OPEN ACCESS 136

ISSN 2356-5462
http://socj.telkomuniversity.ac.id/ijoict/

Intl. Journal on ICT
Vol. 11, No. 2, Dec 2025. pp. 136-149
doi:10.21108/ijoict.v11i2.9513

A Comparative Study on Handling Imbalanced Data in Indonesian Hate Speech Detection Using FastText and BiLSTM

Akmal Muhamad Faza[1*], Yuliant Sibaroni[2], Sri Suryani Prasetyowati[3]

[1,2,3]*School of Computing, Telkom University*
Bandung, Indonesia

[*]akmalfaza@student.telkomuniversity.ac.id

Abstract

Online hate speech poses a significant threat to social harmony in Indonesia, necessitating effective automated detection systems. This study addresses the challenge of data imbalance, a common issue in hate speech datasets, by developing a Bidirectional Long Short-Term Memory (BiLSTM) model with FastText word embeddings. We systematically compare three oversampling techniques—Random Oversampler, SMOTE, and ADASYN—across varying degrees of imbalance in the Indonesian Hate Speech Superset dataset (14,306 comments), a gap in existing literature. Evaluated using Stratified K-fold Cross-Validation with Accuracy, Precision, Recall, and F1-score, our results indicate that oversampling generally enhances model performance, particularly for the minority class. The optimal oversampling strategy depends on imbalance severity: SMOTE achieved the best balance trade-off within Recall (78.9%) and F1-score (75.3%) on the original dataset, while Random Oversampling was superior for severely imbalanced scenarios, reaching F1-scores of 60.6% (30% minority) and 38.6% (10% minority). These findings offer vital insights for building more adaptive hate speech classification systems in the Indonesian context with imbalanced data distribution.

Keywords: BiLSTM, Deep Learning, FastText, Hate Speech, Imbalanced, Oversampling

I. INTRODUCTION

THE rapid development of technology has greatly facilitated public access to social media. In recent years, the number of social media users in Indonesia has continuously increased, with Twitter (now X), YouTube, and Instagram being widely utilized platforms [1][2][3]. The large number of social media users often leads to uncontrolled communication, where many individuals use harsh language or hate speech [4]. The presence of social media has transformed what was originally freedom of expression into freedom to hate. Therefore, to address this issue, it is crucial to develop classification systems capable of automatically and adaptively detecting hate speech [5][6]. However, one of the primary challenges in developing such systems is data imbalance, where the volume of hate speech data (minority class) is significantly smaller than that of non-hate speech data (majority class) [7][4]. This data imbalance can cause classification models to be biased towards the majority class, potentially failing to detect the true context of texts that contain hate speech.

Various previous studies have proposed deep learning models to address this problem of hate speech detection. In this research, the Bidirectional Long Short-Term Memory (BiLSTM) model was chosen due to its ability to capture word sequence context in two directions [8][9][10][11], which is crucial for recognizing the complex textual meaning of hate speech. Furthermore, FastText word embeddings are utilized because of their capability to leverage sub-word information (n-grams) [12][13][14][15], making them effective in handling non-standard vocabulary in the Indonesian language [16].

Received on 9 Jul 2025. Revised on 29 Jul 2025. Accepted and Published on 2 Dec 2025.

To tackle the data imbalance challenge, this study implements and compares three oversampling methods with distinct characteristics. Random Oversampling works by randomly duplicating minority class data [7]; this simple approach serves as a baseline for comparison. SMOTE (Synthetic Minority Over-sampling Technique) generates synthetic data based on the nearest neighbors of minority samples [7], while ADASYN (Adaptive Synthetic Sampling) is more adaptive as it focuses on minority samples that are harder for the model to learn [17][18]. While individual studies have explored these components, a systematic and comparative analysis of how these specific oversampling techniques perform across varying degrees of data imbalance, when integrated with a BiLSTM-FastText architecture for Indonesian hate speech detection, and crucially, an explicit comparison of their performance outcomes, remains an underexplored area in the current literature.

The dataset used in this study is the Indonesian Hate Speech Superset, a publicly accessible dataset available on the Hugging Face platform [19]. This dataset consists of 14,306 Indonesian social media comments that have been binary-labeled as hate speech (label 1) and non-hate speech (label 0), collected from social media platforms Twitter (X), YouTube, and Instagram [19][20]. This dataset was chosen for its large scale, open access, and its representative of hate speech forms commonly found on Indonesian social media.

II. LITERATURE REVIEW

Research in automated hate speech detection has advanced considerably, driven by the increasing volume of harmful content on social media platforms. This section reviews the relevant literature, focusing on the evolution of deep learning models, effective text representation techniques, and strategies for handling data imbalance, culminating in a summary of key related works. Early approaches to hate speech detection often employed traditional machine learning algorithms or simpler neural network architectures. For instance, initial studies on Indonesian text classification for abusive language utilized models like Long Short-Term Memory (LSTM) [6]. However, the sequential nature of language often benefits from processing context from both directions. Subsequent research consistently demonstrated that Bidirectional LSTM (BiLSTM) models deliver superior performance by processing text sequences both forward and backward, thereby capturing a richer semantic context and long-range dependencies within a sentence [8][9]. This advantage of the bidirectional approach has been validated in multiple recent studies on Indonesian hate speech detection [8][9][11]. Further advancements in natural language processing (NLP) have seen the rise of Transformer models, such as BERT and its variants, which have achieved state-of-the-art performance across various tasks. For Indonesian, models like IndoBERTweet have emerged as powerful benchmarks for complex, multi-label hate speech tasks, achieving high precision by leveraging large-scale pre-training on Indonesian social media data [10]. While Transformer-based models offer cutting-edge performance, this study focuses on a systematic comparison of oversampling techniques within a BiLSTM-FastText framework, aiming to provide a foundational understanding of data balancing strategies before exploring more computationally intensive architectures.

The efficacy of neural models in text classification is heavily reliant on the quality of their text representation. For morphologically rich languages like Indonesian, traditional one-hot encoding or bag-of-words models often fall short in capturing semantic meanings. Word embeddings provide dense vector representations that encode semantic and syntactic relationships between words. FastText, developed by Facebook AI, has emerged as a particularly effective word embedding technique [12][15]. Unlike methods like Word2Vec which treat each word as an atomic unit, FastText represents a word as a sum of its character n-gram vectors. For instance, the word "apple" with n=3 would be defined by the n-grams <ap, app, ppl, ple, le> and the full word token <apple>. This capacity to learn vectors from sub-word units enables FastText to effectively manage out-of-vocabulary (OOV) words and capture morphological nuances, which is a frequent issue in informal social media text [12], [13][14]. Multiple studies confirm that the use of FastText leads to significant improvements in hate speech classification accuracy [12][13]. Its versatility is further demonstrated through successful applications with other architectures, such as Convolutional and Recurrent Neural Networks (CNN and RNN), for sentiment analysis [14]. The availability of pre-trained multilingual embeddings that include Indonesian has also provided a robust foundation for various Natural Language Processing tasks in the region [16].

A critical and persistent challenge in hate speech detection is the problem of data imbalance, where non-offensive content significantly outnumbers hateful content [4][7]. This imbalance can lead to models that are biased towards the majority class, resulting in poor detection rates for the critical minority (hate speech) class,

characterized by low Recall and F1-scores. To address this, researchers employ various oversampling strategies that aim to balance the class distribution in the training data. Random Oversampling is a straightforward method that works by randomly duplicating samples from the minority class until its size is balanced with the majority class [7]. While simple, its effectiveness often serves as a baseline for comparison with more sophisticated techniques. Synthetic Minority Over-sampling Technique (SMOTE) is a foundational approach that creates new, synthetic data points by interpolating between existing minority class instances and their nearest neighbors [7]. SMOTE generates "new" samples that are not exact duplicates but are within the feature space of the minority class, thus expanding the decision boundary. Adaptive Synthetic Sampling (ADASYN) is an advanced variant of SMOTE. This method adaptively generates more synthetic samples for minority class instances that are harder to learn (i.e., those near the decision boundary), focusing on areas where the model struggles most [17]. Its evolution and diverse applications are well-documented in recent comprehensive surveys [17].

In a practical application, Wenando et al. [18] demonstrated ADASYN's ability to improve model sensitivity on Indonesian social media data when used with an LSTM model. Furthermore, the success of hate speech detection systems is heavily influenced by dataset quality. Researchers highlight the importance of minimizing cultural and geographic biases in the annotation process, particularly for non-English languages [20]. Despite advancements in models and techniques, the development of high-quality, representative datasets for Indonesian hate speech remains a significant challenge, with persistent issues in class distribution and the capture of local context [4]. This has prompted ongoing research, including the recent creation of new benchmark datasets for Indonesian hate speech, such as the Indonesian Hate Speech Superset used in this study [19].

To provide a clearer overview of existing research and contextualize the contribution of this study, TABLE I summarizes key findings from relevant literature on Indonesian hate speech detection and related NLP tasks. Although previous studies have individually affirmed the effectiveness of BiLSTM models, FastText, and various oversampling methods, there remains a significant gap in the literature regarding a direct and systematic comparison of these specific oversampling techniques within a unified BiLSTM-FastText framework for Indonesian hate speech detection, particularly across varying degrees of data imbalance. This research directly addresses that gap by systematically evaluating the impact of Random Oversampling, SMOTE, and ADASYN, aiming to clarify the most effective strategy for managing imbalanced data in this specific linguistic context.

TABLE I
SUMMARY OF RELATED WORKS IN INDONESIAN HATE SPEECH

Author(s)	Methodology	Key Findings	Limitations	Research Gap
Sihombing et al	BiLSTM	BiLSTM superior for Indo HS.	No imbalance handling.	Systematic oversampling with BiLSTM.
Fajri et al	IndoBERTweet + BiLSTM	High performance with hybrid Transformer-BiLSTM.	High computational cost.	Detailed oversampling in non-Transformer.
Kovács	Various Oversampling techniques	Comprehensive empirical oversampling comparison.	General datasets; not specific to Indonesian HS	Oversampling application in Indo HS with BiLSTM-FastText.
Wenando et al	ADASYN + LSTM	ADASYN improved sensitivity on Indo social media.	Unidirectional LSTM; limited comparison.	BiLSTM integration, systematic multi-oversampling comparison.
Cahyaningtyas et al	Dataset Creation	New benchmark dataset for Indo HS.	Dataset creation focus; no model application	Model application & imbalance handling on this dataset.

III. RESEARCH METHOD

A. System Design

This section outlines the systematic methodology employed to develop and evaluate the Indonesian hate speech detection system. The overall workflow, from initial data processing to final model assessment, is comprehensively illustrated in Fig. 1.

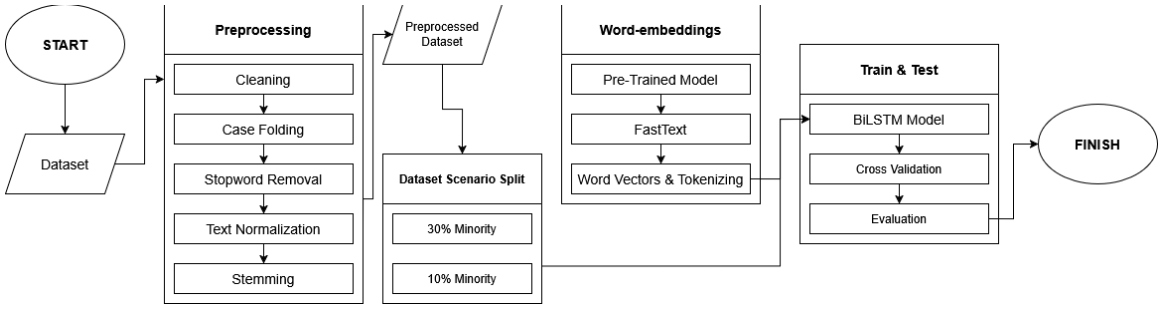

Fig. 1. The proposed system architecture.

The proposed system follows a structured pipeline typical of machine learning projects. The process begins with importing the necessary libraries and the dataset. The raw data then undergoes a comprehensive preprocessing stage, which includes five steps: cleaning, case folding, stopword removal, text normalization, and stemming. The resulting clean data is then used in two parallel processes: word embedding and model training. For word embeddings, a pre-trained FastText model is used to perform tokenization and generate word vectors. These vectors serve as the input for the BiLSTM model during the training and testing phase. To ensure robust evaluation without manual data splitting, the study employs Stratified K-Fold Cross-Validation. Within each fold of the cross-validation, oversampling techniques are applied to the training set to address class imbalance before the model is trained and evaluated.

B. Data Collection & Experimental Setup

The data used in this study is a publicly available dataset sourced from the Hugging Face platform, known as the Indonesian Hate Speech Superset [19]. It is a superset composed of social media comments that have been annotated with a binary label: 1 for hate speech and 0 for non-hate speech.

As shown in the data distribution in Fig. 2, the dataset is inherently imbalanced. It contains 8,237 samples of non-hate speech (the majority class) and 5,934 samples of hate speech (the minority class). This characteristic presents a common challenge in text classification, as it can lead to a model that is biased towards the majority class. Therefore, specific techniques must be applied to ensure fair and effective training [7][17][20].

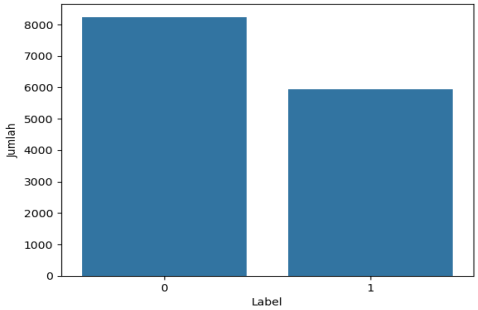

Fig. 1. Label distribution of the original dataset.

The initial dataset contained 14,306 social media comments. To rigorously evaluate the robustness of the models against varying degrees of class imbalance, this study was conducted across three distinct dataset scenarios:

1) Original Dataset: The dataset after initial cleaning (14,171 samples), where oversampling is applied to the original class distribution.
2) 30% Minority Scenario: A more imbalanced version where the minority class (hate speech) was first down sampled to constitute 30% of the majority class's size before the oversampling techniques were applied.
3) 10% Minority Scenario: A severely imbalanced version where the minority class was downsampled to just 10% of the majority class's size.

C. *Data Preprocessing*

Before the text data could be used for model training, it underwent a critical preprocessing phase to ensure quality and consistency[21]. This process involved the following five steps to clean, normalize, and structure the data for the BiLSTM architecture:

1) Cleaning: This step removes punctuation, retweet symbols (RT), URLs, regular expressions, and emoticons from the text.
2) Case Folding: All text is converted to lowercase to ensure that the same word with different capitalization (e.g., "Hate" and "hate") is treated as a single token, reducing vocabulary size and improving consistency.
3) Stopword Removal: Common words that appear frequently but offer little distinguishing value (e.g., "yang," "dan," "di") are removed. This process is augmented with an external stopword list (stopwords_extend) to create a more comprehensive corpus of words to be excluded.
4) Text Normalization: Informal words, slang, abbreviations, and typos are converted into their standard Indonesian forms. This is vital as non-standard language is prevalent in social media. The process utilizes a pre-existing slang dictionary (new_kamusalay) from previous research.
5) Stemming: Words are reduced to their root or base form (e.g., "berbicara" becomes "bicara," "kebencian" becomes "benci"). This is performed using the Sastrawi library for the Indonesian language

D. *Bidirectional Long Short-Term Memory*

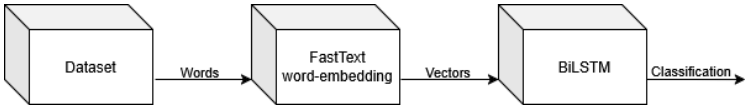

Fig. 2. Workflow of the BiLSTM model for text classification.

The core of the detection system is a Bidirectional Long Short-Term Memory (BiLSTM) model. This architecture is a type of Recurrent Neural Network (RNN) specifically designed for processing sequential data like text. It is highly effective for hate speech classification because it is well-suited at learning long-range dependencies within a sentence and understanding complex contextual nuances, capabilities that are crucial for distinguishing between harmless and hateful language [8][9][10].

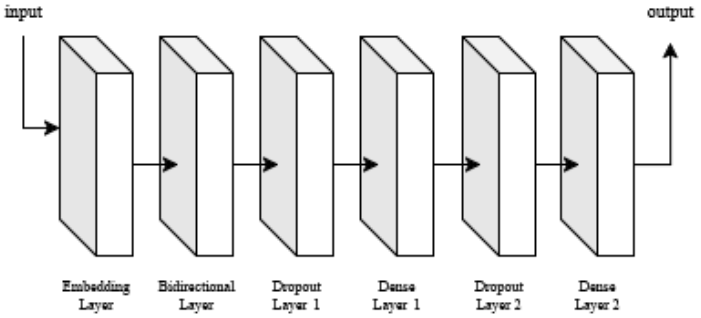

Fig. 4. Architecture of the BiLSTM model for hate speech detection.

The model's architecture, depicted in Fig. 4, is constructed from a series of layers, each with a distinct purpose:

1) *Input Layer:* This is the initial entry point for the data. It takes the preprocessed text, which has been converted into sequences of numerical tokens, as its input to the model.
2) *Embedding Layer:* This crucial layer is responsible for converting the numerical tokens into dense, meaningful vector representations. It utilizes the pre-trained FastText embedding matrix of size (13624, 300). By doing so, the model leverages rich semantic information learned from a vast corpus, which is especially effective for handling the complex morphology and out-of-vocabulary words found in Indonesian social media text.
3) *Bidirectional Layer:* As the core of the architecture, this layer processes the text sequence simultaneously from both left-to-right and right-to-left. By concatenating the outputs of these forward and backward passes, the BiLSTM layer provides a rich, bidirectional context for every word in the sequence. This comprehensive understanding is crucial for deciphering complex linguistic patterns, for which the layer utilizes 128 LSTM units. An LSTM unit's ability to learn long-term dependencies stems from its internal "gates" that control the flow of information into and out of the cell state. For a given time step t, with input x_t and previous hidden state h_{t-1}, the operations are defined as follows:
    a) *Forget Gate (f_t):* This gate determines what information from the previous cell state C_{t-1} should be forgotten. It takes h_{t-1} and x_t as input and outputs a value between 0 and 1, where 0 means "completely forget" and 1 means "completely keep."

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{1}$$

    b) *Input Gate (i_t):* This gate decides what new information from the current input x_t and h_{t-1} should be stored in the cell state. It has two parts: a sigmoid layer that decides which values to update, and a tanh layer that creates a vector of new candidate values $\tilde{C}_t$.

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{2}$$

$$\tilde{C}_t = tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \tag{3}$$

    c) *Cell State Update (C_t):* The previous cell state C_{t-1} is updated to the new cell state C_t by first multiplying it by the forget gate's output (forgetting old information) and then adding the input gate's output multiplied by the candidate values (adding new information).

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t \tag{4}$$

    d) *Output Gate (o_t):* This gate determines what part of the cell state will be outputted as the new hidden state ht. It takes h_{t-1} and x_t as input, and then multiplies the sigmoid output by the tanh of the new cell state.

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{5}$$

$$h_t = o_t \cdot \tanh(C_t) \tag{6}$$

    e) *Bidirectional Mechanism:* A BiLSTM layer consists of two independent LSTM layers: one processes the input sequence in the forward direction (from left to right), and the other processes it in the backward direction (from right to left). The outputs of these two hidden layers, $\overrightarrow{h_t}$ (forward) and $\overleftarrow{h_t}$(backward), are then concatenated at each time step to form the

final output $h_t = [\overrightarrow{h_t}; \overleftarrow{h_t}]$. This concatenation provides a comprehensive context for each word, considering both its past and future dependencies in the sequence.

4) *Dropout Layer (1):* To combat overfitting, a dropout layer is applied immediately after the BiLSTM layer. It works by randomly setting a fraction of input units to 0 during each training update. With a rate of 0.5, this technique encourages the model to learn more robust and distributed feature representations rather than relying on any single neuron, thereby improving its ability to generalize to new, unseen data.
5) *Dense Layer (Hidden):* This fully-connected hidden layer introduces non-linearity into the model by using the ReLU (Rectified Linear Unit) activation function. By transforming the data non-linearly, this layer enables the model to learn more intricate and abstract patterns from the features extracted by the BiLSTM layer. The ReLU function is defined as:

$$f(x)_{ReLU} = max\ (0, x) \tag{7}$$

6) *Dropout Layer (2):* A second dropout layer, also with a rate of 0.5, is applied after the hidden layer to provide further regularization and reduce the risk of overfitting.
7) *Dense Layer (Output):* The final layer is a dense output layer that condenses the features learned by the preceding layers into a single predictive value. It uses the sigmoid activation function, which is ideal for binary classification as it maps any real-valued number into a probability between 0 and 1. This output directly corresponds to the model's predicted likelihood that the input text is hate speech (approaching 1) or not (approaching 0). The sigmoid function is defined as:

$$f(x)_{sigmoid} = \frac{1}{1+e^{-x}} \tag{8}$$

The key hyperparameters used for training the model are detailed as:

TABLE I
MODEL HYPERPARAMETERS

Hyperparameter	Value
Dropout Rate	0.5
Epochs	10
Batch Size	64

To further prevent overfitting and ensure the model generalizes well, an Early Stopping callback was implemented during the training process. This callback monitored the validation loss at the end of each full pass over the training data (an epoch). If the validation loss did not show improvement (decrease) for three consecutive epochs (patience=3), the training was automatically halted. Furthermore, the restore best weigth parameter was enabled, ensuring that the model's final weights were reverted to those from the epoch with the lowest validation loss.

E. FastText

FastText is a word representation method that considers sub-word units (character n-grams), allowing it to capture the morphological structure of words. Developed by Facebook AI, this approach excels at handling new or rare words, which are common in social media [6], [7], [9]. Unlike methods like Word2Vec which treat each word as an atomic unit, FastText represents a word as a sum of its character n-gram vectors. For instance, the word "apple" with n=3 would be defined by the n-grams <ap, app, ppl, ple, le> and the full word token <apple>.

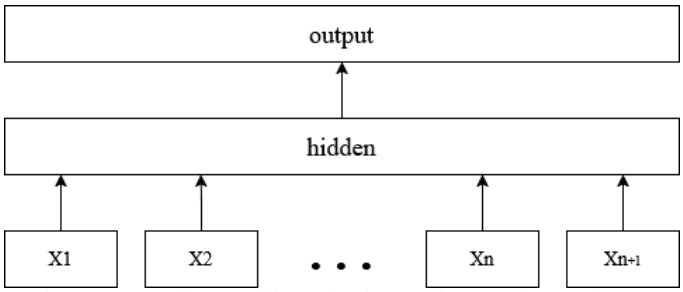

Fig. 5. The FastText architecture, illustrating the use of N-gram features for a sentence.

F. Oversampling Techniques

To address the imbalanced dataset, this research implements and compares three distinct oversampling methods applied to the training data within each cross-validation fold:

1) *Random Oversampler:* This method works by randomly duplicating samples from the minority class until its size is balanced with the majority class [7].
2) *SMOTE (Synthetic Minority Over-sampling Technique):* This technique generates new synthetic data points by interpolating between existing minority class instances and their nearest neighbors [7].
3) *ADASYN (Adaptive Synthetic Sampling):* An advanced variant of SMOTE, this method adaptively generates more synthetic samples for minority class instances that are harder to learn, focusing on the decision boundary [17].
4) *No Oversampling:* Baseline model where the BiLSTM trained to imbalanced dataset without oversampling techniques mentioned above.

The three techniques (Random Oversampling, SMOTE, ADASYN) were applied within a 5-fold Stratified Cross-Validation inside the train and evaluation process to prevent data leakage and biased model. Model performance was evaluated using Accuracy and F1-score, derived from a confusion matrix.

G. Stratified K-Fold Cross-Validation

To obtain a robust and reliable measure of the model's performance, this study uses Stratified K-Fold Cross-Validation. In this method, the dataset is divided into k equally sized "folds," ensuring that the proportion of each class (hate vs. non-hate) is the same in each fold as it is in the original dataset. The training and evaluation process is repeated k times. In each iteration, one fold is used as the validation set, and the remaining k-1 folds are used as the training set. This ensures that every data point is used for validation exactly once. The final performance metrics are calculated by averaging the results from all k iterations.

H. System Performance Evaluation

Model performance is evaluated using a Confusion Matrix, a table that summarizes the classification results. From the matrix, four key metrics are calculated: Accuracy, Precision, Recall, and F1-score.

TABLE II
CONFUSION MATRIX

Confusion Matrix	Predicted Positive	Predicted Negative
Actual Positive	True Positive (TP)	False Negative (FN)
Actual Negative	False Positive (FP)	True Negative (TN)

These metrics are calculated using the following standard equations:

1) *Accuracy:* This metric measures the ratio of all correct predictions (both TP and TN) to the total number of samples. While it provides a general overview of performance, accuracy can be misleading on imbalanced datasets like ours, as a high score can be achieved by simply predicting the majority class. Therefore, it is considered alongside other, more nuanced metrics.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{9}$$

2) *Precision:* Also known as the positive predictive value, precision measures the proportion of comments flagged as hate speech that were *actually* hate speech. A high precision score indicates a low false positive rate, meaning the model is trustworthy when it makes a positive prediction.

$$Precision = \frac{TP}{TP + FP} \tag{10}$$

3) *Recall (Sensitivity):* Also known as sensitivity or the true positive rate, recall measures the model's ability to identify all of the actual hate speech comments within the dataset. High recall is critical for this task, as it signifies that the system is effective at catching hateful content and has a low false negative rate.

$$Recall = \frac{TP}{TP + FN} \tag{11}$$

4) *F1-score:* The F1-score is the harmonic mean of Precision and Recall. It provides a single, balanced measure of a model's performance, which is especially useful when dealing with imbalanced classes. A high F1-score indicates that the model maintains a healthy balance between minimizing false positives (Precision) and minimizing false negatives (Recall).

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} = \frac{2TP}{2TP + FP + FN} \tag{12}$$

IV. RESULTS AND DISCUSSION

In this research, a systematic evaluation was conducted to evaluate the performance of the sentiment analysis model using Bidirectional Long Short-Term Memory (BiLSTM) under varying conditions of class imbalance with FastText as word-embeddings. The evaluation process involved several stages, including preprocessing with techniques such as cleaning, case folding, stopword removal, text normalization, and stemming, with the core of the investigation involved a comparative analysis of three oversampling techniques—Random Oversampling, Synthethic Minority Oversampling Technique (SMOTE), and Adaptive Synthethic Sampling (ADASYN)—across three dataset scenarios to identify the most effective method for handling different levels of imbalanced data. Model validation was carried out using Stratified K-Fold Cross Validation with 5-folds while the oversampling methods are applied to the training dataset for each folds to prevent data leakage and ensure reliable—unbiased result. Additionaly, confusion matrix was conducted after performance evaluation to visualize the impact of oversampling for model to handle imbalanced dataset.

A. Model Performance Evaluation

The experiments were conducted across three scenarios: the original dataset, a dataset with the minority class reduced to 30% of the majority, and a dataset with the minority class reduced to 10%. By using a 5-fold Stratified Cross-Validation approach, the dataset was repeatedly split, with 80% used for training and 20% for validation in each folds. To prevent data leakage and ensure an unbiased evaluation, oversampling techniques were applied to the training data within each fold of the cross-validation, leaving the validation set in its original

distribution. The average Accuracy and F1-score for each technique across all 5-folds are presented in table below:

TABLE III
AVERAGE EVALUATION RESULTS OF THE MODEL WITH EACH OVERSAMPLING TECHNIQUE.

Dataset Scenario	Technique	Recall (%)	F1-score (%)
Original	Random Oversampling	78.2	75.2
	SMOTE	**78.9**	75.3
	ADASYN	73.9	74.7
	No Oversampling	74.3	**75.4**
30% Minority	Random Oversampling	**71.9**	**60.6**
	SMOTE	68.4	57.2
	ADASYN	67.8	56.7
	No Oversampling	49.3	57.2
10% Minority	Random Oversampling	**59.7**	**38.6**
	SMOTE	41.4	26.8
	ADASYN	45.6	29.1
	No Oversampling	19.1	28.6

From the data presented in Table III, several key results can be observed. In the *Original Dataset* scenario, SMOTE achieved the highest Recall (78%), while model with No Oversampling technique recorded the highest F1-score (75.4%). However, as the class imbalance became more severe, a clear shift in performance occurred. Random Oversampling consistently outperformed the No Oversampling and synthethic techniques the more imbalanced scenarios, achieving the highest F1-score in both the *30% Minority* (60%) and *10% Minority* (38%) datasets.

To further illustrate the impact of oversampling on the model's ability to correctly classify both hate speech (minority) and non-hate speech (majority), confusion matrices for the best-performing models in each scenario are presented below. These matrices provide a detailed breakdown of True Positives (TP), False Negatives (FN), False Positives (FP), and True Negatives (TN), offering insights into the model's sensitivity and specificity.

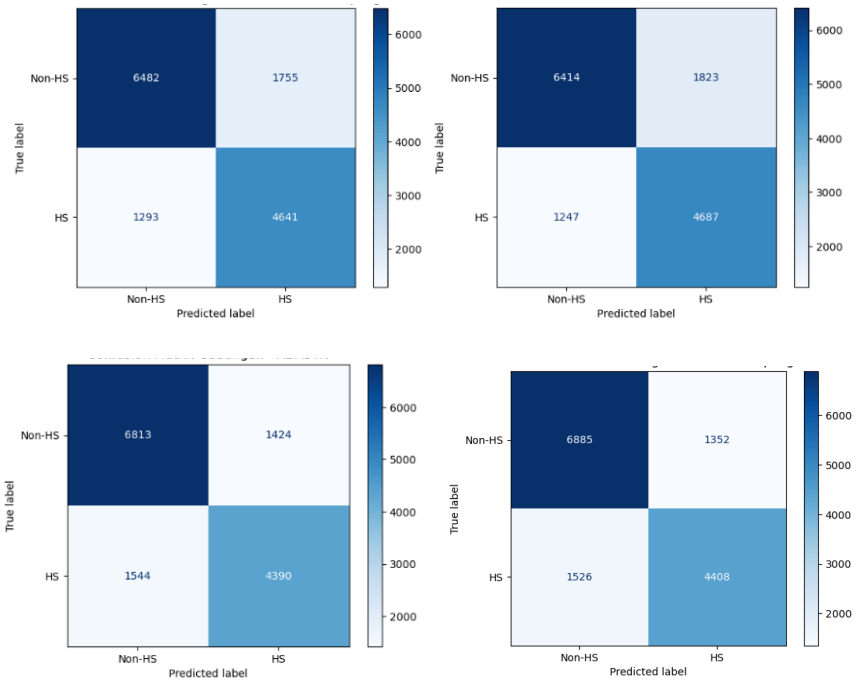

Fig. 6. Combined confusion Matrices of the Original Dataset.

In the Original Dataset scenario, the synthetic oversampling techniques demonstrated their strengths. SMOTE achieved the highest number of True Positives (4,687) and the lowest number of False Negatives (1,247), which directly accounts for its superior Recall and F1-score in this balanced context. Conversely, ADASYN produced the fewest False Positives (1,424), confirming its high precision and validating its effectiveness at preventing the misclassification of non-hate speech comments.

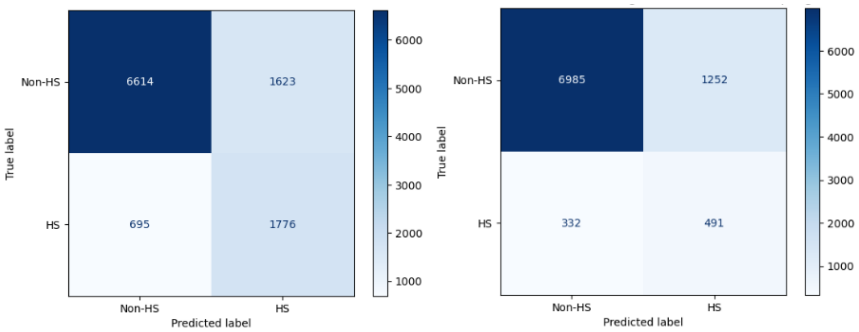

Fig. 3. Confusion Matrices for Random Oversampling (30% and 10% Scenarios).

However, as the class imbalance intensified in the 30% and 10% scenarios, the performance dynamics changed dramatically. The matrices reveal a sharp drop in the number of True Positives and a corresponding rise in False Negatives for all models, confirming that they struggled to identify the increasingly rare hate speech class. The observed trend underscores the "accuracy paradox" in the 10% scenario, where high accuracy—driven by 6,985 true negatives in Random Oversampling—conceals the model's reduced effectiveness in identifying hate speech, as reflected by the relatively low number of true positives (491).

In the 10% minority scenario, Random Oversampling proved to be the most effective, yielding the highest number of true positives (491) and the lowest number of false negatives (332), outperforming SMOTE, ADASYN, and the baseline model without oversampling.

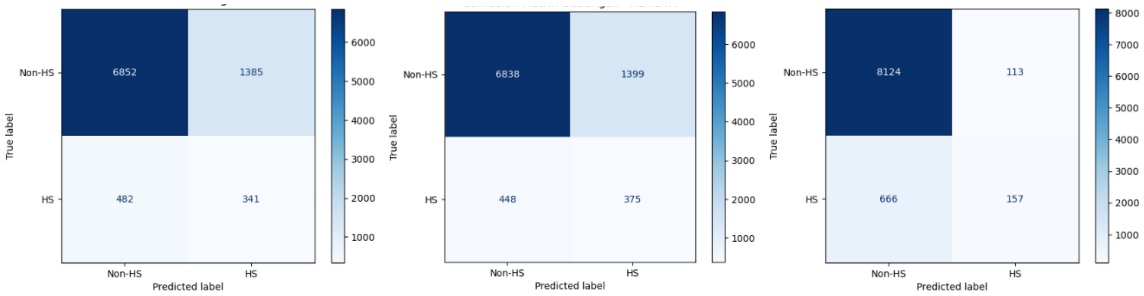

Fig. 4 Confusion Matrices for each SMOTE, ADASYN, and No Oversampling in 10% scenario.

Suggesting that when the original data from the minority class is very weak, simply duplicating proves to be more reliable strategy than creating synthetic data, which may introduce to noises in the training data. These findings are practically relevant for building models to detect other rare but critical content, such as misinformation or online radicalization, where it is essential to preserve the quality and integrity of the few positive instances. Thus, selecting an optimal oversampling technique depends greatly on the specific context and level of class imbalance, with no universally superior method.

B. Discussion.

Based on the three testing scenarios conducted in Bidirectional Long Short-Term Memory (BiLSTM) Model for sentiment analysis on dataset acquired from the Indonesian Hate Speech Superset, after undergoing preprocessing (cleaning, case folding, stopword removal, text normalization, stemming), using FastText as word-embeddings, and using Hyperparameter as such mentioned in TABLE II, reveals an interesting findings.

While generally the F1-score of the model with oversampling, but the more severe imbalanced of the dataset, the more evaluation results of the oversampling technique shines compare to the model trained without oversampling techniques. The simplicity of Random Oversampling proves unexpectedly effective in highly imbalanced datasets with achieved the highest F1-score in both the *30% Minority* (60%) and *10% Minority* (38%) datasets, potentially because Synthethic Minority Over-Sampling Technique (SMOTE) and Adaptive Synthethic Sampling (ADASYN) struggle to generate samples when the original minority class data is extremely low. In contrast, SMOTE's interpolation strategy achieved balanced results between Recall (78%) and F1-score (75.4%), when there is enough minority data for the model to learn from effectively.

This study acknowledges several potential threats to the validity of its findings. The dataset, while comprehensive and publicly available [19], may carry inherent biases from social media demographics, linguistic variations, and annotation subjectivity, potentially limiting its full representativeness of all Indonesian hate speech nuances. Consequently, the generalizability of these findings is primarily specific to the Indonesian language and this particular dataset; while methodologies are broadly applicable, optimal performance and oversampling strategies might vary across different languages, platforms, or datasets. Furthermore, oversampling techniques themselves have limitations: Random Oversampling risks overfitting through duplication, while synthetic methods like SMOTE and ADASYN might introduce noise or unrepresentative samples, especially in high-dimensional or severely imbalanced text data. This research also deliberately focuses on a BiLSTM-FastText architecture and systematic oversampling comparison, meaning it does not include direct performance comparisons against more complex, state-of-the-art Transformer-based models (e.g., IndoBERTweet)[10] or other traditional machine learning algorithms. Finally, while hyperparameters were chosen through experimentation and Early Stopping was used, a more exhaustive hyperparameter optimization might yield slightly different results, though it was not the primary focus.

While this study thoroughly investigates the impact of different oversampling strategies within a BiLSTM-FastText framework, it's important to acknowledge its scope. This research did not include a direct experimental comparison with state-of-the-art Transformer-based models, such as IndoBERTweet [10], or other traditional machine learning algorithms (e.g., SVM, Naive Bayes) or simpler deep learning architectures (e.g., vanilla LSTM). These models are known to achieve high performance in various NLP tasks, including hate speech detection. Our primary objective was to analyze the role of oversampling techniques on a robust deep learning architecture. Therefore, directly comparing our results with these other benchmarks is beyond the current scope and is considered an important direction for future work. Such comparisons would offer further insights into the trade-offs between model complexity, computational requirements, and performance gains in the challenging task of highly imbalanced hate speech detection.

V. CONCLUSION

This This study successfully developed and evaluated a Bidirectional Long Short-Term Memory (BiLSTM) model, integrated with FastText word embeddings, for the detection of Indonesian hate speech. A primary focus was the systematic comparison of three oversampling techniques—Random Oversampling, SMOTE, and ADASYN—across varying degrees of data imbalance. The results clearly demonstrate that applying oversampling techniques significantly enhances model performance, particularly in improving the Recall and F1-score for the minority (hate speech) class, which is crucial for effective detection.

A key finding of this research is that the optimal choice of oversampling technique is highly dependent on the severity of the data imbalance. Specifically, SMOTE proved most effective for the original, moderately imbalanced dataset, achieving the highest F1-score. Conversely, Random Oversampling demonstrated superior robustness and performance in severely imbalanced scenarios (30% and 10% minority classes), outperforming synthetic generation methods. These findings provide valuable insights into selecting appropriate data balancing strategies, emphasizing that a one-size-fits-all approach is not effective in handling class imbalance in hate speech detection.

For future work, it is recommended to explore direct comparisons with state-of-the-art Transformer-based models, such as IndoBERTweet, to further benchmark the performance and efficiency trade-offs. Additionally, future research could investigate the application of ensemble methods, advanced data augmentation techniques beyond oversampling, or explore multi-label classification for more nuanced hate speech categories. Expanding the dataset to include more diverse social media platforms and regional dialects could also enhance the generalizability of the detection system.

DATA AND COMPUTER PROGRAM AVAILABILITY

Data used in this paper can be accessed on the following site: huggingface.co/datasets/manueltonneau/Indonesian-hate-speech-superset, while the program can be accessed on the following site https://github.com/arcielys/BiLSTM-and-FastText-for-Indonesian-Hate-Speech-Detection.

ACKNOWLEDGMENT

All praise and gratitude are extended to Allah SWT for the grace and guidance that made the completion of this research possible. The authors would like to convey their profound appreciation to their supervisor, Mr. Yuliant Sibaroni, for his invaluable mentorship, insightful feedback, and consistent support throughout this entire research project. Sincere gratitude is also extended to the authors' beloved families for their endless encouragement, patience, and prayers, which were a constant source of strength. The authors are also thankful for their friends and colleagues who provided assistance and constructive discussions during the writing process. Finally, the authors thank all parties who have contributed, directly or indirectly, to the successful completion of this paper

REFERENCES

[1] Safitri, D., & Nofrita, M. (2024). Analisis ujaran kebencian pada kolom komentar akun Instagram Lk. *KODE: Komunikasi Digital, 13*(4).

[2] Al Faruqi, M. K., Mubarok, M. S., & Adiwijaya. (2022). Indonesian hate speech detection on Instagram comments using IndoNLU-RoBERTa. *Jurnal Ilmiah Pendidikan dan Ilmu Pengetahuan, 2*(4), 3898–3907.

[3] Putra, I. G. M., & Nurjanah, D. (2020). Hate speech detection in Indonesian language Instagram. In *Proceedings of ICACSIS* (pp. 413–420).

[4] Koto, F., & Kvartan, V. (2020). On the challenges of creating a dataset for Indonesian offensive language detection. In *Proceedings of the Workshop on Abusive Language Online* (pp. 148–154).

[5] Pratama, E. R., Alfaraby, R. A., Fadillah, R. N., & Firdaus, M. (2022). Hate speech detection on Indonesian text using machine learning approach. *TELKOMNIKA (Telecommunication Computing Electronics and Control, 20*(2), 346–354.

[6] Zulaeha, S., Mubarok, M. S., & Adiwijaya. (2021). Hate speech detection on Indonesian Twitter using Long Short-Term Memory (LSTM). In *Proceedings of ICoICT* (pp. 404–409).

[7] Kovács, A. (2022). An empirical comparison and evaluation of minority oversampling techniques on a large number of datasets. *Applied Soft Computing, 130*, 109658.

[8] Sihombing, R. H., Mubarok, M. S., & Adiwijaya. (2023). Indonesian hate speech detection using bidirectional long short-term memory. *Jurnal RESTI, 7*(1), 163–170.

[9] Ilma, R. A., Hadi, S., & Helen, A. (2021). Twitter's hate speech multi-label classification using Bidirectional Long Short-Term Memory (BiLSTM) method. In *Proceedings of ICAIBDA*.

[10] Fajri, M. R., Mubarok, M. S., & Adiwijaya. (2022). Indonesian hate speech detection using IndoBERTweet and BiLSTM. *JOIV: International Journal on Informatics Visualization, 6*(4), 856–862.

[11] Marpaung, A., Rismala, R., & Nurrahmi, H. (2021). Hate speech detection in Indonesian Twitter texts using Bidirectional Gated Recurrent Unit. In *Proceedings of KST* (pp. 186–190).

[12] Badri, N., Kboubi, F., & Chaibi, A. H. (2022). Combining FastText and Glove word embedding for offensive and hate speech text detection. *Procedia Computer Science, 207*, 769–778.

[13] Puteri, F., Sibaroni, Y., & Fitriyani, F. (2023). Hate speech detection in Indonesia Twitter comments using Convolutional Neural Network (CNN) and FastText word embedding. *Jurnal Media Informatika Budidarma, 7*, 1154–1160.

[14] Azizah, N., Guritno, W., & Ardiansyah, F. (2024). Ekspansi fitur dengan FastText untuk analisis sentimen di media sosial X menggunakan Recurrent Neural Network dan Convolutional Neural Network. *Telkom University Open Library*.

[15] Tonneau, M., Liu, D., Fraiberger, S., Schroeder, R., Hale, S., & Röttger, P. (2024). From languages to geographies: Towards evaluating cultural bias in hate speech datasets. In *Proceedings of the 8th Workshop on Online Abuse and Harms (WOAH)* (pp. 283–311).

[16] Grave, E., Bojanowski, P., Gupta, P., Joulin, A., & Mikolov, T. (2018). Learning word vectors for 157 languages. In *Proceedings of LREC* (pp. 3483–3487).

[17] Loyola-Gonzalez, O., Monroy, R., Esponda, R., & Martínez-Trinidad, J. F. (2021). A comprehensive survey on the proposal and application of ADASYN. *IEEE Access, 9*, 63515–63529.

[18] Wenando, F. A., Fadillah, R. N., Kusumawardani, A. S., & Adiwijaya. (2025). Optimizing hate speech detection in Indonesian social media: An ADASYN and LSTM-based approach. *Journal Européen des Systèmes Automatisés, 58*(1), 13–20.

[19] Cahyaningtyas, A. P., Lestari, F., Maulana, E. A., Mubarok, M. S., & Adiwijaya. (2023). IndoNLU-HSD: A benchmark dataset for Indonesian hate speech detection and classification. *Data in Brief, 48*, 109121.

[20] Joulin, A., Grave, E., Bojanowski, P., & Mikolov, T. (2017). Bag of tricks for efficient text classification. In *Proceedings of EACL* (pp. 427–431).

[21] Rifaldi, D., Fadlil, A., & Herman. (2023). Teknik preprocessing pada text mining menggunakan data tweet "mental health". *DECODE: Jurnal Pendidikan Teknologi Informasi, 3*(2), 161–171.