

Performance Comparison of Hybrid IndoBERT-Self-Paced Ensemble and IndoBERT-RUSBoost for Imbalanced Text Classification (Case Study: Insecurity Detection on Social Media X)

Emeylia Safitri¹, I Gusti Ngurah Sentana Putra², Nuramaliyah³, Ika Nur Laily Fitriana⁴, Ria Faulina⁵

^{1,3,4,5}Statistics Study Program, Faculty of Science and Technology, Universitas Terbuka, Indonesia

²Department of Statistic and Data Science, School of Data Science, Mathematics, and Informatics, IPB University, Indonesia

Article Info

Article history:

Received Mar 16, 2026

Revised Mar 31, 2026

Accepted Apr 23, 2026

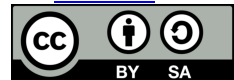
Keywords:

Insecurity detection;
IndoBERT;
RUSBoost;
Self-Paced Ensemble;
Imbalanced text classification

ABSTRACT

Insecurity detection on social media is crucial for early mental health intervention, yet imbalanced data distribution poses significant challenges for text classification. This study compares the performance of hybrid IndoBERT-Self-Paced Ensemble (SPE) and IndoBERT-RUSBoost for imbalanced text classification in insecurity detection on social media X. A dataset of 5,179 Indonesian tweets was collected using keywords "insecure" and "overthinking", preprocessed through text cleaning and slang normalization, and labeled automatically using a fine-tuned IndoBERT model as insecurity (positive) or non-insecurity (negative). Results demonstrated that RUSBoost significantly outperformed SPE across 8 of 13 evaluation metrics, achieving balanced accuracy of 0.8521, F1-score of 0.6763, and AUC-ROC of 0.9459 compared to SPE's 0.8465, 0.5171, and 0.9242 respectively. RUSBoost showed superior specificity (0.9460 vs 0.8381) and precision (0.6104 vs 0.3706), while SPE excelled in recall (0.8548 vs 0.7581). Gradient Boosting achieved the highest AUC-PR (0.7100) and specificity (0.9622). The findings confirm that hybrid IndoBERT-RUSBoost provides more balanced and robust performance for imbalanced text classification, offering practical implications for early insecurity detection systems on social media platforms.

This is an open access article under the [CC BY-SA](#) license.



Corresponding Author:

Emeylia Safitri,
Statistics Study Program,
Universitas Terbuka,
Jl. Cabe Raya, Pondok Cabe, Pamulang, Tangerang Selatan 15437, Banten, Indonesia.
Email: emeylia.safitri@ecampus.ut.ac.id

1. INTRODUCTION

Social media has become an inseparable part of modern society. Platform X (formerly Twitter) stands as one of the most popular social media platforms with millions of daily active users. The high intensity of usage has led to increased social interactions that are not always positive, including the emergence of insecurity phenomena or feelings of anxiety openly expressed on social media [1]. Insecurity is an emotional state characterized by feelings of anxiety, lack of confidence, doubt, and excessive worry about various aspects of life.

In Indonesia, the prevalence of negative content on social media has attracted attention from various stakeholders. X accounts contributed 79 accounts receiving warnings from cyber police regarding violations of the Information and Electronic Transactions (ITE) Law [2]. Cyberbullying, hate speech, and offensive language present overlapping problems that complicate identification [3]. Various approaches for cyberbullying detection on Indonesian Twitter have been demonstrated, using both Random Forest [4] and K-Nearest Neighbor [5]. However, specific

detection of insecurity expressions remains rarely conducted, despite their equally serious psychological impacts.

Automatic detection of insecurity content on social media faces significant technical challenges, particularly class imbalance. Insecurity content data is typically far less abundant compared to non-insecurity content. Class imbalance constitutes a critical challenge in text classification because the underrepresentation of minority classes hinders the classifier's ability to learn patterns effectively [6]. Various binary classifiers on imbalanced datasets tend to be biased toward the majority class [7], [8].

One promising approach is ensemble learning, which combines multiple base classifiers to improve prediction performance. In the context of imbalanced data, two ensemble methods that have attracted researchers' attention are Self-Paced Ensemble (SPE) and RUSBoost. SPE employs a self-paced undersampling approach based on sample hardness distribution [9]. SPE has proven effective for classification with significant data imbalance [8], [10] and has been applied to voice and audio classification [11].

On the other hand, RUSBoost combines random undersampling with the SAMME.R boosting algorithm [12]. Various ensemble techniques for handling class imbalance demonstrate that RUSBoost is competitive [13]. Implementation of RUSBoost in the Indonesian educational context has also shown its effectiveness on local data [14].

Direct comparison between SPE and RUSBoost in the context of poverty classification in West Java revealed that RUSBoost consistently outperformed SPE on important metrics such as AUC-ROC, AUC-PR, and G-Mean, with substantially higher recall in detecting the minority class [15]. However, this study employed tabular data rather than text data, thus the results may not necessarily apply to the text classification context. Comprehensive comparison of various methods for imbalanced data found that ensemble learning methods generally perform better [16].

In the domain of Indonesian language text classification, the development of transformer models such as IndoBERT opens new opportunities for richer and more contextual feature extraction. IndoBERT has been implemented for public sentiment analysis with 85% accuracy while applying imbalanced data handling strategies [17]. Several transformer models including IndoBERT have also been compared for hate speech detection [3]. The advantages of IndoBERT for various Indonesian text analysis tasks have been demonstrated in multiple studies [18], [19].

Research on integrating transformer models with ensemble methods also shows promising results. Heterogeneous ensemble methods combining various deep learning and machine learning models for text classification achieved better performance than single models [20]. The combination of BERT with additional features in fake news detection achieved 98% accuracy, outperforming models without such integration [21].

However, research integrating IndoBERT with imbalance handling methods such as SPE and RUSBoost remains very limited, particularly for insecurity detection on Indonesian social media. Undetected insecurity expressions can have serious impacts on users' mental health. Existing research focuses more on cyberbullying [4], [22], [23] or hate speech [3], while insecurity as a potential indicator of mental health issues has not received adequate attention.

The selection of appropriate evaluation metrics is also crucial in the context of imbalanced data. Simple accuracy can be misleading, necessitating metrics such as balanced accuracy, F1-score, G-mean, and Matthews Correlation Coefficient [3]. Notably, Youden's Index emerges as an important metric because it measures the classifier's ability to balance sensitivity and specificity simultaneously [15], [24]. This is particularly relevant for insecurity detection where false negatives and false positives carry different consequences.

Based on the above elaboration, this study aims to compare the performance between hybrid IndoBERT-SPE and IndoBERT- RUSBoost for imbalanced text classification in insecurity detection on social media X. The hybrid approach refers to using IndoBERT as a feature extractor, which then becomes input for the SPE and RUSBoost algorithms. Thus, this study combines the advantages of contextual representation from transformers with the imbalance handling advantages of ensemble methods. The research results are expected to contribute to the development of early detection methods for public mental health indicators in the digital era.

2. RESEARCH METHOD

Figure 1 illustrates the research methodology used in this study. The process starts with data loading and text preprocessing, followed by splitting the data into training and testing sets. IndoBERT is used for feature extraction, while SVM and RusBoost models are optimized using Grid Search Hyperparameter Tuning. Finally, the models are evaluated using performance metrics, and the optimal classification threshold is determined based on the Youden Index.

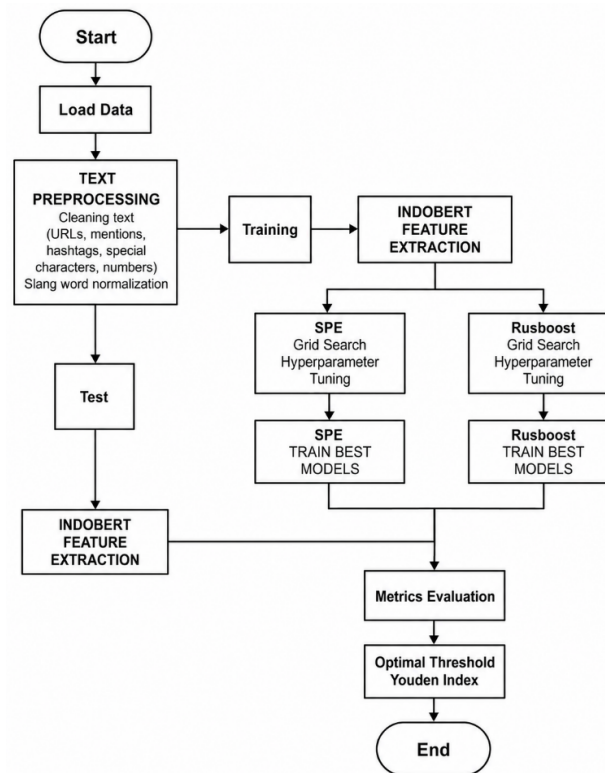


Figure 1. Research Methodology

2.1. Dataset

The dataset used in this study was collected from social media platform X (formerly Twitter) through a crawling process using the keyword "insecure" and "overthinking" in Indonesian language. Data collection was conducted over a specific time period to capture a representative sample of public expressions related to insecurity and overthinking. The crawling process utilized the Twitter API integrated with Google Colab to retrieve tweets containing the specified keywords [23]. The raw dataset consists of several attributes including tweet ID, username, tweet content, timestamp, and metadata such as number of likes and retweets. The collected data represents various forms of insecurity expressions, including self-doubt, social comparison, anxiety about the future, body image concerns, and excessive rumination about personal problems. The dataset is designed to capture the linguistic characteristics of insecurity expressions in Indonesian social media context, which often includes informal language, slang, and emotional vocabulary [17].

2.2. Dataset Preparation

Dataset preparation involves a series of preprocessing steps to clean and normalize the raw text data before feature extraction. The preprocessing pipeline follows the approach implemented in the TextPreprocessor class, which consists of two main stages: text cleaning and slang normalization. Text cleaning begins with removing noise elements such as URLs, mentions (@username), and hashtags (#topic) that do not contribute to the semantic meaning of the text [4]. Special characters, punctuation marks, and numbers are also removed as they typically do not carry significant information for sentiment classification. The text is then converted to lowercase through case folding to ensure uniformity and prevent the duplication of words with different cases. Finally, excessive whitespace is removed to obtain clean, standardized text.

The second stage of preprocessing involves slang normalization, which is particularly important for Indonesian social media text where informal language and abbreviations are prevalent. The normalization process utilizes a comprehensive slang dictionary compiled from multiple sources, including `slang_indo.csv`, `kamusalay.csv`, and `slang_indonesia.csv`. Each word in the cleaned text is checked against the slang dictionary, and if a match is found, the word is replaced with its formal equivalent [18]. For example, informal words such as "gak", "nggak", or "ga" are normalized to "tidak", while "banget" is normalized to "sangat". This normalization step ensures that the text representation captures the semantic meaning consistently, regardless of the informal variations used by social media users. The preprocessing steps are applied to all text data before feature extraction to ensure data quality and consistency for subsequent modeling stages.

2.3. Labeling

The labeling process for insecurity detection utilizes a fine-tuned IndoBERT model to automatically annotate tweets as either insecurity (positive class) or non-insecurity (negative class). IndoBERT, a transformer-based model pre-trained on a large Indonesian text corpus, has demonstrated strong performance in various Indonesian text classification tasks, including sentiment analysis and hate speech detection. The model was fine-tuned on a manually annotated dataset of Indonesian tweets to learn the linguistic patterns associated with insecurity expressions, such as self-doubt, excessive worry, social comparison, feelings of inadequacy, and rumination about personal problems. This approach enables efficient large-scale labeling while maintaining consistency through the model's contextual understanding of informal social media language.

The final labeled dataset includes the original tweet text along with its binary label, which is then encoded numerically using LabelEncoder where 0 represents the negative class (non-insecurity) and 1 represents the positive class (insecurity). The class distribution is carefully examined to determine the imbalance ratio, which is crucial for selecting appropriate imbalance handling techniques in the subsequent modeling stages [6]

2.4. IndoBERT Feature Extraction

IndoBERT is a transformer-based model built on the BERT architecture, specifically developed for Indonesian language processing. The model is trained on a large Indonesian text corpus, enabling it to understand linguistic contexts, sentence structures, and various language variations within the Indonesian language [17]. IndoBERT has several variants, such as IndoBERT-base and IndoBERT-large, which are distinguished by the number of parameters and the depth of their transformer layers. The base variant is lighter and commonly used for text classification or sentiment analysis tasks with medium-sized datasets, featuring 12 transformer layers and a 768-dimensional vector representation [18]. IndoBERT's ability to capture semantic and contextual nuances of the Indonesian language has proven superior to traditional representations such as TF-IDF or static word embeddings, as its bidirectional architecture allows the model to analyze relationships between tokens in both directions, so the meaning of each word is understood based on the entire sentence [25].

The feature extraction process with IndoBERT begins with the tokenization stage, as illustrated in Figure 2, where the model adds special tokens: [CLS] at the beginning of the sentence as a representation of the entire input, [SEP] at the end of the sentence as a boundary marker, and [PAD] to equalize sequence lengths within each batch [17]. All tokens are then converted into numerical indices based on the vocabulary established by the IndoBERT tokenizer. These token indices are subsequently mapped into a vector space through embeddings that include token embedding, position embedding, and segment embedding. The embedding results serve as input to the IndoBERT architecture, which consists of 12 transformer encoder layers, each equipped with a multi-head self-attention mechanism [19]. The 768-dimensional vector representation from the [CLS] token is commonly used as extracted features for input to other classification algorithms, including in hybrid approaches with ensemble learning methods. Mathematically, the self-attention process in IndoBERT can be represented as:

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

where Q , K , and V are the query, key, and value matrices respectively, and d_k is the dimension of the key vectors. The resulting contextual representations enable the model to capture word meanings based on the entire sentence context, rather than just surrounding words in a unidirectional manner.

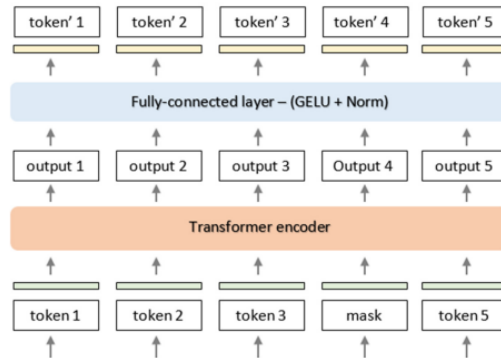


Figure 2. IndoBERT Tokenization Stage [26]

2.5. Class Imbalanced

Class imbalance refers to a situation where the number of instances between classes in a dataset is disproportionate, with one class (the minority class) being far fewer than the other class(es) (the majority class). In text classification, class imbalance becomes a critical challenge because the underrepresentation of the minority class hinders the classifier's ability to learn patterns effectively [6]. Traditional classification algorithms tend to be biased toward the majority class because their training process is dominated by the majority class, while the minority class is treated as noise [7]. The Imbalance Ratio (IR) is defined as the ratio between the number of majority class samples and minority class samples [9]:

$$IR = \frac{n_{\text{majority}}}{n_{\text{minority}}} = \frac{|\mathcal{N}|}{|\mathcal{P}|}$$

where $\mathcal{N}$ is the set of majority class instances and $\mathcal{P}$ is the set of minority class instances. Various methodologies have been developed to mitigate the effects of class imbalance in classification. These methodologies are generally divided into three groups [16]. The first group involves interventions at the data level through resampling techniques such as oversampling, which increases the number of minority class observations, and undersampling, which reduces the number of majority class observations. Oversampling risks causing overfitting and requires substantial computational resources, while undersampling risks losing important information from the majority class [3]. The second group focuses on modifying classification algorithms to account for the significance of the minority class during training, such as cost-sensitive learning where different misclassification costs are assigned to each class. The third group consists of hybrid methods designed to combine the strengths of each approach while reducing their limitations, including ensemble techniques from resampled datasets [9]. In the context of insecurity detection on social media, class imbalance becomes a primary challenge because insecurity content is typically far less frequent than non-insecurity content, requiring special handling to enable models to accurately identify the minority class.

2.6. RUSBoost

The workflow of the RUSBoost algorithm is illustrated in Figure 3. RUSBoost is a hybrid algorithm that combines random undersampling (RUS) with boosting algorithms to improve classification performance on imbalanced datasets [12]. This algorithm was developed as a simpler and faster alternative to SMOTEBoost while maintaining competitive performance. In each boosting iteration, RUSBoost applies random undersampling by randomly removing samples from the majority class until a desired class distribution is achieved, then trains a base classifier on the undersampled dataset [13]. The main advantages of RUSBoost lie in its implementation simplicity, computational efficiency, and availability in the imbalanced-learn library, which facilitates integration with various base classifiers. Although undersampling risks information loss, combining it with boosting allows information missing in one iteration to appear in others, so the

overall model remains capable of capturing important patterns from the data [12]. Formally, the RUSBoost algorithm can be expressed as follows. Given training data $S = \{(x_1, y_1), \dots, (x_m, y_m)\}$ with minority class $y^* \in Y$ and $|Y| = 2$. Initialize the weight of each sample as $D_1(i) = 1/m$ for all $i = 1, \dots, m$. For each iteration $t = 1, \dots, T$:

1. Create a temporary training dataset S'_t with weight distribution D'_t using random undersampling
 2. Call *WeakLearn* with examples S'_t and weights D'_t to obtain hypothesis $h_t: X \times Y \rightarrow [0,1]$
 3. Calculate the pseudo-loss $\epsilon_t = \sum_{(i,y): y_i \neq y} D_t(i)(1 - h_t(x_i, y_i) + h_t(x_i, y))$
 4. Calculate the weight update parameter $\alpha_t = \frac{\epsilon_t}{1 - \epsilon_t}$
 5. Update weights $D_{t+1}(i) = D_t(i)\alpha_t^{\frac{1}{2}(1+h_t(x_i, y_i) - h_t(x_i, y))}$ and normalize
- The final hypothesis is produced as a weighted vote of all weak hypotheses [12]:

$$H(x) = \arg \max_{y \in Y} \sum_{t=1}^T h_t(x, y) \log \frac{1}{\alpha_t}$$

Algorithm RUSBoost
Given: Set S of examples $(x_1, y_1), \dots, (x_m, y_m)$ with minority class y^* in Y , $|Y| = 2$
 Weak learner, *WeakLearn*
 Number of iterations, T
 Desired percentage of total instances to be represented by the minority class, N

- 1 Initialize $D_1(i) = \frac{1}{m}$ for all i .
- 2 Do for $t = 1, 2, \dots, T$
 - a Create temporary training dataset S'_t with distribution D'_t using random undersampling
 - b Call *WeakLearn*, providing it with examples S'_t and their weights D'_t .
 - c Get back a hypothesis $h_t: X \times Y \rightarrow [0, 1]$.
 - d Calculate the pseudo-loss (for S and D_t):

$$\epsilon_t = \sum_{(i,y): y_i \neq y} D_t(i)(1 - h_t(x_i, y_i) + h_t(x_i, y)).$$
 - e Calculate the weight update parameter:

$$\alpha_t = \frac{\epsilon_t}{1 - \epsilon_t}.$$
 - f Update D_t :

$$D_{t+1}(i) = D_t(i)\alpha_t^{\frac{1}{2}(1+h_t(x_i, y_i) - h_t(x_i, y; y \neq y_i))}.$$
 - g Normalize D_{t+1} . Let $Z_t = \sum_i D_{t+1}(i)$.

$$D_{t+1}(i) = \frac{D_{t+1}(i)}{Z_t}.$$
- 3 Output the final hypothesis:

$$H(x) = \operatorname{argmax}_{y \in Y} \sum_{t=1}^T h_t(x, y) \log \frac{1}{\alpha_t}.$$

Figure 3. RUSBoost Algorithm [12]

2.7. Self-Paced Ensemble

The Self-Paced Ensemble (SPE) process is illustrated in Figure 4. Self-Paced Ensemble (SPE) is an ensemble learning framework specifically designed to handle imbalanced data classification through an adaptive undersampling mechanism based on sample hardness distribution [9]. The basic concept of SPE is inspired by self-paced learning, where the model learns samples gradually, starting from the easiest to the most difficult. In the context of SPE, hardness is defined as the difficulty level of correctly classifying a sample by the current model, calculated using error functions such as absolute error, squared error, or cross entropy. The main advantage of SPE lies in its ability to consider the hardness distribution across the entire dataset, so the undersampling process not only balances class distribution but also preserves informative samples from the majority class. This approach enables SPE to achieve robust performance even under high class overlapping and extremely skewed data distributions [9].

Algorithmically, SPE is defined as follows. Let $\mathcal{D}$ be the set of all training samples (x, y) , with $\mathcal{P}$ as the minority class set and $\mathcal{N}$ as the majority class set, where $|\mathcal{N}| \gg |\mathcal{P}|$. The hardness function $\mathcal{H}(x, y, F)$ is calculated based on the current ensemble model F . Majority samples are then divided into k bins based on their hardness values [9]:

$$B_\ell = \left\{ (x, y) \mid \frac{\ell - 1}{k} \leq \mathcal{H}(x, y, F) < \frac{\ell}{k} \right\}, \ell = 1, \dots, k$$

The average hardness contribution in each bin is calculated as $h_\ell = \frac{1}{|B_\ell|} \sum_{s \in B_\ell} \mathcal{H}(x_s, y_s, F)$. The self-paced factor α is updated using the tangent function: $\alpha = \tan\left(\frac{i\pi}{2T}\right)$ where i is the current iteration and T is the total number of iterations. The sampling weight for each bin is calculated as [9]:

$$p_\ell = \frac{1}{h_\ell + \alpha}, \ell = 1, \dots, k$$

The undersampling process is then performed from each bin according to its weight to obtain a new balanced training dataset. SPE has also been proven superior in memory efficiency with the lowest memory usage compared to other ensemble methods [16].

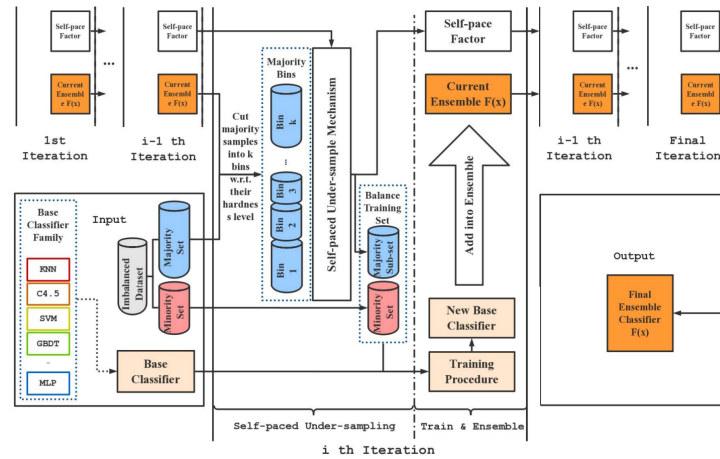


Figure 4. Self-Paced Ensemble Process [9]

2.8. Evaluation Metrics

Selecting appropriate evaluation metrics is crucial in the context of imbalanced data, as simple accuracy can be misleading. A model that always predicts the majority class can achieve high accuracy yet fail to identify the minority class, which is often the primary [15]. Evaluation metrics for imbalanced data are generally derived from the confusion matrix, which contains true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN) [3]. Based on these components, several key metrics are defined as follows:

1. Balanced accuracy is the average of sensitivity and specificity:

$$\text{Balanced Accuracy} = \frac{1}{2} \left(\frac{TP}{TP + FN} + \frac{TN}{TN + FP} \right)$$

This metric is more robust for imbalanced data compared to regular accuracy [15].

2. The F1-score is the harmonic mean of precision and recall:

$$F_1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

with Precision = $\frac{TP}{TP + FP}$ and Recall = $\frac{TP}{TP + FN}$ [16].

3. The G-mean (geometric mean) is calculated as

$$G - \text{mean} = \sqrt{\text{Recall} \times \text{Specificity}}$$

with Specificity = $\frac{TN}{TN + FP}$, measuring the balance between the ability to identify positive and negative classes [9].

4. The Matthews Correlation Coefficient (MCC) is a more informative metric for imbalanced data as it considers all elements of the confusion matrix [3]:

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

MCC values range from -1 to 1, where 1 indicates perfect prediction, 0 indicates random prediction, and -1 indicates completely wrong prediction. AUC-ROC (Area Under the Receiver Operating Characteristic curve) measures the model's ability to distinguish between classes across various thresholds, while AUC-PR (Area Under the Precision-Recall curve) is more sensitive to performance on the minority class [6].

5. Youden's Index, formulated as $J = \text{sensitivity} + \text{specificity} - 1$, is an important metric because it measures the classifier's ability to balance both aspects simultaneously [24]. Youden's Index can also be used to determine the optimal threshold that maximizes this balance. The optimal threshold τ^* is obtained by maximizing Youden's Index:

$$\tau^* = \arg \max_{\tau \in [0,1]} (\text{Sensitivity}(\tau) + \text{Specificity}(\tau) - 1)$$

In the study by [15], the use of Youden's Index to determine the optimal threshold on the RUSBoost model increased the F1-score from 0.190 to 0.220 and MCC from 0.215 to 0.221, demonstrating the importance of this metric in model optimization for imbalanced data.

3. RESULTS AND DISCUSSION

3.1. Dataset Overview

This study used a dataset collected from social media platform X using the keywords "insecure" and "overthinking" in Indonesian language. The data collection process yielded 5,526 raw data points, which then underwent a filtering process to ensure only binary classes (positive and negative) were used in modeling. After filtering, 5,179 data points remained for further processing.

Data labeling was performed automatically using the IndoBERT model fine-tuned for Indonesian sentiment classification. It is important to note that this labeling process was conducted by an AI-based model and not by expert annotators in psychology or mental health; therefore, the resulting labels reflect linguistic patterns learned during pre-training rather than clinical assessments. Following labeling, a resampling step was applied to the training data to achieve a target imbalance ratio of 90:10 (negative: positive), resulting in 2,470 training samples. The test set retained 618 samples with the original class proportions to reflect real-world conditions. This extreme imbalance ratio of 1:9.04 in the training set represents a significantly more challenging scenario than near-balanced distributions and constitutes the primary motivation for employing specialized imbalance-handling ensemble methods.

Table 1. Initial Class Distribution

Class	Train Count	Train %	Test Count	Test %
Negative	2,224	90.04%	556	89.97%
Positive	246	9.96%	62	10.03%
Total	2,470	100%	618	100%

Based on Table 1, the class distribution reflects a severe imbalance condition with the negative class dominating at 90.04% of the training set, while the positive class (insecure) represents only 9.96%. The stratified split maintained this proportion consistently in the test set (89.97% negative, 10.03% positive), ensuring that the evaluation accurately reflects the real-world class prevalence. The imbalance ratio is 1:9.04, which falls into the extreme imbalance category [6]. Nevertheless, handling class imbalance remains necessary to ensure the model can generalize well in both classes.

3.2. Data Preprocessing

The text preprocessing process was conducted in two main stages: cleaning and slang normalization. The slang dictionary consisted of 3,993 words compiled from three sources (slang_indo.csv, kamusalay.csv, and slang_indonesia.csv). Complete preprocessing statistics are presented in Table 2.

Table 2. Text Preprocessing Statistics

Metric	Value
Average Text Length (Original)	102.40 characters
Average Text Length (Cleaned)	97.54 characters
Average Word Count (Original)	16.00 words

Average Word Count (Cleaned)	15.39 words
Number of Slang Words in Dictionary	3,993
Total Slang Detected	44,011
Unique Words (Original)	19,082
Unique Words (Cleaned)	11,209
Minimum Text Length	9 characters

The cleaning process successfully reduced the average text length from 102.40 to 97.54 characters and the average word count from 16.00 to 15.39 words. Slang detection identified 44,011 occurrences of informal words in the dataset, which were successfully normalized using the compiled dictionary. This normalization step is particularly important for informal Indonesian social media language, as it helps the model understand text meaning more consistently [18].

3.3. Training and Testing Data Split

The dataset was divided into training data (80%) and testing data (20%) using a stratified split to maintain class proportions in both subsets. It is important to clarify the distinction between these two evaluation frameworks: the 80/20 split is used exclusively for final model evaluation on held-out test data never seen during training, while 5-fold cross-validation within grid search is confined entirely to the training set for hyperparameter selection purposes. This separation strictly prevents data leakage between parameter tuning and final evaluation [17]. Table 3 presents the class distribution alongside the imbalance ratio.

Table 3. Class Distribution in Training and Testing Data

Class	Training (n)	Testing (n)
Negative	2,224	556
Positive	246	62
Total	2,470	618

The class distribution in training and testing data is highly consistent, with a difference of less than 0.07%. The imbalance ratio of 1:9.04 in the training set falls into the extreme imbalance category, underscoring the necessity of employing dedicated imbalance-handling methods for reliable minority-class detection.

3.4. IndoBERT Feature Extraction

Feature extraction using the IndoBERT-base-p1 model produced 768-dimensional vector representations for each text. This process was performed separately for training data (2,470 samples) and testing data (618 samples). Feature normalization using StandardScaler was applied to training data with `fit_transform`, while testing data was only transformed using the same pre-fitted parameters to prevent data leakage.

Table 4. IndoBERT Feature Statistics After Normalization

Metric	Training Set	Testing Set
Feature Dimensions	(2,470, 768)	(618, 768)
Value Range	[-6.5340, 7.1171]	[-6.7476, 6.5405]
Mean	-0.0000 ± 1.0000	0.0002 ± 1.0042

The normalization results in Table 4 show that the extracted features have been well standardized, with means close to 0 and standard deviations close to 1. Consistency between training and testing data is also maintained, which is important for model generalization [17].

3.5. Hyperparameter Tuning Results with Grid Search

Grid search with 5-fold cross-validation was conducted exclusively on the training data to find the optimal parameter combinations. This process is fully contained within the training set and does not involve the test data in any form, ensuring unbiased final evaluation.

For the RUSBoost model, the optimized parameters included `n_estimators` [30, 50, 100], `learning_rate` [0.05, 0.1], and `estimator__max_depth` [5, 10, 15], with a total of 18 combinations and 90 models trained. The top five combinations are presented in Table 5.

Based on Table 5, the best parameter combination for RUSBoost was `n_estimators` = 100, `learning_rate` = 0.10, and `estimator__max_depth` = 15, achieving a mean test balanced accuracy of 0.8877. This result indicates that higher model complexity, combined with a larger number of

estimators, provides superior performance under extreme class imbalance, as the model requires greater representational capacity to learn sparse minority-class patterns effectively.

Table 5. RUSBoost Grid Search Results (Top 5 Combinations)

n_estimators	Learning_rate	Estimator_max_depth	Mean Test Balanced Accuracy
100	0.10	15	0.8877
100	0.10	10	0.8738
100	0.05	15	0.8597
50	0.10	15	0.8437
100	0.05	10	0.8338

For the SPE model, the optimized parameters included n_estimators [30, 50, 100], max_depth [5, 10, 15], and n_bins [10, 15, 20], with a total of 27 combinations and 135 models trained. The top five combinations are presented in Table 6.

Table 6. SPE Grid Search Results (Top 5 Combinations)

n_estimators	max_depth	n_bins	Mean Test Balanced Accuracy
100	5/10/15	15	0.8616
100	5/10/15	10	0.8612
30	5/10/15	15	0.8609
50	5/10/15	20	0.8602
100	5/10/15	20	0.8589

Table 6 reveals an interesting pattern: SPE achieves nearly identical balanced accuracy values across different max_depth settings (5, 10, and 15) for the same n_bins and n_estimators combinations. This suggests that SPE performance on high-imbalance text data is more sensitive to n_bins and n_estimators than to tree depth, as the self-paced resampling mechanism implicitly controls learning difficulty through difficulty-based sample weighting.

Figure 5 presents heatmaps comparing the performance of SPE and RUSBoost across various parameter combinations. In SPE, the highest balanced accuracy value (0.8616) was achieved with n_estimators = 100 and n_bins = 15, with variations in max_depth showing minimal effect. In RUSBoost, the highest value (0.8877) was achieved with n_estimators = 100 and estimator_max_depth = 15 at learning_rate = 0.10. This indicates that RUSBoost performance is highly sensitive to model complexity and the learning rate, whereas SPE performance is more influenced by the number of estimators and bins.

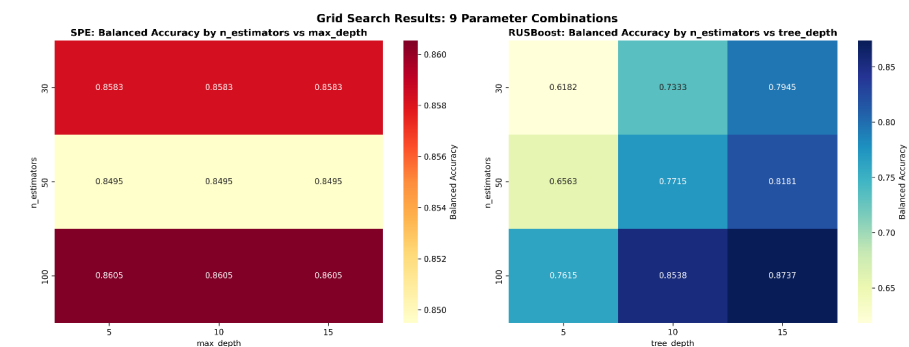


Figure 5. Heatmap Comparison of SPE and RUSBoost Grid Search

3.6. Model Evaluation on Testing Data

After obtaining the optimal hyperparameters, all four models SPE, RUSBoost, Logistic Regression (with class_weight='balanced'), and standard Gradient Boosting were retrained on the full training set and evaluated on the held-out test set of 618 samples. Logistic Regression and Gradient Boosting were included as baseline comparators to provide a more comprehensive performance benchmark beyond the two proposed ensemble methods, addressing the limitation of exclusively comparing SPE and RUSBoost without stronger baselines. Table 7 presents a comprehensive comparison across all models.

Table 7. Performance Comparison of SPE and RUSBoost on Testing Data

Metric	SPE	RUSBoost	Logistic Regression	Gradient Boosting	Best Model
Accuracy	0.8398	0.9272	0.9126	0.9239	RUSBoost
Balanced Accuracy	0.8465	0.8521	0.7866	0.7714	RUSBoost
Recall	0.8548	0.7581	0.6290	0.5806	SPE
Specificity	0.8381	0.9460	0.9442	0.9622	Gradient Boosting
Precision	0.3706	0.6104	0.5571	0.6316	Gradient Boosting
F1-Score	0.5171	0.6763	0.5909	0.6050	RUSBoost
G-Mean	0.8464	0.8469	0.7707	0.7475	RUSBoost
MCC	0.4937	0.6405	0.5434	0.5637	RUSBoost
Kappa	0.4385	0.6358	0.5422	0.5630	RUSBoost
AUC-ROC	0.9242	0.9459	0.8890	0.9424	RUSBoost
AUC-PR	0.6084	0.6142	0.5568	0.7100	Gradient Boosting
Youden's Index	0.6966	0.7915	0.6281	0.7828	RUSBoost

Based on Table 7, RUSBoost achieves the best overall balance, leading in 8 out of 13 metrics including Balanced Accuracy (0.8521), F1-Score (0.6763), MCC (0.6405), Kappa (0.6358), AUC-ROC (0.9459), Youden's Index (0.7915), Accuracy (0.9272), and G-Mean (0.8469). SPE excels exclusively in Recall (0.8548), reflecting its tendency toward aggressive positive classification driven by the self-paced undersampling mechanism. However, this high recall comes at a significant cost to precision (0.3706 vs. 0.6104 for RUSBoost), resulting in a substantially lower F1-Score (0.5171 vs. 0.6763).

A particularly noteworthy finding is that standard Gradient Boosting without any explicit imbalance handling achieves the highest Specificity (0.9622), Precision (0.6316), and AUC-PR (0.7100) among all models, while exhibiting the lowest Balanced Accuracy (0.7714). This pattern confirms the well-known majority-class bias in standard classifiers under extreme imbalance, where high overall accuracy metrics mask poor minority-class detection performance [12].

Regarding training time, there is a practically meaningful difference among models. Logistic Regression requires only 0.53 seconds, making it highly feasible for real-time applications. SPE requires 52.36 seconds and RUSBoost 76.34 seconds a difference of approximately 24 seconds (approximately 46% longer for RUSBoost). Unlike the original study which reported comparable training times (SPE: 138.12s, RUSBoost: 140.87s) and characterized the gap as non-significant, the updated results reveal a more meaningful time gap between SPE and RUSBoost that warrants consideration in frequent model retraining scenarios. Gradient Boosting's 253.01 seconds make it the least computationally efficient option.

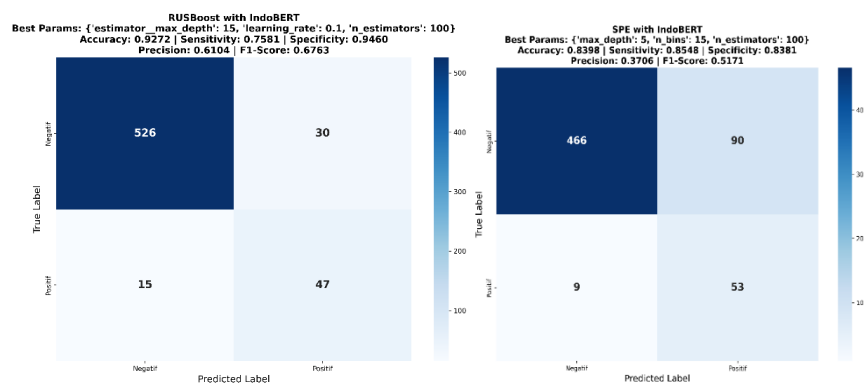


Figure 6. SPE & RUSBoost Confusion Matrix

Figure 6 illustrates the confusion matrix of SPE and RUSBoost models. The confusion matrix analysis reveals the distinct characteristics of both models. SPE successfully classified 53 out of 62 positive samples (85.48%) correctly but only 466 out of 556 negative samples (83.81%) were accurately identified. There were 90 negative samples incorrectly classified as positive (false positives). RUSBoost successfully classified 526 out of 556 negative samples (94.60%) correctly, as well as 47 out of 62 positive samples (75.81%). There were 15 positive samples incorrectly classified as negative (false negatives). RUSBoost demonstrates better balance with only 30 false positives and 15 false negatives. In contrast, SPE has much higher false positives (90 samples)

although its false negatives are lower (9 samples). This indicates that SPE is more "aggressive" in classifying samples as positive, while RUSBoost is more "conservative" and balanced [12].

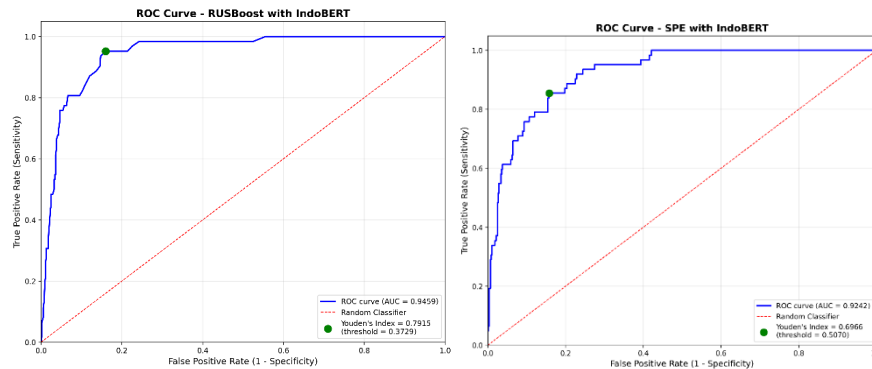


Figure 7. SPE & RUSBoost ROC Curve

Figure 7 illustrates the ROC curves, demonstrating the discriminative ability of both models. RUSBoost achieved an AUC-ROC of 0.9459, higher than SPE's AUC-ROC of 0.9242. The difference of 0.0217 indicates that RUSBoost has a better ability to distinguish between positive and negative classes across various decision thresholds. Both models demonstrate strong performance with AUC-ROC values above 0.92, indicating that the combination of IndoBERT features with ensemble methods can capture patterns effectively [9]. Notably, standard Gradient Boosting achieved an AUC-ROC of 0.9424, closely approaching RUSBoost's performance, further confirming the high quality of the extracted IndoBERT features.

Youden's Index was used to determine the optimal threshold that maximizes the balance between sensitivity and specificity [24].

Table 8. Optimal Threshold Comparison

Metric	SPE	RUSBoost	Logistic Reg.	Gradient Boosting
Youden's Index	0.6966	0.7915	0.6281	0.7828
Optimal Threshold	0.5070	0.3729	0.0005	0.0833
Sensitivity at Optimal	0.8548	0.9516	0.8871	0.8871
Specificity at Optimal	0.8417	0.8399	0.7410	0.8957
F1-Score at Optimal	0.5222	0.5619	0.4215	0.6286

Table 8 shows that RUSBoost requires a substantially lower threshold (0.3729) compared to SPE (0.5070) to achieve optimal balance, reflecting systematic underestimation of positive-class probabilities due to random undersampling during training. Applying this optimal threshold to RUSBoost dramatically increases Sensitivity from 0.7581 to 0.9516 and improves Balanced Accuracy from 0.8521 to 0.8958, though at the cost of a reduced F1-Score (0.5619 vs. 0.6763 at the default threshold). SPE, in contrast, achieves its optimal performance near the default threshold (0.5070), indicating better intrinsic probability calibration. In practical insecure detection applications, threshold selection should be guided by deployment priorities: a lower threshold is preferred when minimizing missed detections (false negatives) is critical, while a higher threshold is appropriate when minimizing false alarms is the priority.

3.7. Discussion

3.7.1. RUSBoost's Superiority in Imbalanced Text Classification

The research results demonstrate that RUSBoost consistently outperforms SPE across most evaluation metrics under extreme class imbalance ($IR = 1:9.04$). RUSBoost achieved Balanced Accuracy of 0.8521, F1-Score of 0.6763, and MCC of 0.6405, compared to SPE's 0.8465, 0.5171, and 0.4937 respectively. This superiority is attributable to three main factors. First, RUSBoost combines random undersampling with boosting, enabling iterative focus on difficult minority samples while partially recovering information lost during undersampling across subsequent iterations [12]. Second, the optimal parameters ($n_estimators=100$, $learning_rate=0.10$,

max_depth=15) reveal that under extreme imbalance, higher model complexity is beneficial rather than detrimental, as greater representational capacity is required to capture the sparse minority class. Third, RUSBoost's substantially higher specificity (0.9460 vs. 0.8381 for SPE) reduces false alarm rates, an important property for insecure detection applications where unnecessary interventions can have adverse consequences.

3.7.2. Characteristics of Self-Paced Ensemble on Text Data

SPE demonstrates a clear advantage in recall (0.8548 vs. 0.7581 for RUSBoost), reflecting greater sensitivity in detecting insecure expressions. This aligns with the self-paced undersampling mechanism that progressively concentrates on difficult minority samples [9]. However, this recall advantage is accompanied by substantially lower precision (0.3706 vs. 0.6104), yielding a significantly lower F1-Score (0.5171 vs. 0.6763). From a computational perspective, SPE's training time of 52.36 seconds is notably shorter than RUSBoost's 76.34 seconds, a difference of approximately 24 seconds (around 46%) which is practically meaningful for frequent retraining scenarios and represents a clearer trade-off than previously characterized in the literature.

3.7.3 Effectiveness of IndoBERT Features

The use of 768-dimensional IndoBERT features proved effective in representing informal Indonesian social media text. High AUC-ROC values across all models 0.9242 (SPE), 0.9459 (RUSBoost), 0.8890 (Logistic Regression), and 0.9424 (Gradient Boosting) indicate that IndoBERT's contextual semantic representations can meaningfully discriminate between insecure and non-insecure expressions even under extreme class imbalance. Feature normalization using StandardScaler also contributed to training stability across all model types.

3.7.4. Baseline Comparison

The inclusion of Logistic Regression and Gradient Boosting as baselines is essential for assessing the true added value of specialized ensemble methods. Logistic Regression with class weighting achieves a competitive Balanced Accuracy of 0.7866 in under one second, establishing a strong efficiency-performance baseline. Gradient Boosting surpasses both SPE and RUSBoost on AUC-PR (0.7100), Specificity (0.9622), and Precision (0.6316), though its Balanced Accuracy (0.7714) and Recall (0.5806) are inferior confirming that without explicit imbalance handling, standard methods cannot reliably detect the minority class. These findings confirm that RUSBoost's advantage over baselines is most pronounced on metrics directly measuring minority-class detection capability (Recall, Balanced Accuracy, MCC, F1-Score), validating the value of combining IndoBERT features with RUSBoost for extreme-imbalance insecure detection.

3.7.5. Qualitative Analysis of Classification Behavior

Given that the dataset consists of informal Indonesian social media language and labels were generated automatically by an AI model rather than domain experts, a qualitative analysis is presented to assess the face validity of classification labels and provide interpretive context for the quantitative results. Table 9 presents representative examples of positive (insecure) and negative (non-insecure) tweets drawn from the dataset.

Table 9. Qualitative Examples of Positive and Negative Class Tweets

No.	Tweet Example (Indonesian)	Label
1	jujur gua agak minder sebenarnya udah 21 th berasa telat bgt jalan hidup gua... tmn tmn gua pada otw semester tua gua masih pertengahan... gua masih perjuangin s1	Negative
2	ini ngomong kaya gini solely karena minder tapi pengen bisa is this a step towards ikhtiar we will never know	Negative
3	liat IG krn bosen akhirnya malah liat pencapaian2 temen2 jadi makin minder bener mending nonton reels yt aja	Negative
4	Gak pake sec acc. Sering lihat highlight sendiri. Kyak.. yah.. ternyata aku cantik juga kok. Knp harus minder pd orang lain.	Positive

Tweets labeled as Positive (Insecure) consistently contain first-person expressions of self-doubt and social comparison. Tweet 4, while framed as a rhetorical question ("Knp harus minder"), represents an introspective reflection on overcoming insecurity, aligning with the positive classification. Tweets labeled as Negative (Non-insecure) exhibit more diverse patterns. Tweets 1,

2, and 3 contain first-person expressions of insecurity but were classified as negative by the model. This may reflect the model's sensitivity to linguistic cues such as self-reflection versus present emotional distress, or subtle contextual differences that are not immediately apparent to human annotators.

This analysis reveals that the distinction between positive and negative classes is subtle and context dependent. The models must therefore rely on IndoBERT's contextual embeddings to capture sentence-level semantics beyond simple keyword matching. This explains the relatively high AUC-ROC values despite the challenging class imbalance. It is important to acknowledge, however, that since labeling was performed automatically without expert validation, some ambiguous boundary cases may exist in the dataset, and the clinical validity of the positive class as a genuine psychological signal of insecurity cannot be fully confirmed without expert annotation.

3.7.6. Practical Implications for Insecurity Detection

1. IndoBERT- RUSBoost with optimal parameters (`n_estimators=100`, `learning_rate=0.10`, `estimator__max_depth=15`) is recommended as the primary model for insecurity detection, providing the best overall balance between precision and recall (F1-Score: 0.6763, Balanced Accuracy: 0.8521).
2. RUSBoost's optimal threshold (0.3729) substantially improves Sensitivity to 0.9516 with Balanced Accuracy of 0.8958, at the expense of reduced F1-Score. Applications prioritizing early detection of at-risk users should employ the lower threshold, while applications targeting high-confidence flagging should use a higher threshold.
3. When retraining frequency or resource constraints are primary considerations, SPE (52.36s) offers a faster alternative to RUSBoost (76.34s) with only a marginal Balanced Accuracy difference (0.8465 vs. 0.8521). For highly resource-constrained scenarios, Logistic Regression (0.53s) remains a viable and competitive baseline.

3.8. Research Limitations

This study has several limitations that should be noted:

1. Data labeling was performed automatically using a fine-tuned IndoBERT model. Labels reflect AI-learned linguistic patterns from training data rather than clinical or psychological ground truth assessments.
2. Engineered Imbalance Ratio: The extreme imbalance (1:9.04) was achieved through deliberate resampling of the original near-balanced dataset (IR: 1:1.16).
3. Domain Specificity: Models were trained on Indonesian social media text relating to insecurity expressed through specific keywords.
4. Parameter Search Space: Grid search covered 18 combinations for RUSBoost and 27 for SPE; other hyperparameter configurations or search strategies (e.g., Bayesian optimization) might yield different or improved results.

4. CONCLUSION

This study successfully compared the performance of four classification models IndoBERT-SPE, IndoBERT- RUSBoost, IndoBERT-Logistic Regression, and IndoBERT-Gradient Boosting for imbalanced text classification in insecurity detection on social media X under an extreme imbalance ratio of 1:9.04. The inclusion of baseline comparators (Logistic Regression and Gradient Boosting) provides a more rigorous empirical comparison than a two-model analysis alone. The experimental results demonstrate that RUSBoost outperforms all other models across the majority of key evaluation metrics, achieving Balanced Accuracy of 0.8521, F1-Score of 0.6763, AUC-ROC of 0.9459, and Youden's Index of 0.7915. SPE shows a competitive advantage solely in Recall (0.8548), making it suitable for recall-critical applications where minimizing missed detections is the primary objective. The optimal parameters for RUSBoost were `n_estimators=100`, `learning_rate=0.10`, and `estimator__max_depth=15`, with the optimal operational threshold at 0.3729 yielding Balanced Accuracy of 0.8958 and Sensitivity of 0.9516. High AUC-ROC values across all models (above 0.88) confirm that IndoBERT features effectively capture semantic and contextual nuances of informal Indonesian social media text. Qualitative analysis reveals that the core classification challenge lies in distinguishing genuine first-person expressions

of insecurity from figurative or rhetorical uses of insecurity-related vocabulary a pragmatic distinction well-suited to IndoBERT's contextual embedding representations.

ACKNOWLEDGEMENTS

This study was financially supported by the Research Grant of the Institute for Research and Community Service (LPPM), Universitas Terbuka in 2025.

CONFLICT OF INTEREST

The authors declare that they have no conflict of interest related to this study.

DATA AVAILABILITY

The data used in this study are not publicly available due to privacy and platform restrictions, but supporting materials may be obtained from the corresponding author upon reasonable request.

AUTHOR CONTRIBUTIONS

Emeylia Safitri: Conceptualization; Data Curation; Methodology; Writing-Original Draft; Funding acquisition. **I Gusti Ngurah Sentana Putra:** Methodology; Software; Supervision; Writing-Review & Editing. **Nuramaliyah:** Supervision; Validation; Writing-Review & Editing. **Ika Nur Laily Fitriana:** Project Administration; Formal Analysis; Writing-Review & Editing. **Ria Faulina:** Resources; Writing-Review & Editing.

REFERENCES

- [1] K. Burghardt, D. M. T. Fessler, C. Tang, A. Pisor, and K. Lerman, "Posts of Peril: Detecting Information About Hazards in Text," 2025, [Online]. Available: <http://arxiv.org/abs/2405.17838>
- [2] R. Sazama, A. Muttaqin, R. Ananda, R. Julianansa, M. Ula, and M. Eng, "Klasifikasi Cyberbullying Di Twitter Menggunakan Metode Gradient Boosting," *SENASTIKA Univ. Malikussaleh*, pp. 1–9, 2024, [Online]. Available: <https://proceedings.unimal.ac.id/senastika/issue/view/15>
- [3] S. Nouas, L. Oukid, and F. Boumahdi, "Enhancing imbalanced text classification: an overlap-based refinement approach," *Data Sci. Manag.*, vol. 8, no. 4, pp. 474–484, 2025, doi: 10.1016/j.dsm.2025.03.001.
- [4] D. O. Putri and A. Qoiriah, "Klasifikasi Cyberbullying Pengguna Twitter Menggunakan Random Forest Dengan Penyematan Kata FastText," *J. Informatics Comput. Sci.*, vol. 6, no. 01, pp. 265–272, 2024, doi: 10.26740/jinacs.v6n01.p265-272.
- [5] N. Abdulloh and A. F. Hidayatullah, "Deteksi Cyberbullying pada Cuitan Media Sosial Twitter," *Automata*, vol. Vol 1, no. 1, pp. 1–5, 2021.
- [6] S. F. Taskiran, B. Turkoglu, E. Kaya, and T. Asuroglu, "A comprehensive evaluation of oversampling techniques for enhancing text classification performance," *Sci. Rep.*, vol. 15, no. 1, pp. 1–21, 2025, doi: 10.1038/s41598-025-05791-7.
- [7] A. Nawaz, A. Ahmad, and S. S. Khan, "Beyond Rebalancing: Benchmarking Binary Classifiers Under Class Imbalance Without Rebalancing Techniques," 2025, [Online]. Available: <http://arxiv.org/abs/2509.07605>
- [8] I. G. N. Sentana Putra, M. Yusran, J. R. Sari, C. Suhaeni, B. Sartono, and G. A. Dito, "Evaluating Fasttext and Glove Embeddings for Sentiment Analysis of AI-Generated Ghibli-Style Images," *J. Appl. Informatics Comput.*, vol. 9, no. 5, pp. 2574–2587, 2025, doi: 10.30871/jaic.v9i5.10600.
- [9] Z. Liu, W. Cao, Z. Gao, J. Bian, H. Chen, Y. Chang, and T.-Y. Liu, "Self-paced ensemble for highly imbalanced massive data classification," *Proc. 2020 IEEE 36th Int. Conf. Data Eng. (ICDE)*, pp. 841–852, 2020, doi: 10.1109/ICDE48307.2020.00078.
- [10] S. Bano, W. Zhi, B. Qiu, M. Raza, N. Sehito, M. M. Kamal, G. Aldehim, and N. Alruwais, "Self-paced ensemble and big data identification: a classification of substantial imbalance computational analysis," *J. Supercomput.*, vol. 80, no. 7, pp. 9848–9869, 2024, doi:

- 10.1007/s11227-023-05828-6.
- [11] N. C. Ristea and R. T. Ionescu, "Self-paced ensemble learning for speech and audio classification," *Proc. Annu. Conf. Int. Speech Commun. Assoc. INTERSPEECH*, vol. 2, pp. 1276–1280, 2021, doi: 10.21437/Interspeech.2021-155.
 - [12] C. Seiffert, T. M. Khoshgoftaar, J. Van Hulse, and A. Napolitano, "RUSBoost: A hybrid approach to alleviating class imbalance," *IEEE Trans. Syst. Man, Cybern. Part A Systems Humans*, vol. 40, no. 1, pp. 185–197, 2010, doi: 10.1109/TSMCA.2009.2029559.
 - [13] T. Fulazzaky, A. Saefuddin, and A. M. Soleh, "Evaluating Ensemble Learning Techniques for Class Imbalance in Machine Learning: A Comparative Analysis of Balanced Random Forest, SMOTE-RF, SMOTEBoost, and RUSBoost," *Sci. J. Informatics*, vol. 11, no. 4, pp. 969–980, 2024, doi: 10.15294/sji.v11i4.15937.
 - [14] Y. Permatasari, "PENANGANAN MASALAH KELAS TIDAK SEIMBANG DENGAN RUSBOOST DAN UNDERBAGGING (STUDI KASUS: MAHASISWA DROP OUT SPs IPB PROGRAM MAGISTER)," *Thesis*, 2016, [Online]. Available: <http://repository.ipb.ac.id/handle/123456789/80118>
 - [15] N. A. Setiabudi, B. Sartono, U. D. Syafitri, and K. B. Aryasa, "Comparing Self-Paced Ensemble and RUSBoost for Imbalanced Poverty Classification in West Java," *Indones. J. Stat. Its Appl.*, vol. 9, no. 2, pp. 218–229, 2025, doi: 10.29244/ijsa.v9i2p218-229.
 - [16] H. Chen and M. Ji, "Experimental Comparison of Classification Methods under Class Imbalance," *EAI Endorsed Trans. Scalable Inf. Syst.*, vol. 8, no. 33, pp. 1–20, 2021, doi: 10.4108/EAI.11-6-2021.170234.
 - [17] N. Putu, D. Agustina, I. D. Ayu, P. Pratiwi, I. G. Ngurah, and L. Wijayakusuma, "Public Sentiment Analysis on Demonstration Actions Using IndoBERT Based on Transfer Learning," vol. 9, no. 6, 2026.
 - [18] A. A. Qolbu, N. Fitriyati, and N. Inayah, "Performa Naïve Bayes , SVM , dan IndoBERT pada Analisis Sentimen Twitter IndiHome dengan Strategi Penanganan Data Tidak Seimbang," *J. Fourier*, vol. 814, no. 1, pp. 29–44, 2025, doi: 10.14421/fourier.2025.141.29-44.
 - [19] M. G. Al-Kadzim, R. Rasim, and H. Herbert, "Analisis Perubahan Sentimen Publik di Media Sosial X terhadap Konflik Palestina-Israel Menggunakan Model IndoBERT," *Digit. Transform. Technol.*, vol. 4, no. 2, pp. 1167–1174, 2025, doi: 10.47709/digitech.v4i2.5312.
 - [20] A. F. Abdillah, C. B. P. Putra, A. Aprianoni, S. Juanita, and D. Purwitasari, "Ensemble-based Methods for Multi-label Classification on Biomedical Question-Answer Data," *J. Inf. Syst. Eng. Bus. Intell.*, vol. 8, no. 1, pp. 42–50, 2022, doi: 10.20473/jisebi.8.1.42-50.
 - [21] M. Anusha, "Hybrid Approach to Fake News Detection: Leveraging BERT-Based Model with Word Embedding Features for Sentiment Classification in Social Media Content," *J. Electr. Syst.*, vol. 20, no. 11s, pp. 1044–1060, 2024, doi: 10.52783/jes.7363.
 - [22] M. F. Arifah, Irnawati, Ruslin, K. Nisa, A. Windarsih, and A. Rohman, "The Application of FTIR Spectroscopy and Chemometrics for the Authentication Analysis of Horse Milk," *Int. J. Food Sci.*, vol. 2022, 2022, doi: 10.1155/2022/7643959.
 - [23] Y. K. Sari, J. Iskandar, and A. Prasetya, "Penerapan Metode Naive Bayes Dalam Analisis Sentimen Terhadap Cyberbullying," vol. 5, no. 1, 2026.
 - [24] M. Hassanzad and K. Hajian-Tilaki, "Methods of determining optimal cut-point of diagnostic biomarkers with application of clinical data in ROC analysis: an update review," *BMC Med. Res. Methodol.*, vol. 24, no. 1, pp. 1–14, 2024, doi: 10.1186/s12874-024-02198-2.
 - [25] D. Eko Putro, D. Juarsa, B. Putra Hermiana, B. Bagastian, and H. Sulistiani, "Analisis Sentimen Publik terhadap 'Save Raja Ampat' di Media Sosial Menggunakan Model IndoBERT," *Bull. Comput. Sci. Res.*, vol. 5, no. 5, pp. 1067–1075, 2025, doi: 10.47065/bulletincsr.v5i5.621.
 - [26] C. H. Miranda, G. Sanchez-Torres, and D. Salcedo, "Exploring the Evolution of Sentiment in Spanish Pandemic Tweets: A Data Analysis Based on a Fine-Tuned BERT Architecture," *Data*, vol. 8, no. 6, 2023, doi: 10.3390/data8060096.