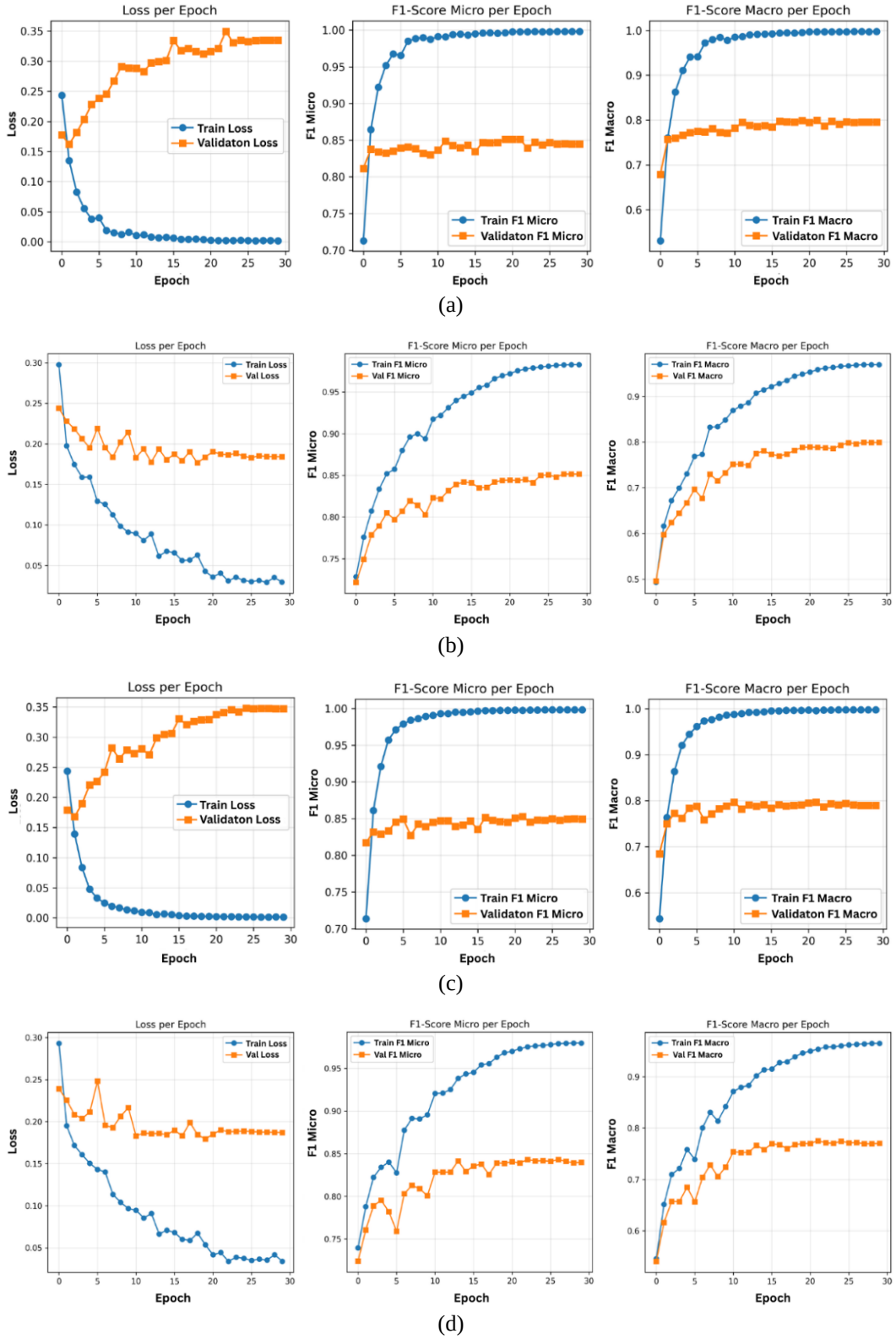
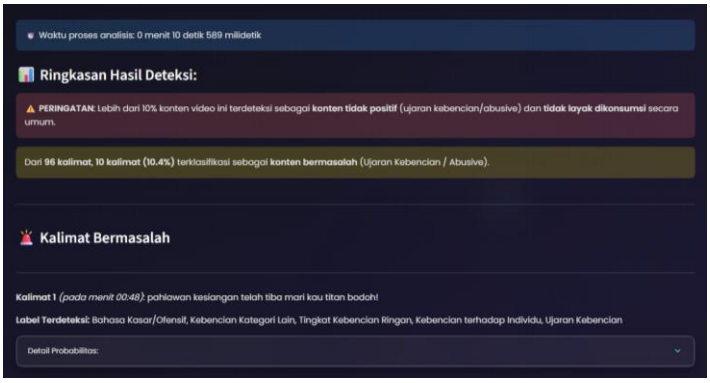


Figure 4. Graphs: (a) conventional model, (b) MAML-based model, (c) conventional model with normalization and dropout, and (d) MAML-based model with normalization and dropout



(a)

(b)

Figure 5. Model implementation results: (a) overall content analysis results and (b) detailed analysis per sentence

Table 10. Detected sentence and predicted class rankings on applications

Time stamp	Detected sentence	Predicted class rankings	
		With MAML	Without MAML
00:48	The late hero has arrived come on, you stupid Titan!	[2]; [9]; [10]; [3]; [1]	[1]; [2]; [3]; [9]; [10]
00:54	Wow, huge body *big come on guys, run!	[2]; [3]; [1]	[2]; [1]; [9]
01:47	Jan Jin Juean [confused about saying his own name] what do you want coming here!?	-	[3]; [10]
02:07	Okay Eren, you can do it damn, almost there!	[2]	[2]; [10]
03:18	Who are you going to kick in the butt?	[2]	[2]
03:24	Come here, you giant Titan why did you disappear? This is an emergency, everyone get ready, babe whatever happens we'll be lovey-dovey forever. Ew, lovey-dovey, bebe, Eren where are you?	[2]	[2]
04:07	Eren is dead!	[1]; [10]; [3]; [9]	-
04:13	WHAT A BURDEN!	[10]; [3]; [1]; [9]; [2]	[10]; [3]; [1]; [2]; [9]
??:??	Bye!	[10]; [3]; [1]; [9]	[2]
05:38	Come on, admit it you're an evil Titan, right?	[10]; [1]; [3]; [9]	[3]; [10]; [1]; [2]; [9]
09:29	You really crossed the line!	-	[3]; [9]; [10]; [1]
10:57	This was all your plan after this you'll definitely be roasted oh, don't worry I can handle it. What the heck, sir? Facilities are destroyed, people are falling, who's responsible?	[10]; [3]; [1]; [9]	[9]

This application implements a multi-label classification approach that analyzes each sentence extracted from YouTube video transcripts. The transcripts are retrieved via SearchAPI.io to avoid limitations in the YouTube API. Sentences are segmented using punctuation-based regex and preprocessed with the same steps used during training: case folding, filtering, and tokenization with the IndoBERTweet tokenizer.

The classification model outputs probabilities for 13 labels per sentence. Any label exceeding 0.5 is considered active. If a sentence contains any harmful category other than positive content (PS), it is classified as problematic. A video is flagged as potentially inappropriate if problematic sentences exceed 10% of its total content. The 10% threshold is based on studies on threshold optimization [30], moderation trade-offs between sensitivity and false positives [31], and evidence that small amounts of harmful content can influence online discourse [32]. It also aligns with evaluations that emphasize proportional cutoffs [33]. Figure 5 illustrates the application workflow and output interface, including video input, overall analysis, sentence-level detection, and detailed predictions.

To compare inference performance, the fifth conventional supervised model was also evaluated. The MAML-based model processed a sample video in 11.42 seconds, faster than the baseline model at 13.99 seconds. It also produced slightly more accurate results, detecting 10 problematic sentences compared to 11 detected by the baseline, which included several misclassifications. Table 10 shows the predicted class rankings for each detected sentence. The label indices are defined as follows: a) HS, b) abusive language, c) individual, d) group, e) religion, f) race or ethnicity, g) physical attributes, h) gender, i) other categories, j) mild severity, k) moderate severity, l) severe severity, and m) positive content.

The application was developed using Streamlit for the interface, HuggingFace Transformers and PyTorch for inference, and safetensors.torch for efficient model loading, supported by gdown, requests, re, and os for auxiliary processing tasks. Although YouTube's restrictions on cloud-based traffic necessitated the use of a hardcoded API key, occasionally resulting in RequestBlocked or IpBlocked errors, the MAML-based model consistently delivered stronger generalization and inference efficiency during real-world evaluation, demonstrating its practical suitability for content moderation. To ensure reliable deployment, the system should incorporate bias-mitigation strategies such as human-in-the-loop review, continuous feedback refinement, and drift monitoring across dialects and demographic groups, particularly for sensitive HSAL categories involving gender, religion, and ethnicity.

3.3. Cross-domain evaluation

A cross-domain evaluation was conducted to assess how well the conventional and MAML-based models generalize beyond Twitter data. The models were tested on a short news transcript and several Indonesian YouTube comments, which differ in linguistic style and context. As described in section 3.2, the fifth model pair in Table 5 was used for this evaluation. The results, summarized in Table 11, highlight the robustness of both models in real-world moderation scenarios across different online platforms.

Both models successfully detected HS and abusive language in the news and YouTube data, showing broadly similar performance. However, further validation is needed since no officially annotated datasets exist for these sources in the Indonesian context. Thus, these results should be viewed as preliminary evidence of the models' cross-domain generalization, with future research required to confirm their reliability in practical moderation settings.

Table 11. Cross-domain evaluation

Source	Detected sentence	Predicted class
YouTube comment	Even though it's 02 who cheated, they say it's 01 who cheated, remember, Allah doesn't sleep.	HS_other
YouTube comment	My friend in Malaysia said from the start it's 02 who has already cheated, but the one being accused is 01.	HS_other, HS
YouTube comment	But 01 doesn't make noise this and that like the other side.	HS
YouTube comment	Am I the only one who didn't get an envelope or 'dawn attack' (money politics)? Is this Fritz the one who appeared in Ria Ricis's YouTube video?	PS
YouTube comment	KPU is stupid, too much talking… they said don't abstain, so I came to the polling station with my e-KTP but couldn't vote. Leaders tell us not to abstain but in reality we are rejected as voters because e-KTP alone isn't enough to vote. Second, the campaign period is long but why does KPU have so many problems and still seems like an unprepared institution	HS abusive, HS_group, HS_other, HS_moderate
News	"The higher the level of leadership, the broader the scope, the more complex the organization becomes, the more a leader relies on values, not just technical aspects, but values. That is what distinguishes complex leadership from simple leadership. Now, the question I would like to ask you, Sir, is: what is the relationship between a national leader's ethical standards and their ability to safeguard the nation's defense, security, and sovereignty?" Anies asked Prabowo during the 2024 Presidential Debate at Istora Senayan, Jakarta, Sunday (January 7, 2024).	PS
News	According to Prabowo, the data presented regarding "insiders" and the food estate program was incorrect. He stated that all the data Anies presented was wrong. "So, all the data you revealed is entirely incorrect. I am willing for us to sit down and be transparent. If you want to talk about the food estate, or about PT Teknologi Militer Indonesia, we can lay everything out," Prabowo said.	HS, HS_Individual, HS_Other, HS_Weak

4. CONCLUSION

This study investigated the use of MAML to improve multi-label HSAL detection in Indonesian under severe class imbalance. The results show that MAML provides more stable generalization and stronger discrimination across minority labels than conventional supervised training, making it suitable for low-resource classification settings. The approach also offers practical benefits in deployment, including lower memory usage and competitive inference efficiency. These findings position meta-learning as a promising direction for HSAL detection, particularly in environments with distribution shifts and noisy real-world data. Future work may explore alternative meta-learning algorithms and broader cross-domain evaluations to further validate robustness in practical moderation scenarios.

ACKNOWLEDGEMENTS

The authors would like to acknowledge the Institute for Research and Community Service (*Lembaga Penelitian dan Pengabdian Masyarakat*), Institut Teknologi Nasional Bandung, for providing financial support for the publication of this article. This research assignment letter is 78b/J.016/LPPM/Itenas/II/2025.

FUNDING INFORMATION

This research did not receive specific funding for its implementation. However, financial support for article publication was provided by the Institute for Research and Community Service (Lembaga Penelitian dan Pengabdian Masyarakat), Institut Teknologi Nasional Bandung, under Research Assignment Letter No. 78b/J.016/LPPM/Itenas/II/2025.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Jasman Pardede	✓	✓	✓	✓	✓	✓	✓		✓	✓		✓	✓	✓
Ghixandra Julyaneu Irawadi	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			
Rizka Milandga Milenio						✓			✓	✓	✓	✓	✓	✓

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

Authors state no conflict of interest.

INFORMED CONSENT

This study used publicly available data and did not involve direct interaction with human participants. No identifiable personal information was disclosed. Therefore, informed consent was not required.

ETHICAL APPROVAL

This study used publicly available datasets and did not involve human participants or personal data collection. Therefore, ethical approval was not required.

DATA AVAILABILITY

The dataset used in this study was derived from a previously published dataset and further preprocessed for experimental purposes. The processed dataset supporting the findings of this study is publicly available on Kaggle at https://www.kaggle.com/datasets/ghixandrajulyaneu/hsal-dataset-ghixandra/data, in accordance with the original dataset's usage terms. The original dataset can be accessed through its official repository at https://github.com/Bagus324/Indonesian-HSA-Tweet-with-Deep-Learning-PLM.

REFERENCES

[1] C. Leuprecht, D. B. Skillicorn, and D. Kernot, "Linguistic models of abusive language," *Dynamics of Asymmetric Conflict: Pathways toward Terrorism and Genocide*, vol. 18, no. 2, pp. 158-174, 2024, doi: 10.1080/17467586.2024.2407922.

[2] I. Touahri and A. Mazroui, "Studying the effect of characteristic vector alteration on Arabic sentiment classification," *Journal of King Saud University - Computer and Information Sciences*, vol. 33, no. 7, pp. 890–898, Sep. 2021, doi: 10.1016/j.jksuci.2019.04.011.

[3] M. Bilewicz and W. Soral, "Hate speech epidemic. The dynamic effects of derogatory language on intergroup relations and political radicalization," *Political Psychology*, vol. 41, no. S1, pp. 3–33, Aug. 2020, doi: 10.1111/pops.12670.

[4] E. W. Pamungkas, D. G. P. Putri, and A. Fatmawati, "Hate speech detection in Bahasa Indonesia: Challenges and opportunities," *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 14, no. 6, pp. 1175–1181, 2023, doi: 10.14569/IJACSA.2023.01406125.

[5] M. O. Ibrohim and I. Budi, "Hate speech and abusive language detection in Indonesian social media: Progress and challenges," Heliyon, vol. 9, no. 8, 2023, doi: 10.1016/j.heliyon.2023.e18647.

[6] F. Koto, J. H. Lau, and T. Baldwin, "IndoBERTweet: A pretrained language model for Indonesian Twitter with effective domain-specific vocabulary initialization," *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 2021, pp. 10660–10668, doi: 10.18653/v1/2021.emnlp-main.833.

[7] E. W. Pamungkas, A. Fatmawati, Y. S. Nugroho, D. Gunawan, and E. Sudarmilah, "Hate speech detection in code-mixed Indonesian social media: Exploiting multilingual languages resources," *2022 Seventh International Conference on Informatics and Computing (ICIC)*, Denpasar, Bali, Indonesia, 2022, pp. 1-5, doi: 10.1109/ICIC56845.2022.10006940.

[8] Z. M. Farooqi, S. Ghosh, and R. R. Shah, "Leveraging transformers for hate speech detection in conversational code-mixed tweets," *Computation and Language*, vol. 3159, pp. 63–74, Dec. 2021, doi: 10.48550/arXiv.2112.09986.

[9] J. F. Kusuma and A. Chowanda, "Indonesian hate speech detection using IndoBERTweet and BiLSTM on Twitter," *JOIV : International Journal on Informatics Visualization*, vol. 7, no. 3, pp. 773–780, Sep. 2023, doi: 10.30630/joiv.7.3.1035.

[10] M. Mozafari, R. Farahbakhsh, and N. Crespi, "Cross-lingual few-shot hate speech and offensive language detection using meta learning," in *IEEE Access*, vol. 10, pp. 14880-14896, 2022, doi: 10.1109/ACCESS.2022.3147588.

[11] A. A. Alzahrani, A. Bramantoro, and A. Permana, "Multi-label text classification on unbalanced Twitter with monolingual model and hyperparameter optimization for hate speech and abusive language detection," *International Journal of Advanced and Applied Sciences*, vol. 11, no. 5, pp. 177–185, 2024, doi: 10.21833/ijaas.2024.05.019.

[12] A. Muzakir, K. Adi, and R. Kusumaningrum, "Short text classification based on hybrid semantic expansion and bidirectional GRU (BiGRU) based method to improve hate speech detection," *Revue d'Intelligence Artificielle*, vol. 37, no. 6, pp. 1471–1481, 2023, doi: 10.18280/ria.370611.

[13] A. Alamsyah and Y. Sagama, "Empowering Indonesian internet users: An approach to counter online toxicity and enhance digital well-being," *Intelligent Systems with Applications*, vol. 22, p. 200394, Jun. 2024, doi: 10.1016/j.iswa.2024.200394.

[14] T. P. Handayani and H. Gani, "Enhancing multi-label hate speech and abusive language detection on Indonesian Twitter using recurrent neural networks with hyperparameter tuning," *Jurnal Ilmiah Teknik Mesin, Elektro dan Komputer*, vol. 3, no. 3, pp. 602–612, 2023, doi: 10.51903/juritek.v3i3.3022.

[15] B. T. Y. Darmawan, B. R. Irnawan, and Y. Suzuki, "Indonesian hate speech and abusive tweets classification with deep learning pre-trained language models," *2023 6th International Conference of Computer and Informatics Engineering (IC2IE)*, Lombok, Indonesia, 2023, pp. 30-35, doi: 10.1109/IC2IE60547.2023.10331354.

[16] S. Lei, W. Yi, C. Ying, and W. Ruibin, "Review of attention mechanism in natural language processing," *Data Analysis and Knowledge Discovery*, vol. 4, no. 5, pp. 1–14, 2020, doi: 10.11925/infotech.2096-3467.2019.1317.

[17] E. Kankevičiūtė, M. Songailaitė, B. Zhyhun, and J. Mandravickaitė, "Lithuanian hate speech classification using deep learning methods," *Automation of Technological and Business Processes*, vol. 15, no. 3, pp. 20–29, 2023, doi: 10.15673/atbp.v15i3.2621.

[18] K. Clark, M.-T. Luong, Q. V. Le, and C. D. Manning, "ELECTRA: Pre-training text encoders as discriminators rather than generators," *8th International Conference on Learning Representations, ICLR 2020*, pp. 1–18, Mar. 2020, doi: 10.48550/arXiv.2003.10555.

[19] S. Bano, S. Khalid, N. M. Tairan, H. Shah, and H. A. Khattak, "Summarization of scholarly articles using BERT and BiGRU: Deep learning-based extractive approach," *Journal of King Saud University - Computer and Information Sciences*, vol. 35, no. 9, p. 101739, 2023, doi: 10.1016/j.jksuci.2023.101739.

[20] M. Xu and S. Liu, "RB_BG_MHA: A RoBERTa-based model with Bi-GRU and multi-head attention for Chinese offensive language detection in social media," *Applied Sciences*, vol. 13, no. 19, 2023, doi: 10.3390/app131911000.

[21] G. B. Herwanto, A. M. Ningtyas, I. G. Mujiyatna, K. E. Nugraha, and I. N. Prayana Trisna, "Hate speech detection in Indonesian twitter using contextual embedding approach," *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*, vol. 15, no. 2, p. 177, 2021, doi: 10.22146/ijccs.64916.

[22] M. Mallick, Y. D. Shim, H. I. Won, and S. K. Choi, "Ensemble-Based model-agnostic meta-learning with operational grouping for intelligent sensory systems," *Sensors*, vol. 25, no. 6, 2025, doi: 10.3390/s25061745.

[23] E. Hashmi, S. Y. Yayilgan, and M. Abomhara, "Metalinguist: enhancing hate speech detection with cross-lingual meta-learning," *Complex and Intelligent Systems*, vol. 11, no. 4, 2025, doi: 10.1007/s40747-025-01808-w.

[24] D. Prasad, K. V. Kadambari, R. Mukati, and S. Singariya, "Real-time multi-lingual hate and offensive speech detection in social networks using meta-learning," *TENCON 2023 - 2023 IEEE Region 10 Conference (TENCON)*, Chiang Mai, Thailand, 2023, pp. 31-35, doi: 10.1109/TENCON58879.2023.10322364.

[25] M. S. Utomo, E. Utami, Kusrini, and A. Setyanto, "Machine learning innovations in code generation: A systematic literature review of methods, challenges and directions," *2024 International Conference on Information Technology and Computing (ICITCOM)*, Yogyakarta, Indonesia, 2024, pp. 24-29, doi: 10.1109/ICITCOM62788.2024.10762291.
[26] L. Wang, S. Zhou, S. Zhang, X. Chu, H. Chang, and W. Zhu, "Improving generalization of meta-learning with inverted regularization at inner-level," *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Vancouver, BC, Canada, 2023, pp. 7826-7835, doi: 10.1109/CVPR52729.2023.00756.
[27] W. Yin, "Meta-learning for few-shot natural language processing: A Survey," *Computation and Language*, Jul. 2020, doi: 10.48550/arXiv.2007.09604.
[28] X. Han and J. Lundin, "Multi-pair text style transfer for unbalanced data via task-adaptive meta-learning," in *Proceedings of the 1st Workshop on Meta Learning and Its Applications to Natural Language Processing*, Stroudsburg, PA, USA: Association for Computational Linguistics, 2021, pp. 28–35, doi: 10.18653/v1/2021.metanlp-1.4.
[29] Y. Zhao *et al*., "Improving meta-learning for low-resource text classification and generation via memory imitation," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Stroudsburg, PA, USA: Association for Computational Linguistics, 2022, pp. 583–595, doi: 10.18653/v1/2022.acl-long.44.
[30] D. Son *et al*., "Reliable decision from multiple subtasks through threshold optimization: Content moderation in the wild," in *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*, New York, NY, USA: ACM, Feb. 2023, pp. 285–293, doi: 10.1145/3539597.3570439.
[31] V. U. Gongane, M. V. Munot, and A. D. Anuse, "Detection and moderation of detrimental content on social media platforms: current status and future directions," *Social Network Analysis and Mining*, vol. 12, no. 1, p. 129, Dec. 2022, doi: 10.1007/s13278-022-00951-3.
[32] M. Ye, K. Sikka, K. Atwell, S. Hassan, A. Divakaran, and M. Alikhani, "Multilingual content moderation: A case study on Reddit," *Computation and Language*, Feb. 2023, doi: 10.48550/arXiv.2302.09618.
[33] D. Antypas, I. Sen, C. Perez-Almendros, J. Camacho-Collados, and F. Barbieri, "Sensitive content classification in social media: A holistic resource and evaluation," *Computation and Language*, Jun. 2025, doi: 10.48550/arXiv.2411.19832.

BIOGRAPHIES OF AUTHORS

Jasman Pardede earned his Bachelor's degree in Science and Mathematics from Universitas Andalas (Unand), Indonesia, in 2001. He went on to receive his Master of Engineering in Informatics Engineering from Institut Teknologi Bandung (ITB) in 2005, and later completed his doctoral degree in the same field at ITB in 2021. His dissertation was titled "Relevance Feedback by Composite Feedback Objects on CBIR." Since 2005, he has been a lecturer at Institut Teknologi Nasional Bandung, Indonesia. His research interests include image retrieval, data mining, machine learning, and deep learning. He can be contacted at email: jasman@itenas.ac.id.

Ghixandra Julyaneu Irawadi holds a Bachelor of Computer Science degree from the Informatics Study Program at Institut Teknologi Nasional (ITENAS) Bandung, Indonesia. Her academic interests include artificial intelligence, natural language processing, and machine learning-based system development. Her current research focuses on multilabel hate speech and abusive language detection in online content. This is her first academic publication. In this paper, she contributed to dataset preprocessing, model development, and the analysis of experimental results. She actively participates in academic and collaborative research projects at the university. She can be contacted at email: ghixandraj@gmail.com.

Rizka Milandga Milenio earned his Bachelor's degree in Science and Mathematics from Universitas Negeri Malang (UM), Indonesia, in 2017. He completed his Master of Engineering in informatics from Institut Teknologi Bandung (ITB) in 2024. Since 2025, he has been a lecturer at Institut Teknologi Nasional Bandung, Indonesia. His research interests include computer vision, smart education, health informatics, machine learning and deep learning. He can be contacted at email: rizkamilandga@itenas.ac.id.