Indonesian Journal of Electronics, Electromedical Engineering, and Medical Informatics e-ISSN: 2656-8624
Homepage: https://ijeeemi.org/; Vol. 7, No. 2, pp. 370-386, May 2025

RESEARCH PAPER

OPEN ACCES

Machine Learning Implementation for Sentiment Analysis on X/Twitter: Case Study of Class Of Champions Event in Indonesia

Rini Hafizah, Triando Hamonangan Saragih, Muliadi, Fatma Indriani, and Muhammad Itqan Mazdadi

Department of Computer Science, Faculty of Mathematics and Natural Science, Lambung Mangkurat University, Banjarbaru, Indonesia

ABSTRACT

The expansion of social media platforms, especially Twitter (now X), has made them valuable sources for capturing real-time public sentiment. This study addresses the challenge of accurately analyzing sentiment related to the Clash of Champions event in Indonesia by exploring various machine learning techniques. The primary goal is to evaluate the performance of three classification algorithms—Support Vector Machine (SVM), Random Forest, and Extreme Gradient Boosting (XGBoost)—when paired with two widely used text representation methods: Term Frequency-Inverse Document Frequency (TF-IDF) and Bag of Words (BoW). Key contributions of this study include the application of the Synthetic Minority Oversampling Technique (SMOTE) to address class imbalance and the use of GridSearchCV for systematic hyperparameter optimization. A dataset of 1,000 tweets was collected via web scraping, manually annotated, and processed using natural language processing techniques. The data was split into training and testing sets, with the models evaluated based on accuracy, precision, recall, F1 score, and AUC metrics. Results indicate that the Random Forest model with TF-IDF representation achieved the highest classification accuracy of 91%, while the XGBoost model with TF-IDF yielded the best AUC score of 0.91. In general, TF-IDF outperformed BoW in most cases, demonstrating its superior ability to capture more meaningful word patterns. SMOTE effectively addressed the data imbalance issue, and GridSearchCV contributed to optimal model parameter tuning. In conclusion, the research highlights the significant impact of selecting suitable feature extraction techniques and classification algorithms on the success of sentiment analysis for social media data. Future studies should consider incorporating contextual embedding models like Word2Vec, FastText, or transformer-based approaches such as BERT to enhance sentiment analysis accuracy.

PAPER HISTORY

Received Feb. 02, 2025
Accepted April 20, 2025
Published May 18, 2025

KEYWORDS

Sentiment analysis;
Machine learning;
Twitter data;
Feature extraction;
Text classification

CONTACT:

erhafiza96@gmail.com
triando.saragih@ulm.ac.id
muliadi@ulm.ac.id
f.indriani@ulm.ac.id
mazdadi@ulm.ac.id

1. INTRODUCTION

Sentiment represents the subjective expression of an individual's attitudes, opinions, or emotional responses toward a particular topic, product, or event [1]. Sentiment analysis is crucial for gaining insights into public opinion, helping guide strategic decisions in areas such as business intelligence, politics, and the assessment of public services. In the context of the digital age, social media has emerged as a prominent and prolific medium through which individuals articulate their sentiments in real time. Social media platforms are characterized by their dynamic and interactive nature, allowing users to generate, disseminate, and engage with diverse forms of content such as text, images, and videos [2]. The continuous and large-scale production of user-generated content positions social media as a valuable and abundant source of data for sentiment analysis.

Among the available platforms, Twitter (recently rebranded as X) is one of the most extensively utilized in public opinion research, owing to its concise textual format and real-time dissemination capabilities. Nevertheless, the high volume and velocity of data produced on such platforms render manual sentiment analysis impractical and inefficient [3]. Consequently, there is a growing need for automated methods capable of processing unstructured text data at scale. Natural Language Processing (NLP), a branch of artificial intelligence focused on allowing computers to comprehend and analyze human language, has emerged as a key method in this field [3]. By employing NLP techniques, unstructured content from social media can be systematically transformed into structured insights, thereby enhancing the efficacy of data-driven decision-making processes.

Corresponding author: Triando Hamonangan Saragih, triando.saragih@ulm.ac.id, Department of Computer Science, Lambung Mangkurat University, Jalan A. Yani Km 36, Banjarbaru 70714, Kalimantan Selatan, Indonesia.
DOI: https://doi.org/10.35882/ijeeemi.v7i2.81

Numerous researchers have proposed approaches for sentiment analysis, utilizing feature extraction methods like Term Frequency-Inverse Document Frequency (TF-IDF) and Bag of Words (BoW). TF-IDF calculates the importance of terms by comparing how often they appear in a specific document to how frequently they occur across the entire dataset, highlighting more meaningful words while reducing the weight of frequently used ones [4]. In contrast, BoW converts textual input into fixed-length vectors by simply counting word occurrences, disregarding syntax, word order, and semantic context [5]. Although both techniques have shown effectiveness in various natural language processing tasks, they remain limited in their ability to capture contextual relationships—an essential aspect when analyzing informal and nuanced expressions often found in social media platforms such as Twitter [4][5]. This limitation underlines the importance of systematically evaluating the performance of these feature extraction methods in real-world sentiment classification scenarios.

To overcome issues related to feature representation and enhance prediction performance, various machine learning algorithms have been investigated. Support Vector Machine (SVM) stands out for its effectiveness in managing high-dimensional and sparse text data, as well as its capability to create optimal hyperplanes for both binary and multiclass classification tasks [6]. At the same time, ensemble learning methods like Random Forest and Extreme Gradient Boosting (XGBoost) have shown strong performance by effectively modeling complex, non-linear patterns and reducing overfitting—Random Forest through bootstrapping techniques and XGBoost via gradient-based optimization strategies [7][8]. Prior studies emphasize that the interaction between feature extraction techniques and classifier choice significantly influences model effectiveness. Despite the advancements in sentiment analysis, there is a lack of thorough studies that assess the combined application of TF-IDF and BoW with classifiers like SVM, Random Forest, and XGBoost, especially in the context of Twitter sentiment analysis. This gap is what this research seeks to fill.

A key challenge in sentiment analysis is class imbalance, where certain sentiment categories are disproportionately represented. To tackle this problem, the Synthetic Minority Over-sampling Technique (SMOTE) is used to enhance the representation of underrepresented classes by generating synthetic samples through linear interpolation within those classes [9]. SMOTE has been shown to be effective in improving model accuracy on imbalanced data, as demonstrated in research related to analyzing user satisfaction with public services [10]. In addition, various studies have applied hyperparameter optimization using GridSearchCV to find the optimal parameter combination to improve model accuracy [11]. Nonetheless, additional studies are required to assess the impact of combining SMOTE with GridSearchCV on enhancing sentiment analysis performance, particularly with Twitter data.

Recent studies have emphasized the effectiveness of machine learning algorithms for sentiment analysis on social media platforms like Twitter. In analyzing tweets related to the Indian farmer protests, the SVM model was evaluated using two commonly applied feature extraction methods: Bag of Words (BoW) and TF-IDF. The results showed that SVM achieved an accuracy of 83.45% with BoW, while its accuracy slightly dropped to 83.04% with TF-IDF. These findings indicate that although SVM performed well in sentiment classification, there is room for improvement, especially when applying these feature extraction techniques, particularly TF-IDF [12]. Meanwhile, research on public awareness in Saudi Arabia during the COVID-19 pandemic through Arabic tweets showed that SVM combined with TF-IDF bigrams achieved the best accuracy of 85%, outperforming KNN which only reached 64% [13]. Another study involving transformer-based ensemble models for sentiment classification of Arabic dialects showed that SVM integrated with TF-IDF and tuned via GridSearchCV achieved an average accuracy of 85%, while XGBoost performed less effectively with an accuracy of 76.2% [14]. Together, these studies emphasize the effectiveness of combining TF-IDF with traditional machine learning models like SVM and Random Forest for precise sentiment classification in diverse linguistic and contextual contexts. The consistent outcomes across various languages and datasets demonstrate the robustness of TF-IDF as a feature extraction technique. Additionally, they suggest that careful tuning and preprocessing can further improve model performance in challenging sentiment analysis tasks.

The goal of this study is to implement and evaluate the performance of machine learning algorithms—SVM, Random Forest, and XGBoost—in sentiment analysis of Twitter data, focusing on the Clash of Champions event as a case study. To address data imbalance, the study utilizes the SMOTE technique and employs hyperparameter tuning through GridSearchCV to enhance model accuracy. Additionally, this research examines the impact of two feature extraction techniques, TF-IDF and Bag of Words (BoW), on sentiment analysis. Consequently, the outcomes of this study aim to provide a deeper understanding of the best methods for performing sentiment analysis on the Twitter platform.

Therefore, this research contributes as follows: 1) comparison of three algorithms namely SVM, Random Forest, and XGBoost to determine the best model, 2) Assessment of two feature extraction methods, TF-IDF and BoW, to enhance classification accuracy, 3) application of SMOTE to handle data imbalance, and 4) hyperparameter optimization using GridSearchCV to improve model performance. The collected Twitter dataset has gone through preprocessing, resampling, and division of training and testing data. The model was subsequently trained and fine-tuned using different parameter settings to enhance accuracy. Furthermore, this study introduces an evaluation framework to assess

Corresponding author: Triando Hamonangan Saragih, triando.saragih@ulm.ac.id, Department of Computer Science, Lambung Mangkurat University, Jalan A. Yani Km 36, Banjarbaru 70714, Kalimantan Selatan, Indonesia.
DOI: https://doi.org/10.35882/ijeeemi.v7i2.81

Indonesian Journal of Electronics, Electromedical Engineering, and Medical Informatics e-ISSN: 2656-8624
Homepage: https://ijeeemi.org/; Vol. 7, No. 2, pp. 370-386, May 2025

the influence of feature extraction methods and algorithms on sentiment analysis performance.

This study is organized as follows: Section II discusses the dataset used, research methods, and model training and testing stages. Section III presents the sentiment analysis results, including a comparison of the performance of the algorithms used. Section IV discusses the interpretation of the results, comparison with previous research, and limitations of this study. Finally, Section V contains conclusions that summarize the objectives, key findings, and recommendations for future research.

2. MATERIALS AND METHOD

This research compares Term Frequency-Inverse Document Frequency (TF-IDF) and Bag of Words (BoW) feature extraction methods using SVM, Random Forest, and XGBoost methods. The research methodology consists of several stages. First, dataset collection using web scraping technique with the keyword "Clash of Champions". Next, manual labeling of tweets, preprocessing, data splitting, data resampling, model evaluation, and comparative analysis of results are performed. The research flowchart can be seen in Fig. 1.

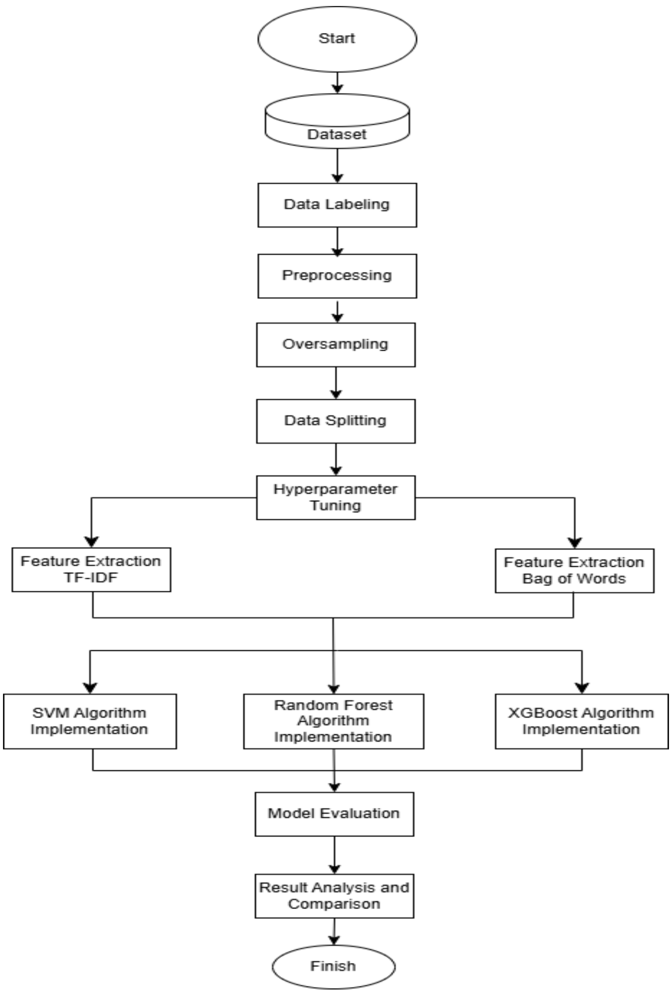


Fig. 1. Research Flowchart for Clash of Champions Sentiment Analysis with Machine Learning.

A. Dataset

This research utilizes a dataset collected from social media Twitter/X using web scraping techniques. The dataset consists of 1,000 tweets related to the "Clash of Champions" event, collected over a period of two weeks using the main keyword "Clash of Champions" as well as related hashtags such as #ClashOfChampions and #CoC. This dataset consists of 15 features, viz: conversation_id_str, created_at, favorite_count, full_text, id_str, image_url, in_reply_to_screen_name, lang, location, quote_count, reply_count, retweet_count, tweet_url, user_id_str, and username.

B. Data Labeling

The dataset was manually labeled to classify the sentiment conveyed in each tweet. During this process, each tweet was assigned to one of two sentiment categories: positive or negative. Tweets expressing favorable opinions—such as excitement, support, or praise for the event—were categorized as positive, while those indicating dissatisfaction, criticism, or negative emotions were labeled as negative [15].

The manual annotation was necessary to capture the informal language, sarcasm, and contextual nuances often found in Twitter posts, which automated tools may overlook. An example of the annotated data can be found in Table 1, with the sentiment class distribution presented in Fig. 2. Consistent annotation guidelines were adhered to throughout the process to maintain the accuracy and reliability of the labels, even though the task was carried out by a single annotator. This careful approach aimed to maintain the integrity of the dataset for further sentiment analysis tasks. The annotator relied on contextual understanding and prior knowledge of the event to interpret the sentiment accurately. Tweets that contained ambiguous or mixed sentiments were carefully reviewed before assigning a final label. Although the annotation was done individually, efforts were made to remain objective and consistent throughout the labeling process. The final labeled dataset serves as a crucial foundation for training and evaluating sentiment classification models.

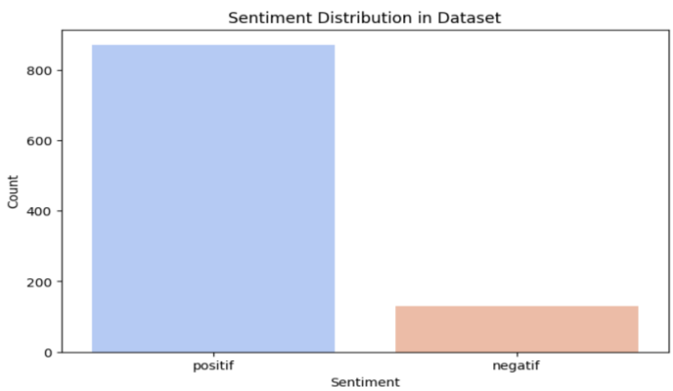


Fig. 2. Research Flowchart for Clash of Champions Sentiment Analysis with Machine Learning.

Corresponding author: Triando Hamonangan Saragih, triando.saragih@ulm.ac.id, Department of Computer Science, Lambung Mangkurat University, Jalan A. Yani Km 36, Banjarbaru 70714, Kalimantan Selatan, Indonesia.
DOI: https://doi.org/10.35882/ijeeemi.v7i2.81

Indonesian Journal of Electronics, Electromedical Engineering, and Medical Informatics e-ISSN: 2656-8624
Homepage: https://ijeeemi.org/; Vol. 7, No. 2, pp. 370-386, May 2025

Table 1. Clash of Champions-related Tweet Data with Sentiment and Interaction Information.

No	conversation_id_str	created_at	favorite_count	full_text	...	label
1	83054E+18	Mon Sep 02 09:38:11 +0000 2024	11	Guys gue jujurly terharu bgt sama kalian kalian yg udah donasi dan adopsi ï¸ benar clash of champions ini membawa dampak positif dan memotivasi banyak orang untuk melakukan kebaikan ï¸	...	positive
2	83016E+18	Mon Sep 02 01:21:42 +0000 2024	0	Tapi dari abis nonton clash of champions ini aku jadi lebih berambisi untuk mencapai mimpiku dan menjadi champion di bidang aku sendiri	...	positive
...	...	...	...	...	...	...
999	82258E+18	Sun Aug 11 10:34:52 +0000 2024	1	Nonton clash of champion tuh bukan cuma penasaran sama pesertanya yg knp pinter2 bgt tapi juga penasaran sama gimana cara orang tua ngedidik anaknya bisa sampe pinter kaya gitu. Nonton coc tuh sambil berdoa yaAllah semoga anak hamba bisa kaya mereka juga. Jad emaknya yg ambis	...	positive
1000	81814E+18	Tue Jul 30 04:09:00 +0000 2024	0	Pengen Jadi Mereka Yang Ada Di Clash of Champions? Siapa yang gak pengen jadi jagoan kayak para juara di Clash of Champions? Ternyata selain latihan fisik makan makanan yang bikin otak sehat juga penting banget loh! #clashofchampions #coc #otaksehat https://t.co/YQoQ1WWoRv	...	positive

C. Preprocessing

Preprocessing is a crucial stage in the sentiment analysis workflow, designed to convert raw text data into a well-structured and standardized format, making it appropriate for feature extraction and model training [16]. This stage plays a crucial role in normalizing text and eliminating irrelevant or noisy elements that may compromise model performance. Through this process, textual data is filtered and refined to retain only semantically meaningful content, thereby reducing redundancy in the vocabulary and improving the overall effectiveness of feature extraction. The preprocessing steps applied in this study consist of the following, with an example of the processed text data provided in Table 2.

1. Data Cleaning

At this stage, the dataset is cleaned by removing unnecessary attributes, so that only the “text” and “label” columns are retained for analysis purposes. In addition, the text is cleaned by removing HTML tags, punctuation marks, numbers, as well as other special characters.

2. Case Folding

Case folding involves transforming all alphabetic characters into lowercase to ensure consistency and minimize redundancy in text representation (e.g., "Event" and "event") [17].

3. Tokenization

Tokenization involves breaking down a sentence into smaller, meaningful units known as tokens, which usually represent individual words. This segmentation is typically done by using spaces and punctuation marks, like commas or periods, to allow for the separate analysis of each word in later processing steps [18].

4. Stopword Removal

Stopword removal is the process of removing frequently occurring words like “and,” “the,” or “is,” which appear often in text but provide little meaningful value for the analysis [19].

5. Stemming

Stemming involves simplifying words to their base or root form by removing prefixes or suffixes attached to the beginning or end of the word. This step is applied after stopword removal to further standardize the text and minimize vocabulary variation. In this study, stemming is implemented using the Sastrawi library in Python, which is specifically designed for processing texts in the Indonesian language [20].

Corresponding author: Triando Hamonangan Saragih, triando.saragih@ulm.ac.id, Department of Computer Science, Lambung Mangkurat University, Jalan A. Yani Km 36, Banjarbaru 70714, Kalimantan Selatan, Indonesia.
DOI: https://doi.org/10.35882/ijeeemi.v7i2.81

Table 2. Results of Preprocessing Text Data for Sentiment Analysis on Clash of Champions Tweets.

Preprocessing	Results
Basic Sentence	Clash Of Champions bukan hanya ajang kompetisi antara mahasiswa tetapi juga menghibur, memotivasi dan inspirasi untuk banyak orang dan juga mengajarkan apa itu arti persahabatan. Terimakasih untuk para cast CoC dan Tim Ruang Guru kalian luar biasa #ruanguru #ThankYouCoC
Data Cleaning	Clash Of Champions bukan hanya ajang kompetisi antara mahasiswa tetapi juga menghibur memotivasi dan inspirasi untuk banyak orang dan juga mengajarkan apa itu arti persahabatan Terimakasih untuk para cast CoC dan Tim Ruang Guru kalian luar biasa ruanguru ThankYouCoC
Case Folding	clash of champions bukan hanya ajang kompetisi antara mahasiswa tetapi juga menghibur memotivasi dan inspirasi untuk banyak orang dan juga mengajarkan apa itu arti persahabatan terimakasih untuk para cast coc dan tim ruang guru kalian luar biasa ruanguru thankyoucoc
Tokenization	['clash', 'of', 'champions', 'bukan', 'hanya', 'ajang', 'kompetisi', 'antara', 'mahasiswa', 'tetapi', 'juga', 'menghibur', 'memotivasi', 'dan', 'inspirasi', 'untuk', 'banyak', 'orang', 'dan', 'juga', 'mengajarkan', 'apa', 'itu', 'arti', 'persahabatan', 'terimakasih', 'untuk', 'para', 'cast', 'coc', 'dan', 'tim', 'ruang', 'guru', 'kalian', 'luar', 'biasa', 'ruanguru', 'thankyoucoc']
Stopword Removal	['clash', 'of', 'champions', 'ajang', 'kompetisi', 'antara', 'mahasiswa', 'menghibur', 'memotivasi', 'inspirasi', 'banyak', 'orang', 'mengajarkan', 'arti', 'persahabatan', 'terimakasih', 'cast', 'coc', 'tim', 'ruang', 'guru', 'luar', 'biasa', 'ruanguru', 'thankyoucoc']
Stemming	['clash', 'of', 'champion', 'ajang', 'kompetisi', 'antara', 'mahasiswa', 'hibur', 'motivasi', 'inspirasi', 'banyak', 'orang', 'ajar', 'arti', 'sahabat', 'terima', 'kasih', 'cast', 'coc', 'tim', 'ruang', 'guru', 'luar', 'biasa', 'ruanguru', 'thankyoucoc']

D. Oversampling

Oversampling is a frequently employed method to address class imbalance, a situation where the distribution of class labels in a dataset is significantly skewed. In these situations, the learning process tends to favor the majority class, while the minority class, due to its lower representation, may be mistakenly identified as noise or outliers. By applying oversampling, the class distribution in the dataset can be balanced to better reflect the analysis goals, enabling the model to learn more fairly across all classes and, as a result, generate more precise and dependable predictions [21].

1. Synthetic Minority Over-sampling Technique (SMOTE)

The Synthetic Minority Over-sampling Technique (SMOTE) is a popular approach used to tackle class imbalance by creating synthetic examples for the minority class. This method seeks to establish a more even distribution between the majority and minority classes, improving the model's capacity to learn from the less-represented data without merely replicating existing samples [22].

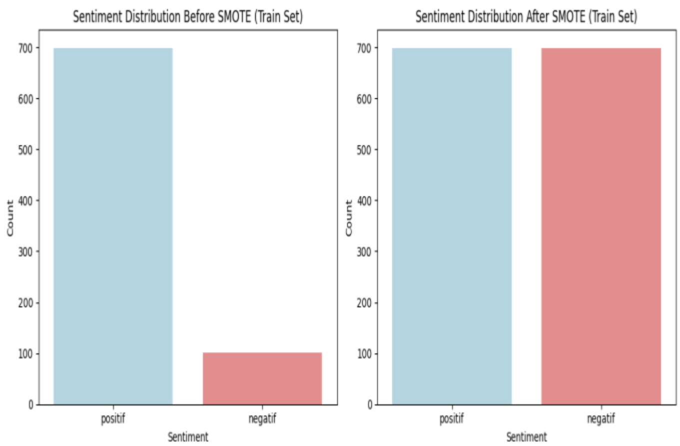


Fig. 3. Sentiment Distribution on Clash of Champions Tweets before and after Using SMOTE.

Corresponding author: Triando Hamonangan Saragih, triando.saragih@ulm.ac.id, Department of Computer Science, Lambung Mangkurat University, Jalan A. Yani Km 36, Banjarbaru 70714, Kalimantan Selatan, Indonesia.
DOI: https://doi.org/10.35882/ijeeemi.v7i2.81

In this study, the sentiment variables targeted for classification were found to have an unbalanced distribution. As shown in Fig. 3, before applying the SMOTE (Synthetic Minority Over-sampling Technique) method, the amount of data with positive sentiment was significantly greater than that with negative sentiment. This imbalance can result in a model that excels at detecting the majority class but has difficulty correctly identifying instances of the minority class.

It is a positive approach that this study applies SMOTE, a widely used oversampling technique designed to generate synthetic examples of the minority class rather than simply duplicating existing samples. This technique was chosen specifically for its effectiveness in improving classifier performance in datasets with high class imbalance. SMOTE works by locating the k-nearest neighbors of an instance from the minority class and creating new synthetic samples along the line segments connecting the original point and its neighbors, thus enhancing the feature space with varied yet plausible samples.

Before applying SMOTE, the dataset consisted of approximately 700 positive sentiment samples and only about 100 negative ones. This disproportionate ratio could potentially bias the model toward the majority class. After the application of SMOTE, the distribution became more balanced, with both positive and negative sentiment data totaling nearly 700 each, as shown in the right part of the figure.

The choice to implement SMOTE was driven by its effectiveness in tackling class imbalance by offering the classifier a more comprehensive representation of the minority class, thus improving generalization. Balancing the data is essential for improving the fairness and accuracy of the sentiment classification model. It helps prevent overfitting to the majority class while enhancing the model's capacity to learn from and generalize across the minority class. Consequently, the model is better positioned to deliver predictions that are both accurate and fair across different sentiment categories.

E. Split Data

Data splitting is the process of dividing the dataset into two primary subsets: the training data and the testing data. The training set is used to develop and fine-tune the classification model, while the testing set is used to assess the model's performance on new, unseen data, ensuring its capability to generalize well [22][23]. In this study, the dataset was divided in an 80/20 ratio, with 80% of the data (800 entries) used for training and 20% (200 entries) set aside for testing. This ratio was chosen because it provides a sufficient amount of data for the model to learn meaningful patterns while still preserving a reasonable portion for unbiased performance evaluation. Using a larger training set helps in building a more robust model, especially when the dataset is relatively small.

To improve the reliability of model evaluation, k-fold cross-validation was incorporated during the training process. In this technique, the training data is divided into k subsets (or folds), and the model is trained and validated k times, with each fold serving as the validation set once, while the other folds are used for training. This method helps mitigate the risk of overfitting and ensures that the model's performance is not overly reliant on any single subset of data. By combining an 80/20 data split with k-fold cross-validation, the study ensures a balanced approach to both model training and evaluation, thereby enhancing the robustness and generalizability of the final classification model.

F. Feature Extraction

Feature extraction is an essential step in the text classification process, where processed text data is converted into numerical formats that can be used as input for machine learning algorithms [24]. In this research, the feature extraction methods used are Term Frequency-Inverse Document Frequency (TF-IDF) and Bag of Words (BoW), which function to calculate the weight of each word based on its frequency in the document to represent words as numerical vectors. TF-IDF was chosen for its ability to assign weights to words according to their relevance in the corpus. It diminishes the impact of frequently occurring yet less informative terms while enhancing the importance of less common, more meaningful words. In the implementation of TF-IDF, normalization is applied to ensure that the term weights are well-scaled and comparable across different documents. Meanwhile, Bag of Words provides a straightforward frequency count of each word's appearance within a document. Although it disregards grammar and word order, BoW effectively captures the distribution of words and serves as a strong baseline method for various text classification tasks.

1. Term Frequency-Inverse Document Frequency (TF- IDF)

Term Frequency-Inverse Document Frequency (TF-IDF) is a word weighting technique in documents that works by considering the number of occurrences of words in a document as well as their distribution across documents in a data set. This method alters the weight of each word depending on its frequency in a document relative to its occurrence across all documents [5]. The TF-IDF calculation is made up of two key components: Term Frequency (TF) and Inverse Document Frequency (IDF), each having its own specific formula and purpose. These two values are then combined to determine the final TF-IDF score for each term in a document [25].

The first component, Term Frequency (TF), is measured by the following formula:

$$tf(t,d) = \frac{f_{t,d}}{\sum_{t' \epsilon d} f_{t',d}} \tag{1}$$

As shown on Eq. (1) [25], t is the term or word being analyzed, while d refers to a particular document. The

Corresponding author: Triando Hamonangan Saragih, triando.saragih@ulm.ac.id, Department of Computer Science, Lambung Mangkurat University, Jalan A. Yani Km 36, Banjarbaru 70714, Kalimantan Selatan, Indonesia.
DOI: https://doi.org/10.35882/ijeeemi.v7i2.81

notation f_{t},d denotes the frequency of occurrence of term t in document d. Meanwhile, $\sum_{t'} \in d\ f_{t'},d$ represents the total count of all terms in document d. TF reflects the frequency of a word's occurrence in a document relative to the total word count within that document. This measure helps evaluate the significance of the word in relation to the specific document.

The second component, Inverse Document Frequency (IDF), is determined using the following formula:

$$idf\ (t, D) = \log \frac{N}{|\{d \in : t \in d\}|} \tag{2}$$

In Eq. (2) [25], t refers to the analyzed term, D epresents the collection of documents (corpus), and N is the total number of documents in the corpus. Meanwhile, $|\{\mathrm{d} \in \mathrm{D}:\mathrm{t} \in \mathrm{d}\}|$ indicates the count of documents that contain the term t. The IDF value reflects how rare or unique a term is within the corpus; the fewer the number of documents that include the term, the higher its IDF value.

The combination of these two components results in a TF-IDF score, which is calculated using the formula:

$$tfidf\ (t, d, D) = tf(t, d) \times idf(t, D) \tag{3}$$

Finally, Eq. (3) shows the calculation is the result of multiplying the TF and IDF values [25]. The TF-IDF value ranges from 0 to 1, with a higher value indicating that the word is more relevant to a specific document [26] [27].

2. Bag of Words (BoW)

Bag-of-Words (BoW) is a basic technique for converting text into fixed-length vectors based on the frequency of words, disregarding the order and grammatical structure of the words [5]. This approach ignores the sequence and grammatical structure of words, concentrating only on whether terms appear or how often they occur within the text. Typically, the process involves building a vocabulary from the dataset, cleaning the text by removing punctuation and special characters, constructing a matrix that links documents to terms, and generating feature vectors that represent term frequency. Its popularity stems from its straightforward implementation and reliable performance in transforming text into a usable format for machine learning applications.

The Bag of Words (BoW) model ignores the context and order of words within a text. It treats each word occurrence independently without considering the relationship between words. As a result, BoW struggles to capture the semantic meaning of a text. However, it continues to be widely adopted because of its simplicity, ease of implementation, and effectiveness in various applications [28][29]. In mathematical notation BOW can be represented if d is a document and v is a set of words or vocabulary from the entire document which will be represented in Eq. (4).

$$\mathrm{BoW(d)} = [\mathrm{count(w1, d), count(w2, d \ldots\ count(wn, d)}] \tag{4}$$

As shown on Eq. (4) $\mathrm{count(wi, d)}$ is the number of occurrences of word wi in document d. while n is the number of words in V.

G. Hyperparameter Tuning

Hyperparameter tuning is an optimization process that aims to find the most optimal parameter values by first specifying a list of parameters to be tuned along with their value ranges. This is achieved by defining a set of hyperparameters along with their respective value ranges, and then systematically evaluating different combinations to determine the configuration that yields the best performance [30]. At this stage, hyperparameter adjustment is performed using the grid search method, which serves to test various parameter combinations and determine the configuration that produces the best performance.

1. GridSearchCV

GridSearchCV is a method provided by the scikit-learn library for performing exhaustive hyperparameter tuning by evaluating all possible combinations of specified parameter values [31]. This process involves training and validating a model for each configuration using cross-validation, thereby identifying the parameter set that yields the best performance [32]. In this study, GridSearchCV is employed for model training and hyperparameter selection, as illustrated in Fig. 4 [11].

The hyperparameter tuning process involved testing several algorithms, with the Support Vector Machine (SVM) model evaluated using parameters such as C ([0.1, 1, 10]), gamma ([1, 0.1, 0.01]), and kernel types (['linear', 'rbf']). These parameters were specifically chosen to adjust the complexity of the decision boundary, ensuring the model could generalize well on new, unseen data. This process also helps to avoid overfitting while maximizing the model's performance. The best configuration for the Bag of Words representation was determined to be C=10, gamma=1, and a 'linear' kernel, achieving an accuracy of 0.89. For the TF-IDF representation, the optimal setting was found to be C=10, gamma=1, and the 'rbf' kernel, leading to an accuracy of 0.97.

In Random Forest algorithm, the optimized parameters include n_estimators ([50, 100, 200]), max_depth ([10, 20, 30, None]), min_samples_split ([2, 5, 10]), and min_sam ples_leaf ([1, 2, 4]). For Bag of Words representation the best configuration is n_estimators=200, max_depth=Non e, min_samples_split=2, and min_samples_leaf=1 with an accuracy of 0.90. While in TF-IDF, the optimal configuration is n_estimators=200, max_depth=3 0, min_samples_split=2, and min_samples_leaf=1resulti ng in an accuracy of 0.96.

Corresponding author: Triando Hamonangan Saragih, triando.saragih@ulm.ac.id, Department of Computer Science, Lambung Mangkurat University, Jalan A. Yani Km 36, Banjarbaru 70714, Kalimantan Selatan, Indonesia.
DOI: https://doi.org/10.35882/ijeeemi.v7i2.81

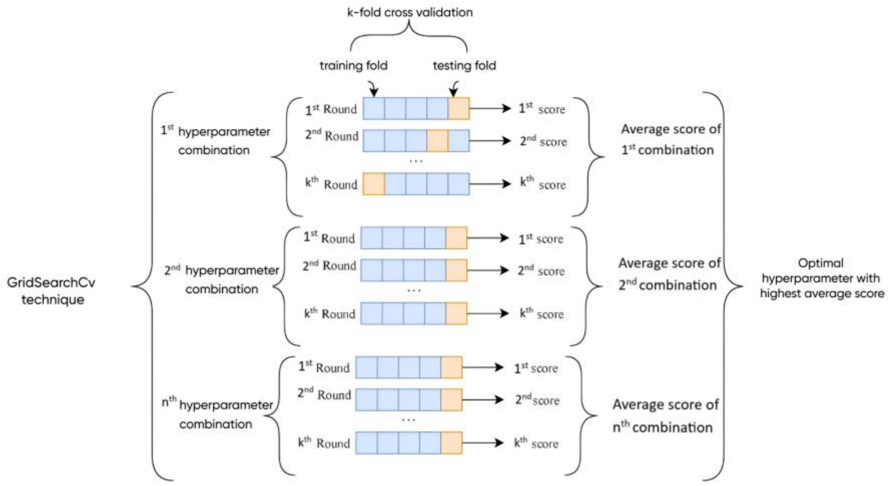


Fig. 4. Hyperparameter Tuning with GridSearchCV for Machine Learning Model Optimization.

Meanwhile, in XGBoost, the parameters tested were n_estimators ([50, 100, 200]), max_depth ([3, 6, 9]), learning_rate ([0.01, 0.1, 0.2]), and subsample ([0.8, 1.0]). The optimization resulted in the best configuration for Bag of Words which is learning_rate=0.01, max_depth=3, n_estimators=200, and subsample=1.0 with an accuracy of 0.89. In TF-IDF, the best configuration is learning_rate=0.01, max_depth=6, n_estimators=200, and subsample=1.0, with an accuracy of 0.91.

Explicit description of the tested parameters and their impact on model accuracy is helpful in understanding how the hyperparameter optimization process through GridSearchCV can significantly improve model performance. The parameters applied in GridSearch for each algorithm are displayed in Table 3.

Table 3. Comparison of GridSearchCV-Based Hyperparameter Optimization for Various Algorithms and Text Representations.

Algorithm	Feature Extraction	Best Parameter
Support Vector Machine	BoW	C=10, gamma=1, kernel='linear'
	TD-IDF	C=10, gamma=1, kernel='rbf'
Random Forest	BoW	n=200, depth=None, split=2, leaf=1
	TD-IDF	n=200, depth=30, split=2, leaf=1
XGBoost	BoW	lr=0.01, depth=3, n=200, subsample=1.0
	TD-IDF	lr=0.01, depth=6, n=200, subsample=1.0

H. Machine Learning Model

Machine learning, a key area within artificial intelligence, allows systems to analyze data using algorithms and make decisions by recognizing patterns in the data [33]. The learning process generally involves two main stages: training and testing. This approach focuses on creating algorithms that can enhance their performance by learning from data and experience. Machine learning is commonly divided into three primary categories: supervised learning, unsupervised learning, and reinforcement learning.

In this research, a range of machine learning algorithms are chosen for their demonstrated success in text classification tasks. The Support Vector Machine (SVM) is utilized because of its strong performance in high-dimensional spaces, making it especially effective for sparse text data [33]. Random Forest is selected for its strength in handling intricate data structures and its capacity to reduce overfitting by combining predictions from several decision trees [35]. Furthermore, Extreme Gradient Boosting (XGBoost) is employed due to its computational efficiency and exceptional accuracy, particularly in large-scale and highly competitive machine learning contexts [7].

1. Support Vector Machine (SVM)

Support Vector Machine (SVM) is a commonly employed advanced machine learning technique [7]. In machine learning, SVM is a supervised learning model that processes data using related learning techniques. A well-functioning SVM can classify data into distinct categories by identifying the boundary that separates the classes. These dividing lines are designed to reduce error by creating a maximum distance between the boundary and the class. SVM uses a hyperplane as a separator between classes, which is of three types: optimal, negative, and positive. The mathematical equations of each type of hyperplane are presented in Table 4. This ability to classify data accurately makes SVM a valuable tool in machine learning.

Corresponding author: Triando Hamonangan Saragih, triando.saragih@ulm.ac.id, Department of Computer Science, Lambung Mangkurat University, Jalan A. Yani Km 36, Banjarbaru 70714, Kalimantan Selatan, Indonesia.
DOI: https://doi.org/10.35882/ijeeemi.v7i2.81

Indonesian Journal of Electronics, Electromedical Engineering, and Medical Informatics e-ISSN: 2656-8624
Homepage: https://ijeeemi.org/; Vol. 7, No. 2, pp. 370-386, May 2025

Table 4. Equations and Types of Hyperplane in Support Vector Machine (SVM).

Equation	Hyperplane Type
$\vec{w}.\vec{x}+b=1$	positive
$\vec{w}.\vec{x}+b=-1$	negative
$\vec{w}.\vec{x}+b=0$	optimal

Shown in Table 4, w represents the margin width, x represents the bias, dan b is the feature.

In order for the model to obtain an optimal hyperplane, the margin width should be enlarged as much as possible. By utilizing a kernel function that projects the data into a higher-dimensional space, the algorithm becomes capable of addressing non-linear classification tasks more effectively. Commonly used kernels include the linear kernel and the Radial Basis Function (RBF) kernel [33].

2. Random Forest

Random Forest builds an ensemble of decision trees by randomly selecting splits and using bootstrap samples drawn with replacement from the training data. It represents a variant of the bagging technique, where class probabilities are determined by the proportion of votes cast by all trees within the ensemble [7][35]. The structural design of the Random Forest algorithm is illustrated in Fig. 5 [37].

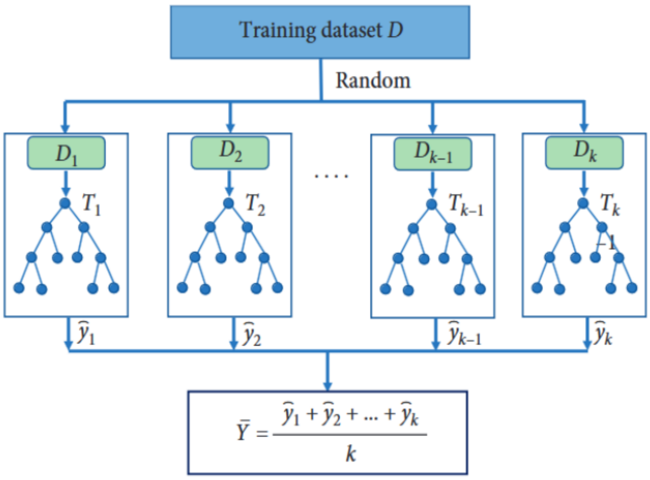


Fig. 5. Ensemble Process in Random Forest Algorithm Architecture for Accurate Prediction.

3. Extreme Gradient Boosting (XGBoost)

Extreme Gradient Boosting (XGBoost) is an ensemble-based approach that builds a powerful classifier by combining several decision tree models using gradient boosting techniques [7]. Unlike traditional boosting approaches, XGBoost incorporates a regularization component in its objective function to manage model complexity, prevent overfitting, and enhance generalization performance. The structure of each decision tree is optimized through a greedy algorithm that evaluates all possible feature split points [34]. The algorithm selects the split that maximizes the gain in the objective function relative to a single leaf node, while applying a threshold to prevent excessive tree depth. The objective function and complexity formula in XGBoost are shown in Eq. (5) and Eq. (6).

$$Obj^{(r)}=\sum_{i=1}^{m}L(y_i,\hat{y}_i^{(r)})+\sum_{k=1}^{r}\Omega(g_r) \tag{5}$$

As shown on Eq. (5) [37], the total objective function at iteration r consists of two main components. The variable m denotes the total number of training instances utilized. The initial component $L(y_i,\hat{y}_i^{(r)})$, represents the loss function that measures the discrepancy between the true value yi and the predicted value $\hat{y}_i^{(r)}$ at iteration r. Meanwhile, $\Omega(g_r)$ serves as the regularization term, assessing the complexity of the model within the k-th decision tree. This function is designed to balance predictive accuracy and model complexityThe algorithm incrementally adds decision trees throughout the training process to minimize the risk of overfitting and enhance the model's ability to generalize effectively to new, unseen data.

$$\Omega(g_r)=\gamma T+\frac{1}{2}\lambda\sum_{j}^{T}w^2 \tag{6}$$

In Eq. (6) [37], defines the regularization function, where T denotes the total number of leaves in a decision tree. The variable wj represents the weight of the j-th leaf, while γ serves as a penalty parameter applied to the number of leaves in the tree. Additionally, λ is a regularization coefficient used to control the magnitude of the weights in each leaf. The parameter γ is responsible for limiting tree growth to prevent excessive complexity, whereas λ helps stabilize the leaf weights. With these two components, XGBoost is able to produce models that are not only powerful in prediction but also efficient and robust against overfitting.

XGBoost evaluates the complexity of the model by considering the conventional error function. This algorithm divides the decision tree based on its level or depth. Each tree determines the optimal branches, relevant features, as well as the best threshold values. The growth of the tree structure occurs gradually through an iterative splitting process [38][39].

I. Evaluation

In this study, model performance is assessed using several evaluation metrics, including Area Under the

Corresponding author: Triando Hamonangan Saragih, triando.saragih@ulm.ac.id, Department of Computer Science, Lambung Mangkurat University, Jalan A. Yani Km 36, Banjarbaru 70714, Kalimantan Selatan, Indonesia.
DOI: https://doi.org/10.35882/ijeeemi.v7i2.81

Curve (AUC), Accuracy, Precision, Recall, and F1-Score. These metrics offer a well-rounded evaluation of how effectively the model performs sentiment classification. Utilizing multiple evaluation criteria provides deeper insights into various performance dimensions. For example, while Accuracy reflects the overall proportion of correct predictions, Precision and Recall help examine the trade-off between false positives and false negatives. The F1-Score, which harmonizes Precision and Recall into a single metric, is particularly valuable when dealing with imbalanced datasets.

The selection of evaluation metrics like Accuracy, AUC, Precision, Recall, and F1-Score is crucial for understanding model performance. The methodology section details how each metric is computed, all chosen to assess different performance aspects. These metrics are derived from the confusion matrix, which compares the model's predictions with the actual class labels. The matrix includes four key components: True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN). By analyzing these metrics, researchers ensure the model's consistent performance across criteria and reduce potential bias from using a single metric.

AUC: AUC evaluates the effectiveness of a model by measuring the area under the Receiver Operating Characteristic (ROC) curve, which represents the trade-off between the True Positive Rate and the False Positive Rate across different threshold settings. The AUC score ranges from 0 to 1, with higher values indicating stronger model performance. A visual representation of the AUC curve is presented in Fig. 6 [40].

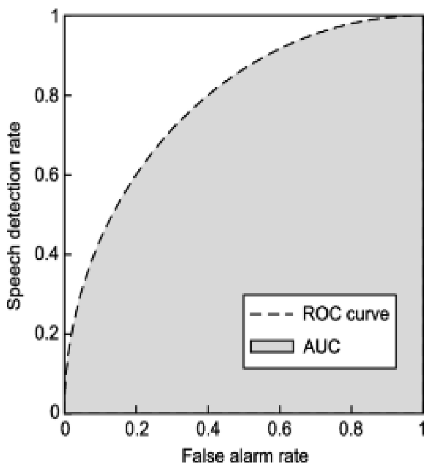


Fig. 6. Illustration of AUC Curve for Classification Model Performance Evaluation.

Accuracy: Indicates how closely the model's predictions align with the actual outcomes and is computed using the formula provided in Eq. (7) [15]:

$$Accuracy = \frac{TP+TN}{TP+TNFP+FN} \tag{7}$$

Precision: Indicates the model's ability to accurately classify positive instances among all instances predicted as positive. The formula for calculating precision is presented in Eq. (8) [41]:

$$Precision = \frac{TP}{TP+FP} \tag{8}$$

Recall: Recall reflects the model's effectiveness in identifying all actual positive instances. The formula used to compute recall is shown in Eq. (9) [41]:

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}} \tag{9}$$

F1-Score: represents the harmonic average of precision and recall, offering a balanced evaluation of both metrics, particularly useful when dealing with imbalanced datasets. The formula for F1-Score is provided in Eq. (10) [41]:

$$F1 = \frac{TP}{TP+0.5(FP+FN)} \tag{10}$$

Description:

TP = The count of correctly predicted positive instances.
TN = The number of correctly identified negative instances.
FP = The number of negative instances that were incorrectly predicted as positive.
FN = The number of positive instances that were incorrectly classified as negative.

Utilizing the confusion matrix to derive these metrics offers not only a numerical assessment of the model's performance but also enhances the interpretability of the results in a more insightful manner.

3. RESULTS

This section presents the evaluation of three classification algorithms—Support Vector Machine (SVM), Random Forest, and Extreme Gradient Boosting (XGBoost)—using two feature extraction techniques, Term Frequency-Inverse Document Frequency (TF-IDF) and Bag of Words (BoW), for sentiment analysis on Twitter regarding the Clash of Champions event. The study aims to assess the effectiveness of combining these classification algorithms with the feature extraction methods to achieve optimal performance.

To address data imbalance, the Synthetic Minority Over-sampling Technique (SMOTE) was applied. Additionally, hyperparameter tuning was performed using GridSearchCV to enhance model accuracy. Performance evaluation was conducted based on multiple metrics, including accuracy, precision, recall, F1-score, and the Area Under the Curve (AUC) of the Receiver Operating Characteristic (ROC) curve, to measure the effectiveness of each combination of algorithm and feature extraction technique in sentiment classification. This comprehensive evaluation ensures that the model's performance is robust and reliable across various evaluation criteria.

Corresponding author: Triando Hamonangan Saragih, triando.saragih@ulm.ac.id, Department of Computer Science, Lambung Mangkurat University, Jalan A. Yani Km 36, Banjarbaru 70714, Kalimantan Selatan, Indonesia.
DOI: https://doi.org/10.35882/ijeeemi.v7i2.81

Indonesian Journal of Electronics, Electromedical Engineering, and Medical Informatics e-ISSN: 2656-8624
Homepage: https://ijeeemi.org/; Vol. 7, No. 2, pp. 370-386, May 2025

Table 5. Performance Matrix Results of Classification Models in Twitter Sentiment Analysis with TF-IDF and Bag of Words Feature Extraction Techniques.

Model	Feature Extraction	Performance Matrix				
		Accuracy	Precision	Recall	F1-Score	AUC
SVM	TF-IDF	0.89	0.88	0.89	0.88	0.86
	Bag of Words	0.81	0.87	0.81	0.83	0.77
Random Forest	TF-IDF	0.91	0.91	0.91	0.91	0.90
	Bag of Words	0.83	0.90	0.83	0.85	0.86
XGBoost	TF-IDF	0.90	0.91	0.90	0.90	0.91
	Bag of Words	0.90	0.91	0.90	0.90	0.86

As seen in Table 5, the findings indicate that the Random Forest model with TF-IDF achieved the highest accuracy of 0.91, suggesting that it had the lowest misclassification rate compared to the other models. Furthermore, the XGBoost model with TF-IDF recorded the highest AUC score of 0.91, demonstrating its superior ability to distinguish between positive and negative sentiments. Although the SVM model with TF-IDF attained a slightly lower accuracy of 0.89, it still exhibited competitive performance in sentiment classification.

Overall, TF-IDF outperformed BoW in improving classification accuracy. Models utilizing BoW generally exhibited lower accuracy, with the highest accuracy being 0.90 in the XGBoost model, while other models yielded lower values. These results indicate that TF-IDF is more effective in extracting meaningful features from text than BoW, particularly in sentiment analysis on this dataset.

Evaluation of performance metrics, including precision, recall, and F1-score, revealed that the Random Forest and XGBoost models achieved more balanced and stable performance compared to SVM. The Random Forest model with TF-IDF attained precision, recall, and F1-score values of 0.91, signifying its consistency in correctly identifying positive and negative sentiments. Additionally, the XGBoost model with TF-IDF demonstrated strong classification ability, achieving an accuracy of 0.90 and the highest AUC of 0.91. Although the SVM model with TF-IDF obtained an accuracy of 0.89, it still performed effectively in sentiment classification. These findings underscore the significance of selecting appropriate feature extraction techniques and classification algorithms for enhancing sentiment analysis performance on social media data.

As shown in Fig. 7, This study evaluated the performance of classification models by testing three algorithms—SVM, Random Forest, and XGBoost—paired with two feature extraction methods: TF-IDF and Bag of Words (BoW). A bar chart presents five key evaluation metrics—accuracy, precision, recall, F1 score, and AUC—for each model-feature extraction combination. Overall, the results indicate that models using the TF-IDF technique generally outperform those using BoW across most metrics. Specifically, the Random Forest model with TF-IDF achieved the highest accuracy and F1 score, while XGBoost with TF-IDF achieved the highest AUC. These results highlight the importance of selecting suitable feature extraction methods, such as TF-IDF, to enhance model performance in sentiment analysis tasks.

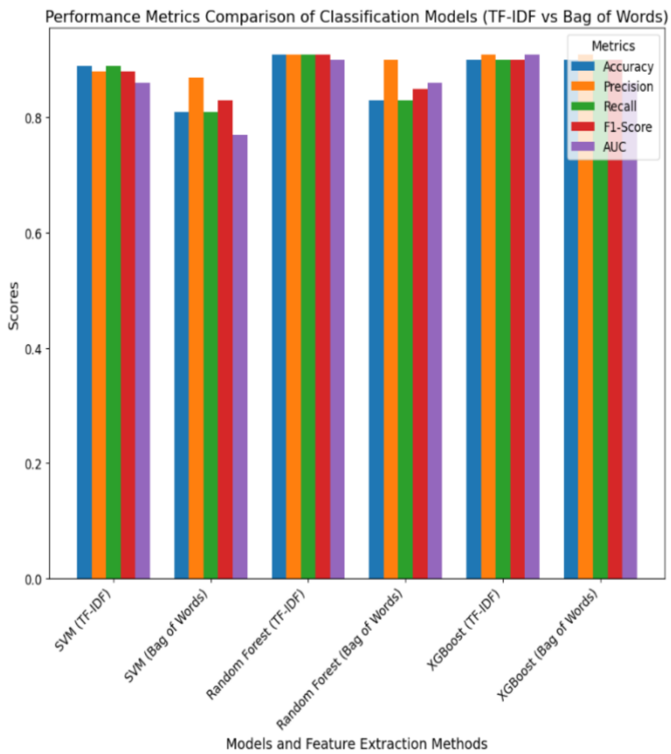


Fig. 7. Performance Comparison of SVM, Random Forest, and XGBoost Models with TF-IDF and Bag of Words Feature Extraction Techniques.

Corresponding author: Triando Hamonangan Saragih, triando.saragih@ulm.ac.id, Department of Computer Science, Lambung Mangkurat University, Jalan A. Yani Km 36, Banjarbaru 70714, Kalimantan Selatan, Indonesia.
DOI: https://doi.org/10.35882/ijeeemi.v7i2.81

4. DISCUSSION

This study emphasizes the critical role of choosing the right feature extraction techniques and classification algorithms in sentiment analysis tasks. The Term Frequency-Inverse Document Frequency (TF-IDF) method consistently outperformed the Bag of Words (BoW) technique across all models, highlighting its superior capacity to capture the significance and contextual meaning of words. Among the classifiers, the Random Forest model with TF-IDF achieved the highest accuracy of 91%, showcasing the effectiveness of ensemble methods in detecting complex patterns in text data. On the other hand, the XGBoost model attained the highest AUC score of 0.91, demonstrating its ability to accurately differentiate sentiment classes, even in datasets with imbalances or noise.

A more detailed analysis of the performance metrics reinforces these conclusions. In addition to high accuracy, the models demonstrated well-balanced precision, recall, and F1-scores. For example, the SVM model with TF-IDF achieved a recall of 0.90, precision of 0.88, and an F1-score of 0.89. These results highlight the model's effective management of classification errors, such as false positives and false negatives, which is particularly important when working with imbalanced class distributions in public opinion analysis.

Table 6. Comparison of Performance Metrics in Sentiment Analysis Models.

Study	Method & Model	Accuracy (%)
Gupta et al. [12]	SVM + BoW	83.45%
	SVM + TF-IDF	83,04%
Aljameel et al. [13]	SVM + TF-IDF	85%
	KNN + TF-IDF	64%
Mansour et al. [14]	SVM+ TF-IDF	85%
	XGBoost + TF-IDF	76.20%
This Study (2025)	SVM + BoW	81%
	SVM + TF-IDF	89%
	Random Forest + BoW	83%
	Random Forest + TF-IDF	91%
	XGBoost + BoW	90%
	XGBoosst + TF-IDF	90%

When compared to previous studies, the results of this research show a consistent improvement in sentiment classification performance. Gupta et al. [12] reported that Random Forest achieved the highest accuracy of 96.62% with BoW and 95.51% with TF-IDF, while SVM performed lower with 83.45% (BoW) and 83.04% (TF-IDF). Aljameel et al. [13] demonstrated that SVM with TF-IDF achieved 85% accuracy, outperforming KNN which only reached 64%. Similarly, Mansour et al. [14] showed that SVM with TF-IDF obtained an accuracy of 85%, while XGBoost performed lower at 76.20%. In this study, the combination of Random Forest and TF-IDF produced the highest accuracy at 91%, followed closely by XGBoost at 90%. These results, as shown in Table 6, confirm that the integration of effective feature extraction (TF-IDF), data balancing (SMOTE), and algorithm optimization (GridSearchCV) significantly contributes to improved model performance in sentiment analysis tasks.

The research also reveals performance variability across sentiment classes. The Random Forest model demonstrated high accuracy in classifying positive sentiments, with the fewest false negatives. Conversely, XGBoost maintained consistent performance across all sentiment types, particularly excelling in identifying neutral sentiments, which are often linguistically ambiguous. Although competitive, the SVM model showed a higher tendency to misclassify positive sentiments when using BoW, indicating its sensitivity to the chosen feature representation method.

Regarding the BoW approach, the XGBoost model delivered the best results, as reflected by a higher number of true positives (162) and fewer false positives and negatives. In contrast, the SVM model using BoW produced more false negatives (30), indicating a higher likelihood of misclassifying positive sentiments as negative. The Random Forest model outperformed SVM in this context by generating fewer false negatives. As shown in Fig. 8, XGBoost was the top performer using BoW. With the TF-IDF method, all models improved in performance. The SVM model, for example, reduced its false negatives to 9 and increased true positives to 165. Likewise, the Random Forest model with TF-IDF saw fewer classification errors than with BoW. XGBoost maintained stable performance, consistently achieving high true positives (162) and low error rates.

From a computational efficiency standpoint, each model demonstrated distinct trade-offs. While Random Forest delivered strong accuracy, it required more training time and computational resources, making it less suitable for real-time applications. XGBoost provided a balanced solution, offering high performance with moderate training time especially when hyperparameters were properly tuned. In contrast, the Support Vector Machine (SVM) showed faster training times but required careful parameter selection to achieve competitive performance.

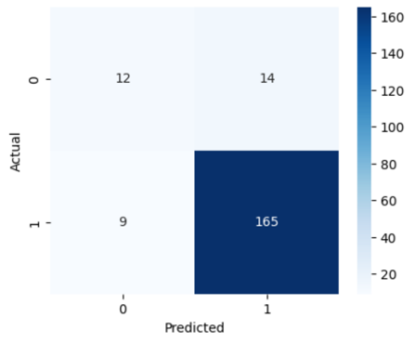
An analysis of misclassification cases also offered valuable insights into the limitations of current approaches. Errors were often found in tweets containing sarcasm, slang, or implicit sentiments that frequency-based methods like TF-IDF and BoW struggled to capture. For instance, sarcastic remarks that appear positive on the surface but convey negative meaning were frequently misclassified. These findings highlight the need

Corresponding author: Triando Hamonangan Saragih, triando.saragih@ulm.ac.id, Department of Computer Science, Lambung Mangkurat University, Jalan A. Yani Km 36, Banjarbaru 70714, Kalimantan Selatan, Indonesia.
DOI: https://doi.org/10.35882/ijeeemi.v7i2.81

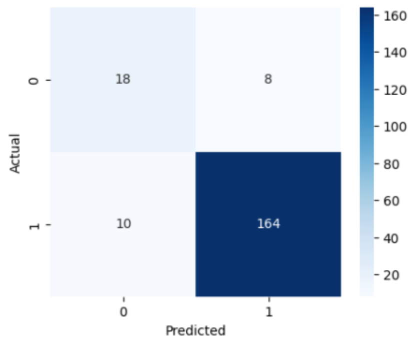
to explore more sophisticated contextual representation techniques in future studies, such as Word2Vec, FastText, or transformer-based models like BERT.

In terms of real-world applications, this study's findings show strong potential across various domains. In business intelligence, accurate sentiment models can help companies understand consumer perception and respond proactively. In political monitoring, these models can be used to track public opinion on key issues or political figures in real time. In the context of social media analytics, combining TF-IDF with ensemble algorithms like Random Forest and XGBoost has proven effective for rapidly and accurately classifying public sentiment, thus supporting data-driven decision-making across sectors.

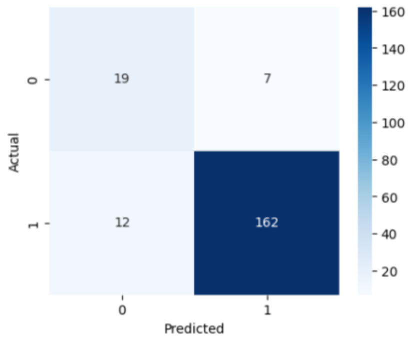
Overall, this research demonstrates that optimizing the combination of feature extraction techniques and classification algorithms not only improves model accuracy but also leads to a more adaptive, context-aware approach to sentiment analysis. These findings lay a solid foundation for future advancements in sophisticated text analytics systems.



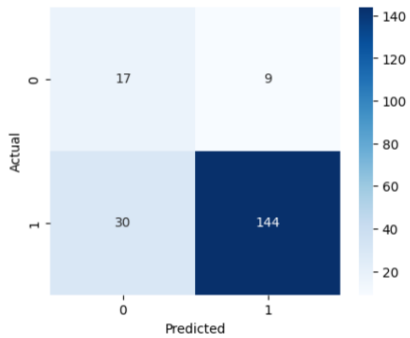
(a) SVM with TF-IDF



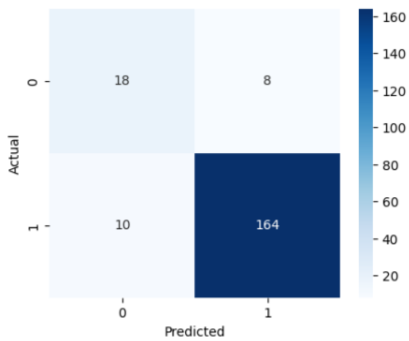
(b) Random Forest with TF-IDF



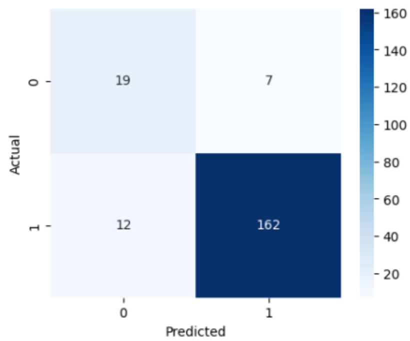
(c) XGBoost with TF-IDF



(d) SVM with Bag of Words



(e) Random Forest with Bag of Words



(f) XGBoost with Bag of Words

Fig. 8. Confusion matrix for the classification models using different feature extraction methods. Subfigure (a) shows the confusion matrix for SVM with TF-IDF, (b) represents Random Forest with TF-IDF, (c) displays XGBoost with TF-IDF, (d) shows SVM with Bag of Words, (e) represents Random Forest with Bag of Words, and (f) displays XGBoost with Bag of Words.

Corresponding author: Triando Hamonangan Saragih, triando.saragih@ulm.ac.id, Department of Computer Science, Lambung Mangkurat University, Jalan A. Yani Km 36, Banjarbaru 70714, Kalimantan Selatan, Indonesia.
DOI: https://doi.org/10.35882/ijeeemi.v7i2.81

5. CONCLUSION

The primary objective of this study was to evaluate the effectiveness of three popular machine learning classification algorithms—Support Vector Machine (SVM), Random Forest, and Extreme Gradient Boosting (XGBoost)—in analyzing sentiment from Twitter (X) data related to the Clash of Champions event. In conjunction with this, the research also aimed to compare two prominent feature extraction techniques, Term Frequency-Inverse Document Frequency (TF-IDF) and Bag of Words (BoW), to determine their influence on classification performance. The experimental findings demonstrated that the combination of TF-IDF with the Random Forest algorithm yielded the highest overall performance, achieving an accuracy, precision, recall, and F1-score of 91%, while the XGBoost model paired with TF-IDF achieved the highest AUC score of 0.91 and an accuracy of 90%. The SVM model with TF-IDF also performed commendably, attaining an accuracy of 89%. Conversely, models utilizing the BoW technique generally performed less effectively, with accuracy ranging from 81% to 90%, underscoring TF-IDF's superior capability in capturing the semantic significance of textual data. Moreover, the integration of the Synthetic Minority Oversampling Technique (SMOTE) effectively addressed class imbalance, and hyperparameter optimization using GridSearchCV contributed to significant performance improvements across all models. These results highlight the critical importance of selecting the appropriate combination of text representation methods and classification algorithms for successful sentiment analysis. For future work, it is recommended to incorporate more context-aware and semantically rich embedding techniques such as Word2Vec and FastText, which could capture deeper linguistic nuances in social media text. Furthermore, leveraging transformer-based models like Bidirectional Encoder Representations from Transformers (BERT) has the potential to significantly enhance classification performance by capturing contextual dependencies and subtle sentiment cues. Expanding the dataset to include multilingual content or data from various social media platforms may also improve model generalizability and ensure broader applicability across diverse domains and user populations.

REFERENCES

[1] Q. A. Xu, V. Chang, and C. Jayne, "A systematic review of social media-based sentiment analysis: Emerging trends and challenges," Decision Analytics Journal, vol. 3, p. 100073, Jun. 2022, doi: 10.1016/J.DAJOUR.2022.100073.

[2] K. L. Tan, C. P. Lee, K. S. M. Anbananthen and K. M. Lim, "RoBERTa-LSTM: A Hybrid Model for Sentiment Analysis With Transformer and Recurrent Neural Network," in IEEE Access, vol. 10, pp. 21517-21525, 2022, doi: 10.1109/ACCESS.2022.3152828.

[3] Albladi, Aish & Islam, Minarul & Seals, Cheryl. (2025). Sentiment Analysis of Twitter Data Using NLP Models: A Comprehensive Review. IEEE Access. PP. 1-1. 10.1109/ACCESS.2025.3541494.

[4] N. C. Dang, M. N. Moreno-García, and F. de la Prieta, "Sentiment analysis based on deep learning: A comparative study," Electronics (Switzerland), vol. 9, no. 3, Mar. 2020, doi: 10.3390/electronics9030483.

[5] Akuma, S., Lubem, T. & Adom, I.T. Comparing Bag of Words and TF-IDF with different models for hate speech detection from live tweets. Int. j. inf. tecnol. 14, 3629–3635 (2022). https://doi.org/10.1007/s41870-022-01096-4.

[6] Kaur, H., Ahsaan, S.U., Alankar, B. et al. A Proposed Sentiment Analysis Deep Learning Algorithm for Analyzing COVID-19 Tweets. Inf Syst Front 23, 1417–1429 (2021). https://doi.org/10.1007/s10796-021-10135-7.

[7] Y. Huang, R. Wang, B. Huang, B. Wei, S. L. Zheng and M. Chen, "Sentiment Classification of Crowdsourcing Participants' Reviews Text Based on LDA Topic Model," in IEEE Access, vol. 9, pp. 108131-108143, 2021, doi: 10.1109/ACCESS.2021.3101565.

[8] R. Zhang, Y. Yu, J. Shen, X. Cui, and C. Zhang, "Local Boosting for Weakly-Supervised Learning," in Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, New York, NY, USA: ACM, Aug. 2023, pp. 3364–3375. doi: 10.1145/3580305.3599417.

[9] S. Wang, Y. Dai, J. Shen, and J. Xuan, "Research on expansion and classification of imbalanced data based on SMOTE algorithm," Sci Rep, vol. 11, no. 1, p. 24039, Dec. 2021, doi: 10.1038/s41598-021-03430-5.

[10] M. Hadwan, M. Al-Sarem, F. Saeed, and M. A. Al-Hagery, "An Improved Sentiment Classification Approach for Measuring User Satisfaction toward Governmental Services' Mobile Apps Using Machine Learning Methods with Feature Engineering and SMOTE Technique," Applied Sciences, vol. 12, no. 11, p. 5547, May 2022, doi: 10.3390/app12115547.

[11] Z. M. Alhakeem, Y. M. Jebur, S. N. Henedy, H. Imran, L. F. A. Bernardo, and H. M. Hussein, "Prediction of Ecofriendly Concrete Compressive Strength Using Gradient Boosting Regression Tree Combined with GridSearchCV Hyperparameter-Optimization Techniques," Materials, vol. 15, no. 21, p. 7432, Oct. 2022, doi: 10.3390/ma15217432.

[12] A. S. Neogi, K. A. Garg, R. K. Mishra, dan Y. K. Dwivedi, "Sentiment analysis and classification of Indian farmers' protest using twitter data," *Int. J. Inf. Manag. Data Insights*, vol. 1, 100019, 2021, doi: 10.1016/j.jjimei.2021.100019.

Corresponding author: Triando Hamonangan Saragih, triando.saragih@ulm.ac.id, Department of Computer Science, Lambung Mangkurat University, Jalan A. Yani Km 36, Banjarbaru 70714, Kalimantan Selatan, Indonesia.
DOI: https://doi.org/10.35882/ijeeemi.v7i2.81

[13] S. S. Aljameel, D. A. Alabbad, N. A. Alzahrani, S. M. Alqarni, F. A. Alamoudi, L. M. Babili, S. K. Aljaafary, dan F. M. Alshamrani, "A Sentiment Analysis Approach to Predict an Individual's Awareness of the Precautionary Procedures to Prevent COVID-19 Outbreaks in Saudi Arabia," *Int. J. Environ. Res. Public Health*, vol. 18, no. 1, p. 218, Dec. 2020, doi: 10.3390/ijerph18010218.

[14] O. Mansour, E. Aboelela, R. Talaat, dan M. Bustami, "Transformer-based ensemble model for dialectal Arabic sentiment classification," *PeerJ Comput. Sci.*, vol. 11, e2644, Mar. 2025, doi: 10.7717/peerj-cs.2644.

[15] Noryasminda, Saragih, T. H., Herteno, R., Faisal, M. R., & Farmadi, A. (2025). Analysis of the Effect of Feature Extraction on Sentiment Analysis using BiLSTM: Monkeypox Case Study on X/Twitter. Indonesian Journal of Electronics, Electromedical Engineering, and Medical Informatics, 7(2), 344–357. https://doi.org/10.35882/ijeeemi.v7i2.88.

[16] L. Afuan and N. Hidayat, "Sentiment Analysis of the Kampus Merdeka Program on Twitter Using Support Vector Machine and a Feature Extraction Comparison: TF-IDF vs. FastText," Journal of Applied Data Sciences, vol. 5, no. 4, pp. 1738–1753, Dec. 2024, doi: 10.47738/jads.v5i4.436.

[17] Rianto, A. B. Mutiara, E. P. Wibowo, and P. I. Santosa, "Improving the accuracy of text classification using stemming method, a case of non-formal Indonesian conversation," J Big Data, vol. 8, no. 1, p. 26, Dec. 2021, doi: 10.1186/s40537- 021-00413-1.

[18] S. H. Imanuddin, K. Adi, and R. Gernowo, "Sentiment Analysis on Satusehat Application Using Support Vector Machine Method," Journal of Electronics, Electromedical Engineering, and Medical Informatics, vol. 5, no. 3, pp. 143–149, Jul. 2023, doi: 10.35882/jeemi.v5i3.304.

[19] A. A. Aladeemy et al., "Advancements and challenges in Arabic sentiment analysis: A decade of methodologies, applications, and resource development," Heliyon, vol. 10, no. 21, p. e39786, Nov. 2024, doi: 10.1016/j.heliyon.2024.e39786.

[20] M. Qorib, T. Oladunni, M. Denis, E. Ososanya, and P. Cotae, "Covid-19 vaccine hesitancy: Text mining, sentiment analysis and machine learning on COVID-19 vaccination Twitter dataset," Expert Systems with Applications, vol. 212, Feb. 2023, doi: 10.1016/j.eswa.2022.118715.

[21] R. Obiedat et al., "Sentiment Analysis of Customers' Reviews Using a Hybrid Evolutionary SVM-Based Approach in an Imbalanced Data Distribution," IEEE Access, vol. 10, pp. 22260–22273, 2022, doi: 10.1109/ACCESS.2022.3149482.

[22] M. M. Febrian, S. W. Saputro, T. H. Saragih, F. Abadi, and R. Herteno, "Hybrid Feature Selection and Balancing Data Approach for Improved Software Defect Prediction," Indonesian Journal of Electronics, Electromedical Engineering, and Medical Informatics, vol. 7, no. 2, pp. 232–244, Apr. 2025, doi: 10.35882/ijeeemi.v7i2.67.

[23] S. Gupta, K. Saluja, A. Goyal, A. Vajpayee, and V. Tiwari, "Comparing the performance of machine learning algorithms using estimated accuracy," Measurement: Sensors, vol. 24, p. 100432, Dec. 2022, doi: 10.1016/j.measen.2022.100432.

[24] K. L. Tan, C. P. Lee, and K. M. Lim, "A Survey of Sentiment Analysis: Approaches, Datasets, and Future Research," Applied Sciences (Switzerland), vol. 13, no. 7, Apr. 01, 2023, doi: 10.3390/app13074550.

[25] F. Rustam, M. Khalid, W. Aslam, V. Rupapara, A. Mehmood, and G. S. Choi, "A performance comparison of supervised machine learning models for Covid-19 tweets sentiment analysis," PLoS ONE, vol. 16, no. 2, Feb. 2021, doi: 10.1371/journal.pone.0245909.

[26] M. T. Mohammed and O. F. Rashid, "Document retrieval using term term frequency inverse sentence frequency weighting scheme," Indonesian Journal of Electrical Engineering and Computer Science, vol. 31, no. 3, p. 1478, Sep. 2023, doi: 10.11591/ijeecs.v31.i3.pp1478-1485.

[27] E. Heikel and L. Espinosa-Leal, "Indoor Scene Recognition via Object Detection and TF-IDF," J Imaging, vol. 8, no. 8, p. 209, Jul. 2022, doi: 10.3390/jimaging8080209.

[28] A. Fatima, Y. Li, T. T. Hills, and M. Stella, "Dasentimental: Detecting depression, anxiety, and stress in texts via emotional recall, cognitive networks, and machine learning," Big Data and Cognitive Computing, vol. 5, no. 4, Dec. 2021, doi: 10.3390/bdcc5040077.

[29] P. Cichosz, "Active Learning for Biomedical Article Classification with Bag of Words and FastText Embeddings," Applied Sciences, vol. 14, no. 17, p. 7945, Sep. 2024, doi: 10.3390/app14177945.

[30] E. Elgeldawi, A. Sayed, A. R. Galal, and A. M. Zaki, "Hyperparameter tuning for machine learning algorithms used for arabic sentiment analysis," Informatics, vol. 8, no. 4, Dec. 2021, doi: 10.3390/informatics8040079.

[31] N. Chandrasekhar and S. Peddakrishna, "Enhancing Heart Disease Prediction Accuracy through Machine Learning Techniques and Optimization," Processes, vol. 11, no. 4, Apr. 2023, doi: 10.3390/pr11041210.

[32] C. Sitaula, A. Basnet, A. Mainali, and T. B. Shahi, ''Deep learning-based methods for sentiment analysis on nepali COVID-19-related Tweets,'' Comput. Intell. Neurosci., vol. 2021, pp. 1–11, Nov. 2021.

[33] J. Cervantes, F. Garcia-Lamont, L. Rodríguez-Mazahua, and A. Lopez, "A comprehensive survey on support vector machine classification: Applications, challenges and trends,"

Corresponding author: Triando Hamonangan Saragih, triando.saragih@ulm.ac.id, Department of Computer Science, Lambung Mangkurat University, Jalan A. Yani Km 36, Banjarbaru 70714, Kalimantan Selatan, Indonesia.
DOI: https://doi.org/10.35882/ijeeemi.v7i2.81

Neurocomputing, vol. 408, pp. 189–215, Sep. 2020, doi: 10.1016/j.neucom.2019.10.118.

[34] S. Bengesi, T. Oladunni, R. Olusegun, and H. Audu, "A Machine Learning-Sentiment Analysis on Monkeypox Outbreak: An Extensive Dataset to Show the Polarity of Public Opinion From Twitter Tweets," IEEE Access, vol. 11, pp. 11811–11826, 2023, doi: 10.1109/ACCESS.2023.3242290.

[35] M. Qorib, T. Oladunni, M. Denis, E. Ososanya, and P. Cotae, "Covid-19 vaccine hesitancy: Text mining, sentiment analysis and machine learning on COVID-19 vaccination Twitter dataset," Expert Systems with Applications, vol. 212, Feb. 2023, doi: 10.1016/j.eswa.2022.118715.

[36] H.-V. T. Mai, T.-A. Nguyen, H.-B. Ly, and V. Q. Tran, "Prediction Compressive Strength of Concrete Containing GGBFS using Random Forest Model," Advances in Civil Engineering, vol. 2021, no. 1, p. 12, Jan. 2021, doi: 10.1155/2021/6671448.

[37] G. Huang, Z. Liu, Y. Wang, and Y. Yang, "A Multi-Objective Prediction XGBoost Model for Predicting Ground Settlement, Station Settlement, and Pit Deformation Induced by Ultra-Deep Foundation Construction," Buildings, vol. 14, no. 9, p. 2996, Sep. 2024, doi: 10.3390/buildings14092996.

[38] R. Szczepanek, "Daily Streamflow Forecasting in Mountainous Catchment Using XGBoost, LightGBM and CatBoost," Hydrology, vol. 9, no. 12, p. 226, Dec. 2022, doi: 10.3390/hydrology9120226.

[39] Z. Gu, M. Cao, C. Wang, N. Yu, and H. Qing, "Research on Mining Maximum Subsidence Prediction Based on Genetic Algorithm Combined with XGBoost Model," Sustainability, vol. 14, no. 16, p. 10421, Aug. 2022, doi: 10.3390/su141610421.

[40] X.-L. Zhang and M. Xu, "AUC optimization for deep learning-based voice activity detection," EURASIP J Audio Speech Music Process, vol. 2022, no. 1, p. 27, Oct. 2022, doi: 10.1186/s13636-022-00260-9.

[41] R. Yacouby and D. Axman, "Probabilistic Extension of Precision, Recall, and F1 Score for More Thorough Evaluation of Classification Models," in Proceedings of the First Workshop on Evaluation and Comparison of NLP Systems, Stroudsburg, PA, USA: Association for Computational Linguistics, 2020, pp. 79–91. doi: 10.18653/v1/2020.eval4nlp-1.9.

AUTHOR BIOGRAPHY

Rini Hafizah is a student of Computer Science Department at Lambung Mangkurat University since 2021. She is interested in Data Science, with a focus on data analysis, machine learning, and predictive modeling. She always strives to find innovative methods in solving various real problems through data utilization. Outside of academia, she actively participates in various collaborative research projects that aim to hone her practical skills and broaden her horizons in the world of Data Science. By continuing to deepen her expertise, she hopes to make a real contribution to the development of technology, especially in the utilization of data for more effective and efficient solutions. As an individual who is committed to continuous learning and innovation, he is always looking for opportunities to develop his skills and collaborate with professionals in his field. She believes that data has great potential to create positive changes in various sectors of industry and life. Email: erhafiza96@gmail.com.

Triando Hamonangan Saragih is currently a lecturer at the Department of Computer Science, Lambung Mangkurat University. He has a deep involvement in the academic world with a main focus on various aspects of Data Science. His academic journey began with completing his undergraduate studies in Informatics at Brawijaya University, Malang, which he successfully completed in 2016. To deepen her understanding in the field of Computer Science, she then continued her education to the master's level at the same university and successfully earned a master's degree in 2018. Her research interests center on Data Science, where she continues to develop her expertise in data analysis, predictive modeling, and the application of data-driven technologies. As an academic, he is active in scientific research and collaboration to develop innovative solutions that can be applied in various fields. He also seeks to contribute to the development of science through publications and academic discussions. Email: triando.saragih@ulm.ac.id Orcid ID: 0000-0003-4346-3323.

Muliadi is a lecturer in the Department of Computer Science at Lambung Mangkurat University, specializing in Artificial Intelligence, Decision Support Systems, and Data Science. His academic journey commenced with earning a bachelor's degree in Informatics Engineering from STMIK Akakomin in 2004. To further enhance his knowledge, he pursued and successfully obtained a master's degree in Computer Science from Gadjah Mada University in 2009.

Corresponding author: Triando Hamonangan Saragih, triando.saragih@ulm.ac.id, Department of Computer Science, Lambung Mangkurat University, Jalan A. Yani Km 36, Banjarbaru 70714, Kalimantan Selatan, Indonesia.
DOI: https://doi.org/10.35882/ijeeemi.v7i2.81

With a strong foundation in Data Science, he possesses extensive expertise in analyzing and interpreting complex datasets. Additionally, he has valuable skills in Start-up Business Development, Digital Entrepreneurship, and Data Management. His experience allows him to contribute significantly to both academic and practical applications of technology. Beyond his role as a lecturer, he actively engages in research and collaborative projects aimed at advancing technological innovation. Through his expertise, he strives to bridge the gap between academia and industry, fostering solutions that drive digital transformation and business growth. Email: muliadi@ulm.ac.id Orcid ID: 0000-0003-2871-9482.

Fatma Indriani is a lecturer in the Department of Computer Science at Lambung Mangkurat University, with a strong research interest in Data Science. Before pursuing an academic career, she completed her undergraduate studies in the Informatics Department at the Bandung Institute of Technology. In 2008, she began her journey as a lecturer at Lambung Mangkurat University, contributing to the field of Computer Science through teaching and research.To further expand her expertise, she pursued a master's degree at Monash University, Australia, which she successfully completed in 2012. Her academic journey continued with a doctorate in Bioinformatics from Kanazawa University, Japan, which she completed in 2022. With a focus on both Data Science and Bioinformatics, she actively engages in research, exploring innovative ways to leverage data-driven technologies for scientific advancement. Her dedication to academia and research allows her to contribute significantly to the development of knowledge in her field, while also mentoring students and collaborating on interdisciplinary projects. Email: f.indriani@ulm.ac.id Orcid ID: 0009-0006-7180-6708.

Muhammad Itqan Mazdadi He is a lecturer in the Department of Computer Science at Lambung Mangkurat University, with research interests primarily in Data Science and Computer Networking. His academic journey began at Lambung Mangkurat University, where he earned his undergraduate degree in Computer Science in 2013. To further deepen his expertise, he pursued and successfully completed a master's degree in Informatics from the Islamic University of Indonesia, Yogyakarta. Currently, he holds the position of Secretary in the Computer Science Department at Lambung Mangkurat University, contributing to both academic and administrative responsibilities. His research focuses on exploring advancements in data-driven technologies and network systems. Through his role as a lecturer and researcher, he actively engages in academic development, fostering innovation in his field. His dedication to education and research enables him to support students and collaborate on projects that enhance technological progress. Email: mazdadi@ulm.ac.id Orcid ID: 0000-0002-8710-4616

Corresponding author: Triando Hamonangan Saragih, triando.saragih@ulm.ac.id, Department of Computer Science, Lambung Mangkurat University, Jalan A. Yani Km 36, Banjarbaru 70714, Kalimantan Selatan, Indonesia.
DOI: https://doi.org/10.35882/ijeeemi.v7i2.81