Journal of Soft Computing Exploration
Homepage: htttps://shmpublisher.com/index.php/joscex
p-ISSN: 2746-7686
e-ISSN: 2746-0991

.

Comparative performance of IndoBERT-based deep learning models with SMOTE for trade tariff sentiment analysis

Muhammad Rizki Roihan[1], Muhammad Itqan Mazdadi[2], Muliadi[3], Irwan Budiman[4], Fatma Indriani[5]
[1,2,3,4,5]Department of Computer Science, Universitas Lambung Mangkurat, Indonesia

Article Info

Article history:

Received May 18, 2026
Revised June 30, 2026
Accepted July 4, 2026

Keywords:

IndoBERT
Sentiment analysis
SMOTE
Deep learning
Trade tariff

ABSTRACT

The rapid growth of social media discussions on trade tariff policies has produced large volumes of Indonesian-language opinion data, making sentiment analysis an important tool for understanding public responses. However, studies in this domain remained limited, particularly those addressing class imbalance and comparing multiple hybrid architectures. This study aimed to compare four IndoBERT-based model configurations, with and without the Synthetic Minority Over-sampling Technique (SMOTE), for classifying sentiment in Indonesian trade tariff discussions under class imbalance. The dataset consisted of Indonesian tweets related to trade tariffs collected from X (Twitter) between January and July 2025. The configurations were Logistic Regression as a baseline, IndoBERT-BiLSTM, IndoBERT-CNN, and IndoBERT-BiLSTM-CNN. The tweets were passed through IndoBERT to generate contextual embeddings without fine-tuning. Performance was evaluated using accuracy, precision, recall, and macro F1-score, with macro F1-score as the primary metric. The results showed that IndoBERT-CNN with SMOTE achieved the best macro F1-score of 0.7657 and an accuracy of 0.8654. SMOTE consistently improved recall across all deep learning architectures, with IndoBERT-CNN gaining the most, from 0.7304 to 0.7875. These findings showed that, among the four compared configurations, IndoBERT-CNN with SMOTE provided the most balanced performance for imbalanced Indonesian trade tariff sentiment classification.

Corresponding Author:

Muhammad Itqan Mazdadi,
Department of Computer Science, Faculty of Mathematics and Natural Science,
Universitas Lambung Mangkurat,
Jenderal Ahmad Yani Street KM 36, Banjarbaru, South Kalimantan 70714, Indonesia
Email: mazdadi@ulm.ac.id
https://doi.org/10.52465/joscex.v7i2.122

1. INTRODUCTION

Tariff measures have become more prominent in international trade as many governments use them more actively in policy decisions. Tariffs are often associated with protecting domestic industries, but they can also be used to support a country in trade negotiations. Evidence from tariff escalation between the United States and China shows that higher tariffs are followed by a significant decline in cross-border exports [1]. Studies on production networks also show that import tariffs can move through domestic supply chains and reduce Gross Domestic Product (GDP) as well as consumer welfare, especially in sectors with important positions in the network [2]. In April 2025, the United States introduced a reciprocal tariff scheme that raised

concerns for Indonesian exports, which brought this issue into sharper focus for emerging economies [3]. Since export activity remains important for several Indonesian sectors, changes in tariff rules quickly drew public attention. Social media became one of the main spaces where people reacted to the issue. Although these posts are often short, informal, and difficult to organize manually, they can still provide useful signals about public views on tariff-related matters.

Public reaction to tariff policy needs to be considered because it can show how people perceive economic risks, government decisions, and the wider social impact of trade policy. Sentiment analysis, which is part of Natural Language Processing (NLP), provides a way to classify these reactions into positive, neutral, and negative sentiment. With this approach, social media posts that were originally unstructured can be processed into information that is more useful for policy evaluation and decision-making [4]. Reviews in the literature have also noted the value of NLP-based sentiment analysis for studying public opinion at scale, especially on social media platforms where users from different regions share their opinions continuously [5]. Text from social media is not always easy to process, especially when the posts are short, informal, and contain slang, abbreviations, or code-switching. The sentiment labels may also be imbalanced. When one class appears much more often, the model may pay more attention to that class and become less responsive to minority opinions. These issues make contextual text representation and class balancing important parts of the modeling process in this study.

Different computational methods have been used in sentiment analysis. Classical machine learning methods, such as Support Vector Machine and Naive Bayes, have been widely used for sentiment analysis. However, they usually depend on manually designed features, which can limit their ability to represent sentence meaning and context [6]. Neural network-based models were later introduced to process text with more flexible representations. Recurrent Neural Network (RNN)-based models can read text as a sequence, but they tend to struggle with long-range dependencies, which limits their ability to capture context across distant positions in a sentence. Convolutional Neural Network (CNN)-based models have also been shown to perform well on sentence-level classification tasks by applying convolutional filters over local word windows, making them effective for capturing local n-gram features in text [7]. Their limitation is that they are less suitable when important relationships between words appear across longer parts of a sentence. Transformer-based models, such as Bidirectional Encoder Representations from Transformers (BERT), address these limitations by learning context from both directions across the full sequence [8]. For Indonesian text, IndoBERT [9], which was pre-trained on a large Indonesian corpus as part of the IndoNLU benchmark, has shown strong performance compared with classical and earlier deep learning methods, especially on informal texts such as application reviews and social media posts [10], [11].

Hybrid models can capture useful information from text, but their performance can still be affected when the training data have an imbalanced class distribution. In social media sentiment datasets, some opinion categories usually appear less often than others. When this happens, the model may learn more from the dominant class and become less sensitive to the smaller classes . This problem can be seen more clearly through macro-averaged recall and macro F1-score, because both metrics give the same importance to each class regardless of its sample size. To reduce this issue from the data side, this study uses the Synthetic Minority Over-sampling Technique (SMOTE). SMOTE creates synthetic samples for the minority class by interpolating each minority sample with its nearest neighbors in the feature space [12]. Unlike simple duplication, SMOTE does not merely replicate existing minority samples, but generates new feature combinations based on the original minority data [12]. Previous text classification studies have also reported that SMOTE and its variants can improve recall and balanced accuracy, especially when combined with high-dimensional features from pretrained language models [13], [14].

To position this study within current research, it is useful to review recent IndoBERT-based hybrid models for Indonesian sentiment analysis. In hate speech detection on Indonesian Twitter, IndoBERTweet combined with Bidirectional Long Short-Term Memory (BiLSTM) achieved an F1-score of 93.3%, while IndoBERTweet combined with a CNN model reached 87.6%. This result indicates that modeling the relationships between words can be useful for certain types of Indonesian social media text [15]. Another study used IndoBERT-BiLSTM for three-class sentiment classification on TikTok reviews and reported 81% accuracy on holdout data, with an average cross-validation score of 92.03% [16]. A different result was reported in a study on COVID-19 vaccine sentiment, where IndoBERT, a domain-specific transformer, and CNN-LSTM were compared. In that study, the domain-specific model produced the best F1-score of 73%, while standard IndoBERT obtained 64% accuracy and CNN-LSTM reached an F1-score of 61% [17]. These studies show that hybrid transformer-based models can perform well for Indonesian sentiment analysis, although the best architecture may vary depending on the dataset and task. However, each of these studies evaluates only one or two architectures on a different dataset with different preprocessing procedures and text representations, which means that the reported performance differences cannot be attributed solely to the choice of architecture. A direct comparison of these three IndoBERT-based hybrid configurations against a common baseline under identical experimental conditions remains limited in studies on Indonesian sentiment analysis.

The results also show that BiLSTM is not always better than CNN, and that model performance appears to depend on the dataset characteristics and the type of text representation used. Because of this, a fair comparison under the same experimental conditions is still needed to see how each hybrid architecture performs. The effect of class-balancing methods also varies across studies. In Indonesian election sentiment analysis, one study tested several resampling methods with Logistic Regression and found that SMOTE-Tomek Links reduced performance on the test data, resulting in only 40.6% test accuracy. Tomek Links undersampling produced a much higher test accuracy of 93.8% [18]. In contrast, other studies that applied SMOTE with transformer-based models reported better results. For example, one study improved the macro F1-score from 93% to 95% by combining SMOTE with IndoBERT for Indonesian game review sentiment classification [14]. Another study found that IndoBERT with oversampling achieved 96% accuracy in electric vehicle policy sentiment analysis and produced more balanced class performance than DeBERTa, which reached 93% accuracy [19]. These mixed findings suggest that the effectiveness of SMOTE may depend on the model architecture and the representation used. However, this issue has not been examined across several IndoBERT-based hybrid configurations in a single controlled experimental setting.

Despite these recent developments, two research gaps still need to be addressed. First, previous studies have not compared IndoBERT-BiLSTM, IndoBERT-CNN, and IndoBERT-BiLSTM-CNN together with a linear baseline under the same experimental conditions. Because of this, it remains difficult to determine which configuration performs best when the models are evaluated fairly. Second, the effect of SMOTE has not been tested across these hybrid architectures at the same time. As a result, it is still unclear whether synthetic oversampling provides similar benefits for each model or whether its effect depends on the architecture being used. To address these gaps, this study compares four model configurations, consisting of Logistic Regression as the linear baseline, IndoBERT-BiLSTM, IndoBERT-CNN, and IndoBERT-BiLSTM-CNN. Each model is evaluated under two settings, namely without SMOTE and with SMOTE, using Indonesian-language trade tariff sentiment data collected from X, formerly Twitter. Model performance is evaluated using accuracy, precision, recall, and macro F1-score. Through this evaluation, this study aims to determine which model architecture and class-balancing strategy can provide the most balanced results for Indonesian trade tariff sentiment classification.

2. METHOD

This study develops Indonesian sentiment classification that combines IndoBERT-based text representation, deep learning models, and SMOTE [AN12.1]in a single process. The research steps include collecting data, preprocessing text, automatically labeling sentiment, splitting the data into training, validation, and testing sets, generating IndoBERT tokenization and embeddings, applying SMOTE to balance the training data, training the models, and evaluating their performance. Figure 1 presents the overall research framework.

The workflow is used to compare four IndoBERT-based configurations, namely a Logistic Regression baseline and three deep learning models (IndoBERT-BiLSTM, IndoBERT-CNN, and IndoBERT-BiLSTM-CNN). Each configuration is evaluated with and without SMOTE to assess the effect of class balancing on sentiment classification. The following subsections describe each stage in detail.

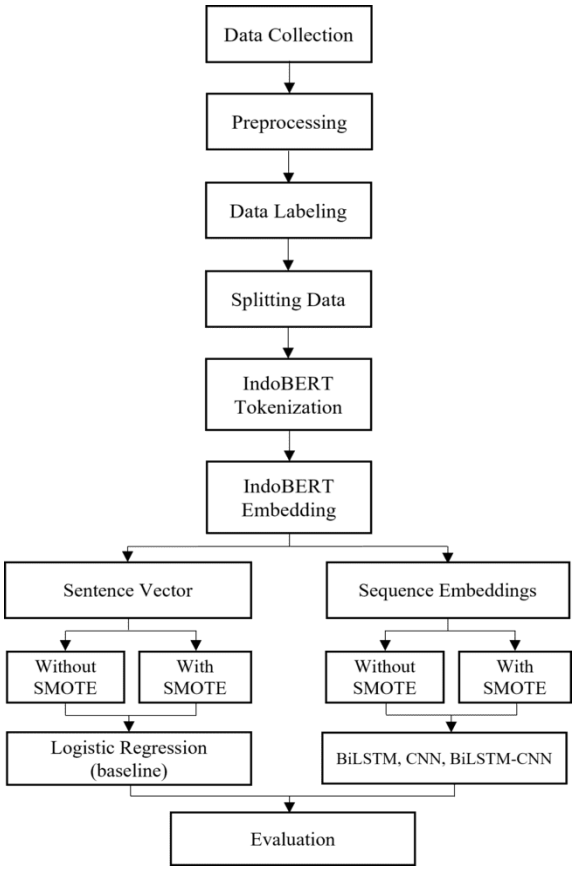

Figure 1. Proposed workflow

Data Collection

The dataset used in this study was collected from X (formerly Twitter) using a web scraping method. The data consist of Indonesian-language tweets discussing trade tariff policies, collected over the period from January 2025 to July 2025, which covers the period surrounding the United States' announcement of reciprocal import tariffs on Indonesia and the subsequent trade negotiations. The scraping process used several keywords, namely "tarif 32%", "impor tarif", "ekspor tarif", "tarif dagang", "tarif donald trump", "amerika tarif dagang", and "perang dagang". Each retrieved entry contains the tweet text along with its associated metadata, such as the posting timestamp and the search keyword. In total, 9,112 raw tweets were collected.

Preprocessing

The collected tweets were then preproce ssed to make the data cleaner, more consistent, and suitable for this study. This step is important in sentiment analysis because social media text often contains noise, informal expressions, and irregular text structures. Previous studies have shown that preprocessing can affect model performance. However, the most effective techniques may vary depending on the dataset and the model that is used [20]. For this reason, preprocessing in this study was applied selectively to reduce noise while keeping the important meaning of each tweet. The preprocessing steps are described as follows:

1. Remove URLs: Eliminating URLs starting with "http", "https", or "www".
2. Remove domain links: Removing domain patterns (".com", ".co.id", and similar formats).
3. Remove user mentions: Eliminating user mentions (e.g., @username) from the text.
4. Remove hashtags: Eliminating hashtags, including the "#" symbol and the associated words.
5. Remove non-relevant symbols: Filtering out non-alphanumeric characters.
6. Normalize spacing: Removing excessive whitespace and ensuring consistent spacing.
7. Convert text to lowercase: Transforming all text into lowercase.
8. Normalize repeated characters: Reducing excessive character repetition.
9. Normalize slang and abbreviations: Converting informal expressions into standard forms using a predefined Indonesian slang dictionary.
10. Duplicate tweet removal: After normalization, duplicate tweets were removed to prevent bias.

Data Labeling

The collected dataset did not contain sentiment labels. Therefore, automatic labeling was performed using a pre-trained transformer-based sentiment classifier. This method was used as an alternative to manual annotation, especially because the dataset contains a large number of tweets [21]. The model used in this study was the Indonesian RoBERTa Sentiment Classifier, which is based on the RoBERTa architecture and adapted for Indonesian sentiment classification [22]. Each tweet was assigned to one of three sentiment classes, namely positive, negative, or neutral. To assess the reliability of the automatic labeling, a validation procedure was conducted on a randomly selected subset of 200 tweets. Two annotators independently labeled this subset following a labeling guideline, and the inter-annotator agreement between them was measured using Cohen's Kappa [23]. Disagreements between the two annotators were resolved through discussion to finalize the agreed label for each tweet, against which the automatic labels were evaluated and reported in the Results section.

Splitting Data

The labeled dataset was divided into training, validation, and test sets using an 80:10:10 stratified split. Stratified splitting was used to keep the class distribution balanced across the three subsets [24]. To prevent data leakage, the split was performed before feature extraction, and resampling was applied only to the training set. The validation set was used for early stopping during training, while the test set was used for the final evaluation. A fixed random seed of 42 was used to ensure that the same data split could be reproduced across experimental runs.

IndoBERT Tokenization and Embedding

This study uses IndoBERT, specifically the indobenchmark/indobert-base-p1 variant, to generate text representation. IndoBERT is a pre-trained transformer-based language model for Indonesian text [9]. Previous studies have shown that BERT-based transformer models perform well in Indonesian sentiment analysis tasks [25]. Each input text is tokenized with a maximum sequence length of 128 tokens. Texts that are longer than this limit are truncated, while shorter texts are padded to the same length. An attention mask is also used to separate real tokens from padding tokens. The tokenized text is then passed through IndoBERT without gradient computation, producing a last hidden state tensor with a shape of (N, L, 768), where N represents the number of samples, L = 128 represents the sequence length, and 768 represents the hidden dimension.

Two types of representations are taken from the last hidden state. For the Logistic Regression baseline, attention-mask-weighted mean pooling is used to convert each sample into a 768-dimensional sentence vector [26]. The mean pooling process is presented in Equation (1).

$$v = \frac{\sum_{i=1}^{L} m_i \cdot h_i}{\sum_{i=1}^{L} m_i} \tag{1}$$

Where v denotes the resulting 768-dimensional sentence vector, h_i denotes the hidden state of the i-th token, $m_i \in \{0, 1\}$ denotes the attention mask value indicating whether the token is an actual token or a padding token, and L denotes the sequence length. By using the attention mask, padding tokens are excluded from the pooling process, so they do not affect the final sentence representation. This process produces a sentence-level embedding matrix $V \in R^{N \times 768}$. For the deep learning models, the full sequence embedding $S \in R^{N \times 128 \times 768}$ is kept so that sequential and positional information across tokens can still be preserved.

Class Imbalance Handling using SMOTE

Class imbalance is a common issue in text classification, including sentiment analysis. When some classes appear much less frequently than others, classifiers may have difficulty learning the patterns of the minority classes [12]. To address this issue, SMOTE is applied to the training data. SMOTE generates synthetic samples for the minority class by interpolating a minority instance with one of its k-nearest neighbors in the feature space. This method has been widely used to address imbalanced data in text classification and other machine learning tasks [12], [27]. The SMOTE formulation is presented in Equation (2).

$$x_{syn} = x_i + \lambda \cdot (x_{nn} - x_i) \tag{2}$$

Where x_{syn} denotes the generated synthetic sample, x_i denotes a randomly selected minority class sample, x_{nn} denotes one of the k-nearest neighbors of x_i, and $\lambda \in [0,1]$ denotes a random interpolation coefficient. In this study, $k = 5$ is used based on the default setting of the SMOTE algorithm. SMOTE is applied exclusively to the training set to prevent data leakage, while the validation and test sets are kept in their original imbalanced form [12], [28]. Two SMOTE variants are used according to the model architecture. For the Logistic Regression baseline, standard 2D SMOTE is applied directly to the 768-dimensional mean-pooled sentence vectors $V \in R^{N \times 768}$. The sampling strategy is set to "not majority" so that all minority classes are oversampled to match the number of samples in the majority class.

For the deep learning models, a custom 3D SMOTE variant is used to process the sequence embeddings $S \in R^{N \times 128 \times 768}$. Applying k-nearest neighbors directly to the flattened 98,304-dimensional representation (128 × 768) is avoided because distance-based methods become less reliable in very high-dimensional spaces. This problem is commonly related to the curse of dimensionality, where distance metrics lose much of their discriminative ability [29]. Instead, the k-nearest neighbors are computed in the 768-dimensional mean-pooled space V, where distance metrics remain reliable, while the interpolation in Equation (2) is applied to the corresponding full sequence embeddings S. This separation allows meaningful neighbors to be identified in a lower-dimensional space while preserving the temporal and contextual structure of the token-level representations during synthetic sample generation.

Model Development

In this study, four classification models are developed by pairing IndoBERT embeddings with classifiers of varying complexity, ranging from a linear baseline to hybrid deep learning architectures, to systematically assess the contribution of architectural design choices to sentiment classification performance.

Logistic Regression (Baseline)

In this study, Logistic Regression is used as the baseline model for multi class sentiment classification [30]. The model takes a 768-dimensional sentence vector generated from IndoBERT using attention-mask-weighted mean pooling, as explained in the IndoBERT Tokenization and Embedding subsection. The maximum number of iterations is set to 1,000 to help the model reach convergence during training. This baseline is used as a reference for comparing the deep learning architectures, since all models are built on the same IndoBERT feature space.

IndoBERT-BiLSTM

The IndoBERT-BiLSTM model uses the full sequence embedding $S \in R^{N \times 128 \times 768}$ as input to a Bidirectional Long Short-Term Memory (BiLSTM) network. BiLSTM reads the input sequence in two directions, from left to right and from right to left. This allows the model to capture information from both previous and following tokens, which is useful for understanding context in Indonesian-language text classification tasks [31]. The hidden states from the forward and backward directions are then combined at each time step, as shown in Equation (3).

$$h_t = \overrightarrow{h_t} \oplus \overleftarrow{h_t} \tag{3}$$

Where $\overrightarrow{h_t}$ and $\overleftarrow{h_t}$ denote the forward and backward hidden states at time step t, while $\oplus$ denotes the concatenation operation. In this study, the hidden dimension for each direction is set to 128, resulting in a 256-dimensional output at each time step. Mean pooling is then applied across the sequence dimension to obtain a fixed-length representation. This representation is passed through a dropout layer and a fully connected output layer to classify the input into three sentiment classes.

IndoBERT-CNN

The IndoBERT-CNN model uses the sequence embedding $S \in R^{N \times 128 \times 768}$ as input to a Convolutional Neural Network (CNN). In this model, several convolutional filters are applied in parallel with kernel sizes of 3, 4, and 5 to capture n-gram features at different lengths. This multi-filter structure has been shown to be effective in text sentiment classification tasks [32]. Each convolutional filter generates 128 feature outputs. To capture the most relevant information from each filter, global max-over-time pooling is applied across the sequence. The pooled outputs from the three kernel sizes are then combined into a single 384-dimensional representation. This representation is passed through a dropout layer and a fully connected layer to produce the final prediction for the three sentiment classes.

IndoBERT-BiLSTM-CNN

The IndoBERT-BiLSTM-CNN model is arranged as a two-stage architecture, where the BiLSTM layer is placed before the CNN layer. The sequence embedding $S \in R^{N \times 128 \times 768}$ is first sent to the BiLSTM layer, which produces a 256-dimensional contextual representation for each token position. This output is then used as the input for the CNN component. In this component, three convolutional filters with kernel sizes of 3, 4, and 5 are used to capture text patterns at different span lengths. The output from each filter is processed using global max pooling, and the pooled features are combined into one fixed-length vector. After that, the vector is passed through a dropout layer and a fully connected layer to produce the final sentiment classification result. This architecture is used because BiLSTM and CNN focus on different aspects of the text. BiLSTM helps model the relationships between tokens across the sequence, while CNN highlights important local n-gram patterns from the contextual representations generated by BiLSTM [33].

Training Configuration

For a fair comparison, the three deep learning models use the same training setup. The models are trained with the AdamW optimizer using a learning rate of 2×10^{-5} and a weight decay of 0.01 [27], [34]. During training, gradient clipping is applied with a maximum norm of 1.0 to keep the gradients stable and reduce the possibility of gradient explosion [35]. Each model also uses a dropout rate of 0.5 to reduce overfitting. The training process uses early stopping with a patience of 2 epochs, based on the validation macro F1-score, so the process stops when the validation performance does not improve further. The batch size is set to 32, and CrossEntropyLoss is used as the loss function. All training parameters are listed in Table 1.

Table 1. Training hyperparameter configuration

Parameter	Value
Base model	indobenchmark/indobert-base-p1
Maximum sequence length	128 tokens
Optimizer	AdamW
Learning rate	2×10^{-5}
Weight decay	0.01
Dropout rate	0.5
Batch size	32
Maximum epochs	10
Early stopping patience	2
Gradient clipping threshold	1.0
BiLSTM hidden dimension	128 (per direction)
CNN number of filters	128
CNN kernel sizes	3, 4, 5
Loss function	CrossEntropyLoss
LR max iterations	1,000
Random seed	42

Evaluation Metrics

All models are evaluated on the held-out test set using accuracy, precision, recall, and F1-score as evaluation metrics. All metrics are computed using macro-averaging, which calculates each metric independently for every class and then takes the unweighted mean, assigning equal weight to each class regardless of its frequency [27]. This approach is particularly suitable for imbalanced datasets, as it prevents dominant classes from masking poor performance on minority classes [12]. Among these metrics, macro F1-score is used as the primary metric for comparing model performance, because it assigns equal importance to each sentiment class regardless of its size and therefore reflects performance on the minority classes more faithfully than accuracy, which is dominated by the majority class.

Accuracy is the ratio of correctly classified samples to all predictions, as shown in Equation (4).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4}$$

Precision is the ratio of correctly predicted positive cases to all cases predicted as positive, as defined in Equation (5).

$$Precision = \frac{TP}{TP + FP} \tag{5}$$

Recall is the ratio of actual positive cases that the model correctly identifies, as shown in Equation (6).

$$Recall = \frac{TP}{TP + FN} \tag{6}$$

F1-score is the harmonic mean of precision and recall, providing a balanced measure between the two metrics, and is defined in Equation (7).

$$F1 - Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{7}$$

Where TP, TN, FP, and FN denote true positives, true negatives, false positives, and false negatives, respectively.

3. RESULTS AND DISCUSSIONS

This section reports the experimental results of sentiment classification on Indonesian-language tweets pertaining to trade tariff discussions sourced from the X platform. Four model architectures are examined, comprising Logistic Regression as a linear baseline alongside three hybrid deep learning architectures, namely IndoBERT-BiLSTM, IndoBERT-CNN, and IndoBERT-BiLSTM-CNN. Each architecture is evaluated under two training scenarios, with and without SMOTE, resulting in eight experimental configurations in total. Following the order of the methodological pipeline, the results are reported sequentially, beginning with the dataset preparation and label validation, followed by the training-set distribution before and after SMOTE, and concluding with the comparative model performance, the per-class performance breakdown, and the confusion matrix analysis.

Experimental Setup

All experiments ran on a local machine equipped with an NVIDIA GeForce RTX 4050 Laptop GPU, 16 GB of RAM, Python 3.12.10, PyTorch 2.6.0, and Hugging Face Transformers 5.1.0. The IndoBERT weights were loaded from the indobenchmark/indobert-base-p1 checkpoint on the Hugging Face Hub. To keep comparisons fair, all eight configurations, covering each of the four architectures under both SMOTE and no-SMOTE conditions, were trained using an identical hyperparameter setup.

Dataset Preparation

The data collection process described in the Method section yielded 9,112 raw tweets. These tweets were then passed through the preprocessing stage, which produced 8,690 clean tweets after 422 entries were removed during normalization and duplicate filtering. Table 2 presents representative examples of the text transformation produced by these preprocessing steps.

Table 2. Examples of text transformation before and after preprocessing

No	Original Text	Preprocessed Text
1	Cuma bisa nyinyir tanpa fakta! Tarif 32% jadi 19% itu kemenangan negosiasi Prabowo, bikin produk kita jago di AS malahan bro	cuma bisa nyinyir tanpa fakta tarif jadi itu kemenangan negosiasi prabowo bikin produk kita jago di as malahan bro
2	Padahal penurunan tarif jadi 19% itu bukan beban, tapi justru penyelamat ekspor kita	padahal penurunan tarif jadi itu bukan beban tapi justru penyelamat ekspor kita
3	salut bgt, pemerintah jaga data pribadi kita! Gk nyangka bisa tegas dlm negosiasi tarif impor sama AS	salut banget pemerintah jaga data pribadi kita enggak menyangka bisa tegas dalam negosiasi tarif impor sama as
4	Utk apa AS minta data pribadi rakyat RI mereka kelola klo gak ada nilai ekonomis, politis, dan keamanan. Pejabat kita aja yg bodoh mau aja menyetujui pengalihan data pribadi utk penurunan tarif ekspor yg gak ada hubungannya padahal	untuk apa as meminta data pribadi rakyat ri mereka kelola kalo enggak ada nilai ekonomis politis dan keamanan pejabat kita saja yang bodoh mau saja menyetujui pengalihan data pribadi untuk penurunan tarif ekspor yang enggak ada hubungannya padahal
5	@grok benarkah trump tetap menerapkan tarif 32% untuk indonesia? apakah berita ini valid?	benarkah trump tetap menerapkan tarif untuk indonesia apakah berita ini valid

Sentiment Labeling and Validation

The clean tweets were labeled using the Indonesian RoBERTa sentiment classifier [22]. The resulting label distribution is shown in Figure 2, where the neutral class dominates with 6,413 tweets, followed by 1,520 negative tweets and 757 positive tweets. This distribution reflects a pronounced class imbalance, with the positive class forming the smallest portion of the dataset. The dominance of the neutral class motivates the use

of macro-averaged metrics, which weight each class equally, and the application of class balancing on the training data in the subsequent stages.

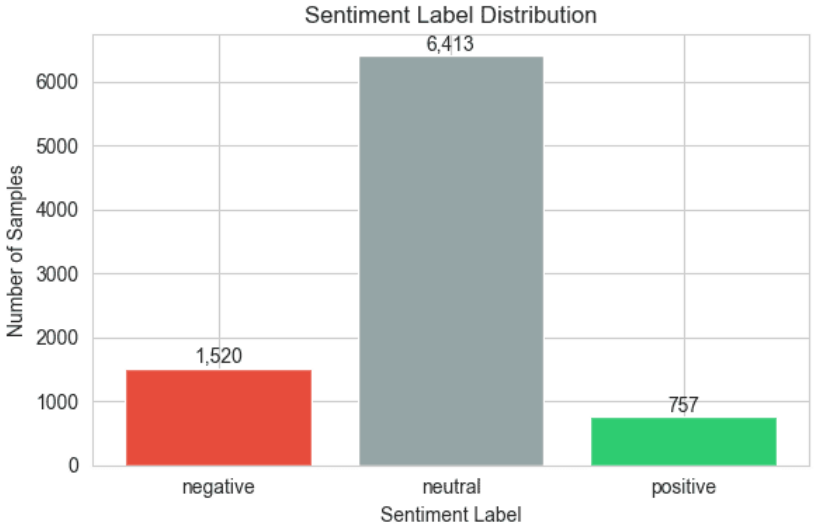

Figure 2. Sentiment label distribution of the labeled dataset

Because these labels were generated automatically, their reliability was assessed before they were used for model training. A randomly selected subset of 200 tweets was independently annotated by two annotators, as described in the Method section. The two annotators reached a raw agreement of 73%, agreeing on 146 of the 200 tweets, with a Cohen's Kappa of 0.4357, indicating moderate agreement. Almost all disagreements were between adjacent classes, particularly at the boundary between the neutral and negative classes, while opposing disagreements between the positive and negative classes were negligible. This pattern is consistent with the inherent subjectivity of sentiment interpretation in short and informal social media text, where the distinction between a neutral statement and a mildly negative one is often unclear.

Using the agreed annotations as the reference, the automatic labels achieved an accuracy of 0.85 and a macro F1-score of 0.78, with per-class F1-scores of 0.81 for the negative class, 0.89 for the neutral class, and 0.65 for the positive class. The lower score on the positive class reflects its small size and the difficulty of distinguishing genuinely positive opinions from neutral statements in this domain, with a notable portion of positive tweets being labeled as neutral by the automatic classifier. These results indicate that, although the automatic labels are not entirely free of noise, particularly for the minority positive class, their overall quality is sufficient to support the comparative evaluation conducted in this study.

Data Splitting and Class Balancing

The labeled dataset was divided into training, validation, and test sets using the stratified 80:10:10 split described in the Method section, resulting in 6,952 training samples, 869 validation samples, and 869 test samples. The stratified procedure preserved the original class proportions across the three subsets, so the test set retained the imbalance present in the full dataset, with 642 neutral, 152 negative, and 75 positive tweets.

SMOTE was then applied to the training set only. Before resampling, the training set contained 5,130 neutral, 1,216 negative, and 606 positive samples. After applying SMOTE with the not-majority strategy, the two minority classes were oversampled to match the neutral class, producing 5,130 samples per class and a balanced training set of 15,390 samples in total. Figure 3 shows the training-set class distribution before and after SMOTE. The validation and test sets were kept in their original imbalanced form, so that the evaluation reflected the realistic class proportions encountered at inference time.

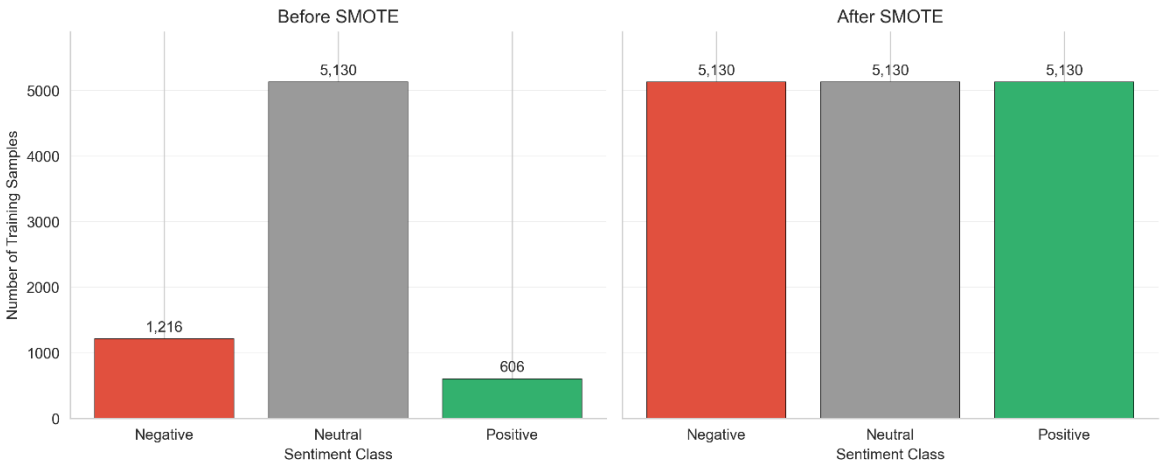

Figure 3. Training-set class distribution before and after SMOTE

Overall Model Performance

Using the balanced training set described in the previous subsection, where each minority class was oversampled to 5,130 samples, all eight experimental configurations were trained and then evaluated on the original imbalanced test set. The classification results are presented in Table 3, which reports accuracy, precision, recall, and macro F1-score for each model under the two training scenarios, without SMOTE and with SMOTE.

Table 3. Classification performance of all experimental configurations

Model	Smote	Accuracy	Precision	Recall	F1-Score
Logistic Regression (baseline)	No	0.8700	0.7671	0.7538	0.7598
IndoBERT-BiLSTM	No	0.8665	0.7598	0.7046	0.7216
IndoBERT-CNN	No	0.8654	0.7708	0.7304	0.7426
IndoBERT-BiLSTM-CNN	No	0.8654	0.7802	0.7130	0.7360
Logistic Regression (baseline)	Yes	0.8527	0.7252	0.7645	0.7426
IndoBERT-BiLSTM	Yes	0.8516	0.7298	0.7829	0.7518
IndoBERT-CNN	Yes	0.8654	0.7482	0.7875	0.7657
IndoBERT-BiLSTM-CNN	Yes	0.8619	0.7386	0.7798	0.7573

As shown in Table 3, IndoBERT-CNN with SMOTE achieves the highest macro F1-score among all configurations, reaching 0.7657 with an accuracy of 0.8654. This indicates that the combination of IndoBERT-CNN and SMOTE provides the most balanced classification performance in this study, since macro F1-score is used as the primary evaluation metric for imbalanced data. Among the configurations without SMOTE, the Logistic Regression baseline records the highest macro F1-score at 0.7598, remaining competitive with the deep learning models in this scenario, while IndoBERT-BiLSTM without SMOTE records the lowest macro F1-score overall, at 0.7216. The fact that IndoBERT-CNN outperforms IndoBERT-BiLSTM-CNN can be explained by how IndoBERT is used in this study. Since IndoBERT is used as a feature extractor without fine-tuning, the embeddings it produces already carry contextual information. CNN can directly pick up the most useful patterns from these embeddings through its convolutional filters. Adding a BiLSTM layer on top of that, as in IndoBERT-BiLSTM-CNN, does not necessarily help and may even make the optimization harder, which could explain the slightly lower performance.

A comparison between the without-SMOTE and with-SMOTE scenarios shows that SMOTE has the clearest impact on recall. All three deep learning architectures experience recall improvements after SMOTE is applied. IndoBERT-CNN shows the largest increase, from 0.7304 to 0.7875, followed by IndoBERT-BiLSTM, which improves from 0.7046 to 0.7829. IndoBERT-BiLSTM-CNN also shows a recall increase, from 0.7130 to 0.7798. These results show that SMOTE helped the models recognize more minority class samples. This is important because the sentiment classes in the dataset are not evenly distributed.

However, the improvement was not seen in all evaluation metrics. After SMOTE was applied, recall generally increased, but accuracy and precision tended to decrease. This shows that the models became more sensitive to minority classes, though they also produced more false positive predictions. This trade-off is most clearly seen in the Logistic Regression baseline, where applying SMOTE raised recall from 0.7538 to 0.7645 but lowered precision from 0.7671 to 0.7252, so that its macro F1-score dropped from 0.7598 to 0.7426. In contrast, for the three deep learning architectures, the gain in recall outweighed the loss in precision, and their macro F1-scores increased after SMOTE. These results indicate that SMOTE did not consistently improve performance across all models, and that its effect depends on the model architecture.

Among the tested models, SMOTE yielded the greatest improvement when combined with the IndoBERT-CNN architecture. The increase in recall and macro F1-score suggests that this model was better able to use the more balanced training data produced by SMOTE. Even so, the results also show that the effect of SMOTE depends on the model architecture. Therefore, its impact should not be judged solely by accuracy, but also by class-sensitive metrics such as recall and macro F1-score.

Per-Class Performance Breakdown

While Table 3 reports the aggregate performance of each configuration, it does not reveal how each model behaves on the individual sentiment classes, which is essential given that the main concern of this study is the minority classes. Table 4 therefore presents the precision, recall, and F1-score for the negative, neutral, and positive classes across all eight configurations.

Table 4. Per-class performance breakdown of all configurations

Model	SMOTE	Neg P	Neg R	Neg F1	Neu P	Neu R	Neu F1	Pos P	Pos R	Pos F1
Logistic Regression (baseline)	No	0.73	0.76	0.74	0.93	0.93	0.93	0.64	0.57	0.61
IndoBERT-BiLSTM	No	0.70	0.77	0.74	0.92	0.94	0.93	0.65	0.40	0.50
IndoBERT-CNN	No	0.70	0.80	0.75	0.92	0.93	0.93	0.69	0.47	0.56
IndoBERT-BiLSTM-CNN	No	0.70	0.74	0.72	0.92	0.94	0.93	0.72	0.45	0.56
Logistic Regression (baseline)	Yes	0.70	0.74	0.72	0.94	0.90	0.92	0.53	0.65	0.59
IndoBERT-BiLSTM	Yes	0.66	0.84	0.74	0.95	0.88	0.91	0.58	0.63	0.60
IndoBERT-CNN	Yes	0.71	0.84	0.77	0.95	0.90	0.92	0.59	0.63	0.61
IndoBERT-BiLSTM-CNN	Yes	0.70	0.80	0.75	0.95	0.90	0.93	0.56	0.64	0.60

The per-class results show that the effect of SMOTE is concentrated almost entirely on the two minority classes, while the majority neutral class remains stable. For the neutral class, the F1-score stays within a narrow band of 0.91 to 0.93 across all configurations, indicating that oversampling the minority classes does not degrade performance on the dominant class.

The clearest effect of SMOTE is the increase in recall for the minority classes. For the positive class, recall rises after SMOTE in every architecture, from 0.40 to 0.63 for IndoBERT-BiLSTM, from 0.47 to 0.63 for IndoBERT-CNN, and from 0.45 to 0.64 for IndoBERT-BiLSTM-CNN. The negative class shows the same direction, with recall increasing for all three deep architectures, for example from 0.80 to 0.84 for IndoBERT-CNN. This confirms that SMOTE makes the models more capable of recognizing minority-class samples that would otherwise be overlooked in favor of the dominant neutral class.

This gain in recall comes at the cost of precision on the positive class, which decreases after SMOTE in all configurations, for instance from 0.69 to 0.59 for IndoBERT-CNN. The models become more willing to predict the minority classes, capturing more true minority instances but also producing more false positives. The net effect on the positive-class F1-score is favorable for the three deep architectures, which improve from 0.50 to 0.60 for IndoBERT-BiLSTM, from 0.56 to 0.61 for IndoBERT-CNN, and from 0.56 to 0.60 for IndoBERT-BiLSTM-CNN. For the Logistic Regression baseline, by contrast, the positive-class F1-score declines slightly from 0.61 to 0.59, so that its macro F1-score also decreases. This difference indicates that the benefit of SMOTE on the minority classes is realized by the deep architectures but not by the linear baseline, which is consistent with prior findings that the effectiveness of oversampling depends on the classifier [12].

These per-class results also explain why the overall accuracy does not increase after SMOTE. Because accuracy is dominated by the neutral class, which accounts for the largest share of the test set, the shift in model capacity toward the minority classes produces only a small change in accuracy while improving the balance across classes. This is most evident in IndoBERT-CNN. As shown in Table 3, its accuracy remains unchanged at 0.8654 while its macro F1-score rises from 0.7426 to 0.7657, and the per-class results in Table 4 explain this improvement, since its positive-class recall increases from 0.47 to 0.63 after SMOTE. This demonstrates that the value of SMOTE in this study lies in improving minority-class sensitivity rather than overall accuracy, and that macro F1-score, which weights each class equally, is a more appropriate measure of effectiveness under class imbalance than accuracy alone.

Despite these improvements, the positive class remains the most difficult to classify, with its F1-score not exceeding 0.61 in any configuration. This reflects both the small size of the positive class and the noise observed during label validation, where a notable portion of positive tweets were assigned to the neutral class. SMOTE mitigates but does not fully overcome these limitations, indicating that further gains would likely require additional positive-class data or stronger representation learning rather than oversampling alone.

Confusion Matrix Analysis

Although aggregate metrics are useful for summarizing overall performance, they do not show where the classification errors occur. For this reason, the confusion matrices in Figure 4 are used to examine the predictions of the eight model configurations in more detail. All matrices are computed on the same test set, which consists of 152 negative, 642 neutral, and 75 positive tweets. The upper row presents the models trained without SMOTE, while the lower row presents the models trained with SMOTE. This comparison makes it easier to observe how class imbalance affects each model and how the use of SMOTE changes the prediction patterns across sentiment classes.

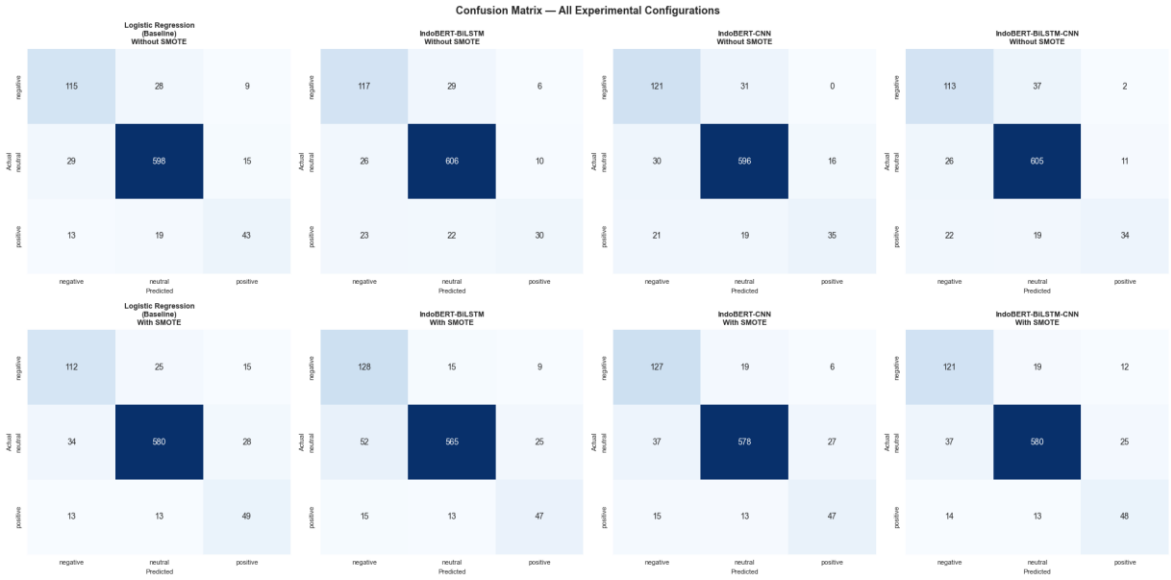

Figure 4. Confusion matrix result for all models

The neutral class is identified well by all models, which is expected because it has the highest number of samples in the test set. From 642 neutral samples, each model correctly classifies between 565 and 606 samples. This result shows that the models are generally reliable in recognizing the majority class. The performance is less stable for the smaller classes, especially the positive class, which consists of only 75 test samples. In the without-SMOTE setting, many positive samples are predicted as neutral or negative, showing that the models are still affected by the dominant class in the training data.

Among the deep learning models trained without SMOTE, the positive class remains the most difficult to identify. IndoBERT-BiLSTM correctly classifies 30 of 75 positive samples, while IndoBERT-CNN and IndoBERT-BiLSTM-CNN classify 35 and 34 samples correctly. These numbers show that the models still struggle to learn positive-class patterns when class balancing is not applied. The Logistic Regression baseline gives competitive results, correctly classifying 43 positive samples, which indicates that IndoBERT embeddings already provide useful information for separating sentiment classes. However, the uneven class distribution in the training data still limits how well the deep learning models recognize minority-class patterns.

After SMOTE is applied, the prediction pattern changes most clearly in the positive class. The deep learning models identify more positive samples than in the without-SMOTE setting. IndoBERT-CNN with SMOTE shows a clear improvement by correctly classifying 47 of 75 positive samples, compared with 35 without SMOTE, and reducing positive-to-neutral errors from 19 to 13 samples. This model also gives the best result for the negative class, with 127 of 152 samples classified correctly. Even so, the improvement in the positive class helps IndoBERT-CNN with SMOTE produce a more balanced result, as reflected in its macro F1-score of 0.7657. The Logistic Regression baseline does not gain the same benefit from SMOTE. Its correct positive predictions increase from 43 to 49, but its correct neutral predictions decrease from 598 to 580, which leads to a lower macro F1-score. This result indicates that SMOTE is more useful for the non-linear deep learning models than for the linear baseline in this experiment.

Overall, the confusion matrix results show that class imbalance affects how the models make predictions, especially for the positive class. SMOTE helps reduce this problem by improving positive-class recognition, particularly when combined with IndoBERT-CNN. These findings also show that the best result is not always produced by the most complex architecture. In this study, better performance comes from combining a suitable model architecture with an effective class balancing strategy.

Learning Dynamics and Loss Curve Analysis

The training and validation loss curves were used to observe how each model learned during training and how well it maintained performance on validation data. As shown in Figure 5, the loss values are presented for IndoBERT-BiLSTM, IndoBERT-CNN, and IndoBERT-BiLSTM-CNN under both without SMOTE and with SMOTE settings. Early stopping was also used to reduce the risk of overfitting by keeping the model weights from the epoch with the best validation performance.

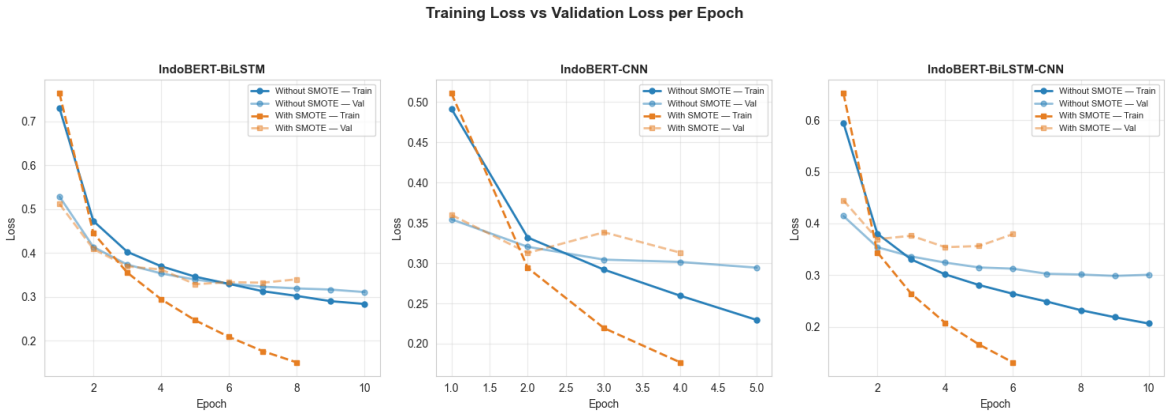

Figure 5. Loss trends during model training and validation

Without SMOTE, the three models showed a steady learning pattern during training. The training loss and validation loss both decreased gradually across epochs, which indicates that the models could learn from the original imbalanced data without showing clear overfitting. Among the three models, IndoBERT-BiLSTM-CNN without SMOTE showed the most consistent progress. Its validation loss decreased from 0.4144 in the first epoch to around 0.3007 by the final epoch, suggesting that this model maintained good validation performance when trained on the original data distribution. This suggests that the hybrid BiLSTM-CNN architecture maintains stable generalization throughout the full training process when trained on the original data distribution.

In the with-SMOTE scenario, the training loss decreases more sharply than the validation loss across all architectures. This pattern is especially visible in IndoBERT-BiLSTM and IndoBERT-BiLSTM-CNN, where the gap between training and validation loss becomes wider as training progresses. For IndoBERT-CNN with SMOTE, the training loss continues to decline rapidly in the early epochs and training stops at epoch 4 through early stopping, while the validation loss decreases more moderately, indicating faster adaptation to the augmented training data. Meanwhile, IndoBERT-BiLSTM-CNN with SMOTE shows a clearer divergence pattern, where the training loss continues to decrease to 0.1311 at epoch 6, while the validation loss begins to increase after reaching its lowest point at epoch 4. This suggests that SMOTE-augmented training may improve learning on minority-class patterns, but it can also increase the risk of overfitting when the model continues to adapt too strongly to the synthetic training samples.

4. CONCLUSION

This study evaluated four IndoBERT-based model configurations for classifying sentiment in Indonesian-language trade tariff discussions from X (Twitter), both with and without SMOTE. The configurations included Logistic Regression as a baseline, IndoBERT-BiLSTM, IndoBERT-CNN, and IndoBERT-BiLSTM-CNN, making eight experiments in total. Among all configurations, IndoBERT-CNN with SMOTE achieved the highest macro F1-score at 0.7657 and an accuracy of 0.8654, providing the best overall balance across the three sentiment classes. The application of SMOTE increased recall in all three deep

learning models, with the most pronounced change shown by IndoBERT-CNN, where recall increased from 0.7304 to 0.7875. This improvement in minority-class recall is the main objective of applying SMOTE under class imbalance, and it is more meaningful than overall accuracy, which is dominated by the majority neutral class. For IndoBERT-CNN, the accuracy remained unchanged at 0.8654 while the macro F1-score improved, showing that better handling of the minority classes was achieved without reducing overall performance. For the other models, the small decrease in accuracy after SMOTE reflects the shift of model capacity toward the minority classes rather than a degradation in learning, which is why macro F1-score is used as the primary evaluation metric in this study. The Logistic Regression baseline produced a lower macro F1-score when SMOTE was applied, indicating that synthetic oversampling was more beneficial for the IndoBERT-based deep learning models than for the linear baseline. The loss curves also indicate that SMOTE may increase the risk of overfitting, for which early stopping was applied to maintain stable validation performance during training.

The main contribution of this study is a systematic comparison of IndoBERT-based hybrid architectures with and without SMOTE for Indonesian trade tariff sentiment analysis, a domain that has received limited attention, together with a demonstration that the benefit of class balancing depends on the model architecture rather than applying uniformly across classifiers. In practical terms, the findings indicate that combining IndoBERT embeddings with a convolutional architecture and SMOTE offers an effective approach for analyzing public sentiment toward trade tariff policies under class imbalance, which can support the monitoring of public opinion on economic policy from social media data.

This study still has several limitations. The dataset is limited to Indonesian-language social media posts about trade tariffs within a specific period. The sentiment labels were generated automatically using a pre-trained classifier and validated on a sample through inter-annotator agreement, rather than being fully manually annotated. IndoBERT was also used only as a feature extractor, without fine-tuning. Future research can explore other class imbalance strategies, such as focal loss or class-weighted learning. It can also compare more Indonesian transformer-based models and apply end-to-end fine-tuning to improve the detection of minority classes in Indonesian sentiment analysis.

REFERENCES

[1] T. Yang, W.-Y. Lau, and E. N. A. Bahri, "The Impact of US-China Trade War on China's Exports: Evidence From Difference-in-Differences Model," *Sage Open*, vol. 15, no. 2, p. 21582440251328480, 2025, doi: 10.1177/21582440251328482.

[2] H. Kreuter and M. Riccaboni, "The impact of import tariffs on GDP and consumer welfare: A production network approach," *Econ. Model.*, vol. 126, p. 106374, 2023, doi: 10.1016/j.econmod.2023.106374.

[3] W. J. McKibbin, M. Noland, and G. Shuetrim, "The Global Economic Effects of Trump's 2025 Tariffs," WP25-13, Jun. 2025. doi: 10.2139/ssrn.5338095.

[4] F. Aftab *et al.*, "A Comprehensive Survey on Sentiment Analysis Techniques," *International Journal of Technology*, vol. 14, no. 6, pp. 1288–1298, 2023, doi: 10.14716/ijtech.v14i6.6632.

[5] J. R. Jim, M. A. R. Talukder, P. Malakar, M. M. Kabir, K. Nur, and M. F. Mridha, "Recent advancements and challenges of NLP-based sentiment analysis: A state-of-the-art review," *Natural Language Processing Journal*, vol. 6, p. 100059, 2024, doi: 10.1016/j.nlp.2024.100059.

[6] Z. Shi and R. Agrawal, "A Comprehensive Survey of Contemporary Arabic Sentiment Analysis: Methods, Challenges, and Future Directions," in *Findings of the Association for Computational Linguistics: NAACL 2025*, L. Chiruzzo, A. Ritter, and L. Wang, Eds., Albuquerque, New Mexico: Association for Computational Linguistics, Apr. 2025, pp. 3760–3772. doi: 10.18653/v1/2025.findings-naacl.208.

[7] Y. Kim, "Convolutional Neural Networks for Sentence Classification," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, A. Moschitti, B. Pang, and W. Daelemans, Eds., Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 1746–1751. doi: 10.3115/v1/D14-1181.

[8] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, J. Burstein, C. Doran, and T. Solorio, Eds., Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 4171–4186. doi: 10.18653/v1/N19-1423.

[9] B. Wilie *et al.*, "IndoNLU: Benchmark and Resources for Evaluating Indonesian Natural Language Understanding," in *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, K.-F. Wong, K. Knight, and H. Wu, Eds., Suzhou, China: Association for Computational Linguistics, Dec. 2020, pp. 843–857. doi: 10.18653/v1/2020.aacl-main.85.

[10] H. Imaduddin, F. Y. A'la, and Y. S. Nugroho, "Sentiment Analysis in Indonesian Healthcare Applications using IndoBERT Approach," *International Journal of Advanced Computer Science and Applications*, vol. 14, no. 8, p. 2023, 2023, doi: 10.14569/IJACSA.2023.0140813.

[11] M. I. K. Sinapoy, Y. Sibaroni, and S. S. Prasetyowati, "Comparison of LSTM and IndoBERT Method in Identifying Hoax on Twitter," *Jurnal RESTI*, vol. 7, no. 3, pp. 657–662, Jun. 2023, doi: 10.29207/resti.v7i3.4830.

[12] 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, p. 21631, 2025, doi: 10.1038/s41598-025-05791-7.

[13] N. A. Semary, W. Ahmed, K. Amin, P. Pławiak, and M. Hammad, "Improving sentiment classification using a RoBERTa-based hybrid model," *Front. Hum. Neurosci.*, vol. Volume 17-2023, 2023, doi: 10.3389/fnhum.2023.1292010.

[14] F. Ansyah and R. R. Suryono, "Sentiment Classification of Indonesian-Language Roblox Reviews Using IndoBERT with SMOTE Optimization," *Journal of Applied Informatics and Computing (JAIC)*, vol. 9, no. 4, pp. 1868–1877, Aug. 2025, doi: 10.30871/jaic.v9i4.10155.

[15] J. Forry Kusuma and A. Chowanda, "Indonesian Hate Speech Detection Using IndoBERTweet and BiLSTM on Twitter," *JOIV — International Journal on Informatics Visualization.*, vol. 7, no. 3, pp. 773–780, Sep. 2023, doi: 10.30630/joiv.7.3.1035.

[16] A. F. Al Farizi and Y. Sibaroni, "Implementation of BiLSTM and IndoBERT for Sentiment Analysis of TikTok Reviews," *JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)*, vol. 10, no. 1, pp. 96–106, Jan. 2025, doi: 10.29100/jipi.v10i1.5815.

[17] S. Saadah, K. M. Auditama, A. A. Fattahila, F. I. Amorokhman, A. Aditsania, and A. A. Rohmawati, "Implementation of BERT, IndoBERT, and CNN-LSTM in Classifying Public Opinion about COVID-19 Vaccine in Indonesia," *Jurnal RESTI*, vol. 6, no. 4, pp. 648–655, Aug. 2022, doi: 10.29207/resti.v6i4.4215.

[18] N. Sulistianingsih and I. N. Switrayana, "Enhancing Sentiment Analysis for the 2024 Indonesia Election Using SMOTE-Tomek Links and Binary Logistic Regression," *International Journal of Education and Management Engineering*, vol. 14, no. 3, pp. 22–32, Jun. 2024, doi: 10.5815/ijeme.2024.03.03.

[19] M. B. Nugroho, A. Khanif Zyen, and A. Widiastuti, "Multiclass Sentiment Analysis of Electric Vehicle Incentive Policies Using IndoBERT and DeBERTa Algorithms," *Journal of Applied Informatics and Computing (JAIC)*, vol. 9, no. 3, pp. 910–919, Jun. 2025, doi: 10.30871/jaic.v9i3.9511.

[20] M. A. Palomino and F. Aider, "Evaluating the Effectiveness of Text Pre-Processing in Sentiment Analysis," *Applied Sciences (Switzerland)*, vol. 12, no. 17, Sep. 2022, doi: 10.3390/app12178765.

[21] D. Rozado, R. Hughes, and J. Halberstadt, "Longitudinal analysis of sentiment and emotion in news media headlines using automated labelling with Transformer language models," *PLoS One*, vol. 17, no. 10, pp. 1–14, Oct. 2022, doi: 10.1371/journal.pone.0276367.

[22] Y. Liu *et al.*, "RoBERTa: A Robustly Optimized BERT Pretraining Approach," Jul. 2019, [Online]. Available: http://arxiv.org/abs/1907.11692

[23] M. T. Uliniansyah, A. Jarin, A. Santosa, and G. Gunarso, "Modeling sentiment analysis of Indonesian biodiversity policy Tweets using IndoBERTweet," *IAES International Journal of Artificial Intelligence (IJ-AI)*, vol. 14, no. 3, p. 2389, Jun. 2025, doi: 10.11591/ijai.v14.i3.pp2389-2401.

[24] J. Sadaiyandi, P. Arumugam, A. K. Sangaiah, and C. Zhang, "Stratified Sampling-Based Deep Learning Approach to Increase Prediction Accuracy of Unbalanced Dataset," *Electronics (Basel).*, vol. 12, no. 21, 2023, doi: 10.3390/electronics12214423.

[25] C.-H. Lin and U. Nuha, "Sentiment analysis of Indonesian datasets based on a hybrid deep-learning strategy," *J. Big Data*, vol. 10, no. 1, p. 88, 2023, doi: 10.1186/s40537-023-00782-9.

[26] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, K. Inui, J. Jiang, V. Ng, and X. Wan, Eds., Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 3982–3992. doi: 10.18653/v1/D19-1410.

[27] M. Mujahid *et al.*, "Data oversampling and imbalanced datasets: an investigation of performance for machine learning and feature engineering," *J. Big Data*, vol. 11, no. 1, Dec. 2024, doi: 10.1186/s40537-024-00943-4.

[28] Y. Jang, "Feature-based ensemble modeling for addressing diabetes data imbalance using the SMOTE, RUS, and random forest methods: a prediction study," *Ewha Med J*, vol. 48, no. 2, pp. e32-, 2025, doi: 10.12771/emj.2025.00353.

[29] N. Ukey, Z. Yang, B. Li, G. Zhang, Y. Hu, and W. Zhang, "Survey on Exact kNN Queries over High-Dimensional Data Space," *Sensors*, vol. 23, no. 2, 2023, doi: 10.3390/s23020629.

[30] T. Hariguna and A. Ruangkanjanases, "Adaptive sentiment analysis using multioutput classification: a performance comparison," *PeerJ Comput. Sci.*, vol. 9, p. e1378, May 2023, doi: 10.7717/peerj-cs.1378.

[31] R. Pramana, M. Jonathan, H. S. Yani, and R. Sutoyo, "A Comparison of BiLSTM, BERT, and Ensemble Method for Emotion Recognition on Indonesian Product Reviews," *Procedia Comput. Sci.*, vol. 245, pp. 399–408, 2024, doi: 10.1016/j.procs.2024.10.266.

[32] X. Kong and K. Zhang, "A novel text sentiment analysis system using improved depthwise separable convolution neural networks," *PeerJ Comput. Sci.*, vol. 9, p. e1236, Feb. 2023, doi: 10.7717/peerj-cs.1236.

[33] S. Susandri, S. Defit, and M. Tajuddin, "Enhancing Text Sentiment Classification with Hybrid CNN-BiLSTM Model on WhatsApp Group," *Journal of Advances in Information Technology*, vol. 15, no. 3, pp. 355–363, 2024, doi: 10.12720/jait.15.3.355-363.

[34] S. Ouamour, W. Benaouda, and H. Sayoud, "Optimization Evaluation for Enhancing Deep Learning Performance in Arabic Text Classification," in *Proceedings of the 2024 7th International Conference on Information Science and Systems*, in ICISS '24. New York, NY, USA: Association for Computing Machinery, 2025, pp. 134–139. doi: 10.1145/3700706.3700729.

[35] A. Ramaswamy, "Gradient Clipping in Deep Learning: A Dynamical Systems Perspective," in *Proceedings of the 12th International Conference on Pattern Recognition Applications and Methods*, SCITEPRESS - Science and Technology Publications, 2023, pp. 107–114. doi: 10.5220/0011678000003411.

.