Vol. 6 No. 3, 2025, pp. 264-275 **Jurnal Pepadun**

Classification of Public Sentiment towards the Performance of the Ministry of Communication and Digital regarding Online Gambling

[1]Ika Rahma Alia, *[2]Favorisen Rosyking Lumbanraja, [3]Aristoteles, [4]Rico Andrian

[1,2,3,4] Department of Computer Science, Faculty of Mathematics and Natural Sciences, Universitas Lampung
Jl. Prof. Dr. Ir. Sumantri Brojonegoro, No. 1, Gedong Meneng, Rajabasa, Bandar Lampung, Lampung, Indonesia
e-mail: [1] ikarahmaalia2310@gmail.com, ***[2]favorisen.lumbanraja@fmipa.unila.ac.id**, [3]aristoteles@fmipa.unila.ac.id, [4]rico.andrian@fmipa.unila.ac.id

Abstract - *Online gambling is a social issue currently in the spotlight in Indonesia. Although the government, particularly the Ministry of Communication and Digital (Kemkomdigi), has taken various measures, such as blocking websites and conducting digital literacy campaigns, online gambling remains rampant and has sparked various public reactions. Social media, particularly Instagram, has become a public space where people express their opinions and sentiments regarding government performance. This study aims to classify public sentiment based on comments directed at the official Kemkomdigi Instagram account regarding the issue of online gambling. This study uses two machine learning algorithms, Random Forest and XGBoost, to compare the effectiveness of the models in classifying positive and negative sentiment. A total of 724 comments were collected and manually labeled by three annotators using a voting method. Preprocessing included cleaning, case folding, tokenization, normalization, stopword removal, and stemming. Feature representation was performed using the TF-IDF method. The data was split with a 70:30 ratio and balanced using Random Oversampling. Model training used 10-fold cross-validation and hyperparameter tuning through GridSearchCV. The evaluation results showed that the tuned Random Forest performed the best, with an accuracy of 0.7082. These findings demonstrate that machine learning approaches, particularly Random Forest, are effective in automatically identifying public sentiment toward emerging public policy issues on social media.*

Keywords: *Online Gambling; Public Sentiment; Random Forest; Sentiment Analysis; XGBoost.*

1. INTRODUCTION

In Indonesia, online gambling has become a significant problem. Transaction values and the number of offenders are both steadily rising. The Financial Transaction Reports and Analysis Center (PPATK) estimates that between January and March 2024, the total value of online gambling transactions was close to 90 trillion rupiah [1]. Through the Ministry of Communication and Digital (Kemkomdigi), the government has put in place a number of programs, such as cross-sector collaboration, digital literacy campaigns, and website blocking. However, people's engagement with government institutions has led to controversy and a loss of public trust.

The digital era has provided people new platforms to express their opinions. The public can now openly evaluate and commend government policies on social media. Because it lets users leave direct comments on posts form government organizations, Instagram is one of the most popular platforms. Interestingly, people often use it not only to share opinions but also to engage in discussions. Around 86.6% of Indonesians actively use Instagram, according to information from Kemkomdigi [2]. Real-time public sentiments are reflected in these comments, and can also provide useful information for assessing policies.

A method that can be utilized to analyze public opinion through social media is sentiment analysis. The process of determining a text's polarity whether positive, negative, or neutral and extracting associated features is called sentiment analysis, sometimes referred to as opinion mining [3]. The aim is to comprehend and recognize the views, evaluation, or beliefs expressed in the text. This method is commonly employed in politics, marketing, and social sciences as it offers a quantitative summary of qualitative views [4].

© 2025 The Authors | doi: 10.23960/pepadun.v6i3.295

Grasping public views on the Ministry of Communication and Digital's effort to tackle online gambling is crucial. The objective is to categorize public opinion into positive or negative sentiments, thus providing the government with insights into assessing the effectiveness of policies. Advances in artificial intelligence and machine learning, such as the Random Forest and XGBoost algorithms, support this process with their ability to process large and complex data sets.

In a previous study [5], GoEmotion data classification was performed using a total of three model: Random Forest, XGBoost, and LightGBM. The dataset under consideration comprised a total of 7.325 data points, which were divided into five distinct categories: angry, afraid, happy, love, and sad. The dataset was partitioned into a training set and a testing set, with a proportion of 80:20, and after hyperparameter tuning, the Random Forest model achieved the highest accuracy of 85%.

Another study analyzed the sentiment of YouTube comments related to Islamophobia using the Random Forest algorithm [6]. From 1,000 comments (631 positive, 369 negative), preprocessing was performed in the form of cleaning, case folding, tokenizing, stopword removal, and stemming, as well as TF-IDF calculation. The model was tested using combination of parameters and achieved the highest accuracy of 79% on a 90:10 data split with the best parameters: n_estimators = 10, max_depth = 25, and min_samples_split = 10. Evaluation metrics include Precision 79%, Recall 95%, and F1-Score 86%.

This study was conducted to classify public sentiment towards the performance of the Ministry of Communication and Digital in addressing online gambling based on comments Instagram. This study uses a machine learning approach by comparing the performance of two classification algorithms, namely Random Forest and XGBoost. By implementing text preprocessing techniques, *TF-IDF* weighting, and model performance evaluation with cross validation and hyperparameter tuning, the results of this study are expected to contribute to the development of public sentiment monitoring systems.

2. RESEARCH METHODOLOGY

2.1 Sentiment Analysis

Sentiment analysis is a process in text mining and Natural Language Processing (NLP) used to identify and classify opinions or emotions in text, be it positive, negative, or neutral [7]. Sentiment analysis is used to evaluate data such as social media comments, reviews, or articles to understand the author's opinion on a topic. Indirectly, sentiment analysis can be performed using a technique called Text Mining.

2.2 Text Mining

Text mining is defined as the process obtaining information and identifying latent patterns from unstructured text, such as comments on social media or digital document [8]. The initial step is typically preprocessing, which involves the application of Natural Language Processing (NLP) methods, including syntax analysis and tokenization. The fields of NLP and text mining have advanced swiftly in recent years, propelled by the growing digitization of data, enhanced access to extensive databases, and the swift proliferation of social media usage [9]. Text classification, which includes sentiment analysis, is one important application of text mining. The goal of this program is to categorize text into one of different opinion categories positive, negative, neutral, or from a predefined class.

2.3 Random Forest

Random Forest is an ensemble learning algorithm that randomly builds a lot of decision trees using the CART method and then combines the result to make predictions more accurate [10]. The features are split up gradually to make each tree, with the goal of reducing error. The Random Forest algorithm makes a final choice by adding up the votes from all trees in the model. The class with the most votes is chosen as the final prediction in classification tasks [11]. Although effective in many cases, Random Forest has limitations in recognizing complex interactions between features, especially if the interactions are not accompanied by the influence of each feature individually.

2.4 Extreme Gradient Boosting (XGBoost)

XGBoost (Extreme Gradient Boosting) is an ensemble based machine learning algorithm developed to improve prediction accuracy through boosting techniques [12]. The algorithm works incrementally, where each new model is built to correct the error of the previous model by iteratively minimizing the loss function [13]. With regularization support and parallel computing, XGBoost is able to produce accurate models that are resistant to overfitting [14].

2.5 k-Fold Cross Validation

k-fold cross validation is a validation method that involves the division of the training data into k parts. In this process, each part is used as validation data, while the rest is used for traing [15]. This process is repeated k times, and the average error from all iterations is used to obtain a more accurate and reliable performance evaluation [16] [17]. A common way to test how well a classification works is with k-fold cross validation. This method is particularly well suited for scenarios where the data set is not overly large.

2.6 Grid SearchCV

GridSearchCV is a hyperparameter tuning method that automates the process of identifying the optimal combination of parameters from a predefined set [18]. so the search process does not need to be done manually [19]. This process involves trying each parameter combination in the model and evaluating its performance using cross-validation. Grid search is often combined with the k-fold cross validation technique, which generates an evaluation metric for classification models [20]. The combination that provides the best performance is selected as the optimal parameter for the model.

2.7 Research Framework

The research involved several steps to analyze sentiment related to the performance of the Ministry of Communication and Digital. Each step in the process plays a crucial role in ensuring the accuracy of the analysis results. Figure 1 illustrates the research workflow.

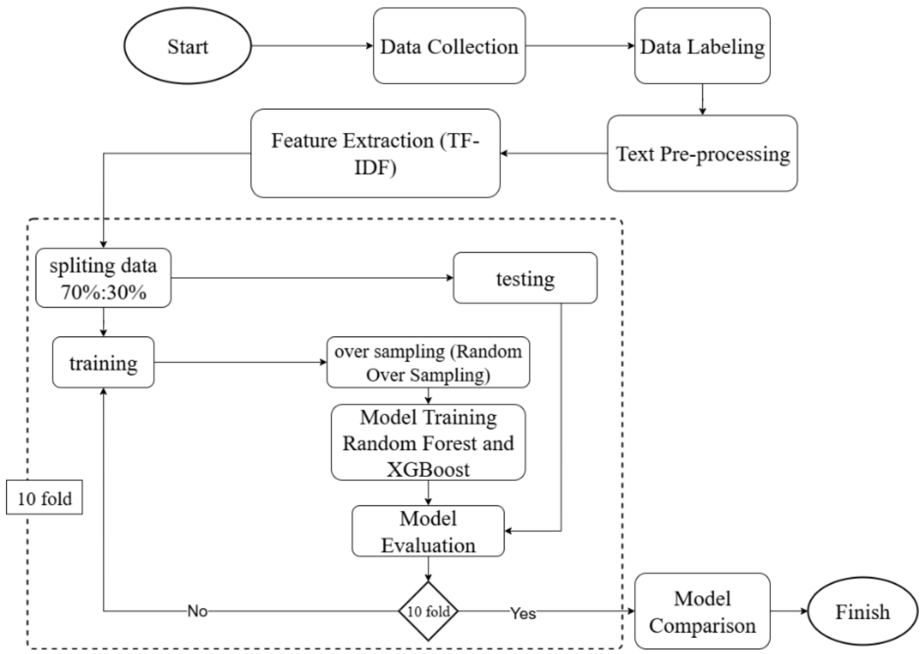

Figure 1. The research workflow.

3.1 Data Collection

This study used Instagram comment data as the object of analysis. The comments were obtained from the official Instagram account of the Ministry of Communication and Digital (Kemkominfo). The comments under review were obtained from posts related to online gambling. To obtain data efficiently, the InsC Instagram Comments Picker & Exporter tool was used, a tool that can automatically extract comments from specific Instagram posts using the post ID. The data collection process focused on posts containing content related to the eradication of online gambling. Data was collected from 12 posts published between October 21 and December 2024. From these uploads, 724 comments were obtained, which became the primary data.

3.2 Data Labeling

Data labeling was done manually by three annotators. Three annotators labeled the sentiments of the collected comments, classifying them into two categories (positive and negative). The labeling process was done independently to avoid inter label influence. The final label for each comment was determined through a majority voting system, with at least two annotators agreeing on the label, as shown in Table 1.

Table 1. Labeling result.

Annotator	Major	University	Total Labeling	
			Positive	Negative
Annotator 1	Undergraduate Computer Science	Universitas Lampung	567	157
Annotator 2	Undergraduate Computer Science	Universitas Lampung	360	364
Annotator 3	Undergraduate Computer Science	Universitas Lampung	391	333
Result Voting			434	290

3.3 Text Pre-processing

Text preprocessing is a set of methods and techniques used to process and manipulate text data. The main objective is to extract important information from the text and convert it into a form that can be understood and processed by a computer [21]. In this stage, there are 5 processes, as follows:

3.3.1 Cleaning

At this stage, relevant attributes are selected and unnecessary elements such as numbers, punctuation marks, emojis, double spaces, and blank lines are removed. An example of cleaning can be seen in Table 2.

Table 2. Cleaning.

Initial Text	Akhir-akhir ini banyak remaja terjebak dalam judi online! 😵 Mereka mulai bermain dari iseng, tapi akhirnya kecanduan. 50% dari mereka bahkan menghabiskan uang kuliah hanya untuk top up akun game-nya!!!
Cleaning Result	Akhir akhir ini banyak remaja terjebak dalam judi online Mereka mulai bermain dari iseng tapi akhirnya kecanduan dari mereka bahkan menghabiskan uang kuliah hanya untuk top up akun gamenya

3.3.2 Case Folding

This step was taken to convert all text into lowercase letters to keep the writing format consistent. The goal is to standardize word forms so that differences in capital and lowercase letters do not affect the meaning or interpretation of the data. Examples of case folding results can be seen in Table 3.

Table 3. Case folding.

Initial Text	Akhir akhir ini banyak remaja terjebak dalam judi online Mereka mulai bermain dari iseng tapi akhirnya kecanduan dari mereka bahkan menghabiskan uang kuliah hanya untuk top up akun gamenya

Case Folding Result	akhir akhir ini banyak remaja terjebak dalam judi online mereka mulai bermain dari iseng tapi akhirnya kecanduan dari mereka bahkan menghabiskan uang kuliah hanya untuk top up akun gamenya

3.3.3 Tokenization

Tokenization is the process of breaking down text into smaller units called tokens. Tokens can be words, phrases, or specific symbols. The purpose of this process is to enable a more detailed analysis of each element in the sentence at a later stage of processing. Examples of tokenization results can be seen in Table 4.

Table 4. Tokenization.

Initial Text	akhir akhir ini banyak remaja terjebak dalam judi online mereka mulai bermain dari iseng tapi akhirnya kecanduan dari mereka bahkan menghabiskan uang kuliah hanya untuk top up akun gamenya
Tokenization Result	'akhir', 'akhir', 'ini', 'banyak', 'remaja', 'terjebak', 'dalam', 'judi', 'online', 'mereka', 'mulai', 'bermain', 'dari', 'iseng', 'tapi', 'akhirnya', 'kecanduan', 'dari', 'mereka', 'bahkan', 'menghabiskan', 'uang', 'kuliah', 'hanya', 'untuk', 'top', 'up', 'akun', 'gamenya'

3.3.4 Normalization

Normalization stage aims to replace words that are not standard, such as abbreviations or slang, into a standard form according to Indonesian language rules. Examples of normalization results provided in Table 5.

Table 5. Normalization.

Initial Text	'akhir', 'akhir', 'ini', 'banyak', 'remaja', 'terjebak', 'dalam', 'judi', 'online', 'mereka', 'mulai', 'bermain', 'dari', 'iseng', 'tapi', 'akhirnya', 'kecanduan', 'dari', 'mereka', 'bahkan', 'menghabiskan', 'uang', 'kuliah', 'hanya', 'untuk', 'top', 'up', 'akun', 'gamenya'
Normalization Result	'akhir', 'akhir', 'ini', 'banyak', 'remaja', 'terjebak', 'dalam', 'judi', 'online', 'mereka', 'mulai', 'bermain', 'dari', 'iseng', 'tetapi', 'akhirnya', 'kecanduan', 'dari', 'mereka', 'bahkan', 'menghabiskan', 'uang', 'kuliah', 'hanya', 'untuk', 'isi', 'ulang', 'akun', 'game'

3.3.5 Stopword Removal

Stopword removal is the process of removing common words that appear frequently in text but do not provide important information for analysis. Although frequently used in sentences, these words are considered to have no significant analytical value so they are removed to simplify the data and focus processing on more relevant words. Examples of stopword removal results can be seen in Table 6.

Table 6. Stopword removal.

Initial Text	'akhir', 'akhir', 'ini', 'banyak', 'remaja', 'terjebak', 'dalam', 'judi', 'online', 'mereka', 'mulai', 'bermain', 'dari', 'iseng', 'tetapi', 'akhirnya', 'kecanduan', 'dari', 'mereka', 'bahkan', 'menghabiskan', 'uang', 'kuliah', 'hanya', 'untuk', 'isi', 'ulang', 'akun', 'game'
Stoword Removal Result	'akhir', 'akhir', 'banyak', 'remaja', 'terjebak', 'judi', 'online', 'mulai', 'bermain', 'iseng', 'akhirnya', 'kecanduan', 'bahkan', 'menghabiskan', 'uang', 'kuliah', 'isi', 'ulang', 'akun', 'game'

3.3.6 Stemming

Stemming is defined as the process of converting affixed words into basic forms or basic words. This process involves removing affixes such as prefixes and suffixes so that different variations of words derived from the same root can be considered as one unit. The purpose of stemming is to simplify word structure, making text analysis more efficient and consistent. Examples of stemming results can be found in Table 7.

Table 7. Stemming.

Initial Text	'akhir', 'akhir', 'banyak', 'remaja', 'terjebak', 'judi', 'online', 'mulai', 'bermain', 'iseng', 'akhirnya', 'kecanduan', 'bahkan', 'menghabiskan', 'uang', 'kuliah', 'isi', 'ulang', 'akun', 'game'
Stemming Result	'akhir', 'akhir', 'banyak', 'remaja', 'jebak', 'judi', 'online', 'mulai', 'main', 'iseng', 'akhir', 'candu', 'bahkan', 'habis', 'uang', 'kuliah', 'isi', 'ulang', 'akun', 'game'

3.4 Feature Extraction (TF-IDF)

TF-IDF stands for "Term frequency-inverse document frequency", a method for determining the importance of a word in a document by considering its frequency in other documents [22]. This technique assigns weight to words in text processing by combining Term Frequency (TF) and Inverse Document Frequency (IDF). TF is indicative of the frequency with which a word appears in a given document, while IDF signifies the word's rarity across the entire corpus. The TF-IDF equation can be found in Equations (1), (2), and (3).

$$\text{TF}\ (t,d) = \frac{f(t,d)}{N_d} \tag{1}$$

$$\text{IDF}(t) = \log\left(\frac{N}{df(t)}\right) \tag{2}$$

$$\text{TF-IDF} = \text{TF}(t,d)\ \times \text{IDF}\ (t) \tag{3}$$

Where TF(t,d) is the frequency of occurrence of term t in document d, N is the number of document sets, $df(t)$ is the number of documents containing term t.

3.5 Data Sharing and Balancing

After the preprocessing and labeling stages, the data is divided into two parts: 70% training data and 30% test data. This division is the basis for the next model validation process. The resulting training data is then validated using the k-fold cross validation method with a value of k = 10, where the data is divided into ten subsets, which are alternately used as training and validation data in each round.

To address the imbalance in data size between classes, Random Over Sampling (ROS) is used during the training process in each fold. This technique was only applied to the training data, while the validation data remained unsampled to maintain evaluation accuracy and avoid bias due to data duplication [23]. In ROS, added instances generally contain more irrelevant labels than relevant labels. Since the label imbalance is dominated by the negative class as the majority, this oversampling process tends to only slightly improve the balance [24].

3.6 Model Training

The model was trained using two classification algorithms Random Forest and XGBoost. In order to comprehensively evaluate the performance of the model, a 10-fold cross validation was used, in which the training data was divided into ten parts and the training process was performed in turn. To achieve optimal model performance, hyperparameter tuning was performed using GridSearchCV. This process finds the best combination of parameters based on the cross-validation results.

3.7 Model Evaluation

The confusion matrix is a performance evaluation method for machine learning classification problems, whether for two or more classes. Evaluation of classification models can be done using metrics such as accuracy, precision, recall, and F1 score [25]. The matrix is a table consisting of four main components that describe the prediction results of the classification model [26]. The structure of the Confusion Matrix is shown in Table 8, with rows representing actual classes and columns representing predicted classes.

© 2025 The Authors | doi: 10.23960/pepadun.v6i3.295

Table 8. Confusion matrix.

		Prediction Class	
		Positive	Negative
Actual Class	Positive	TP	FN
	Negative	FP	TN

* Description: True Positive (TP) = Positive data that is correctly classified; False Positive (FP) = Negative data that is classified as positive; True Negative (TN) = Negative data that is correctly classified; False Negative (FN) = Positive data that is classified as negative.

Accuracy is an evaluation metric that measures how well the model makes correct predictions from the total number of predictions made. The accuracy equation can be seen in Equation (4).

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

Precision is a metric for evaluating the model that calculates the proportion of accurate predictions made for positive categories, relative to the total number of positive predictions made. The equation for calculating the precision value can be found in Equation (5).

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

Recall is an evaluation metric that describes how well the model identifies positive classes correctly. The equation for calculating the recall value can be seen in Equation (6).

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

F1-Score is an evaluation metric that reflects the balance between precision and recall. The equation for calculating the F1-Score value can be seen in Equation (7).

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

3. RESULTS AND DISCUSSION

4.1 GridSearchCV

After preprocessing and choosing an algorithm, GridSearchCV was used to tune the hyperparameters and find the best setup for each model. This method systematically searches for predefined parameter combinations and uses 10-fold cross-validation to check their accuracy and stability [18]. The hyperparameters of the Random Forest and XGBoost models were tuned using training data. The mean F1 score across all cross-validation folds was utilized to select the optimal configuration. The optimal hyperparameter configuration for Random Forest is presented in Table 9, while the configuration for XGBoost is shown in Table 10.

Table 9. Random forest hyperparameters.

Hyperparameters	Value
max_depth	None, 10, 20
max_features	Sqrt, log2
n_estimators	100, 300, 500
min_samples_split	2, 5, 10

 | doi: 10.23960/pepadun.v6i3.295

Table 10. XGBoost hyperparameters.

Hyperparameters	**Value**
learning_rate	0.1, 0.3, 0.5, 0.7, 0.9
max_depth	1, 3, 5, 7, 9
subsample	0.1, 0.3, 0.5, 0.8, 1
colsample_bytree	0.1, 0.3, 0.5, 0.8, 1

The best search results using GridSearchCV show that the Random Forest model obtained the optimal configuration with the following parameter combinations: max_depth = None, max_features = 'log2', min_samples_split = 5, and n_estimators = 100. Meanwhile, for the XGBoost model, the best combination found is: colsample_bytree = 0.5, learning_rate = 0.9, max_depth = 1, and subsample = 0.8.

4.2 Model Training

Model training was performed using two approaches: without hyperparameter tuning (using default parameters) and with tuning using GridSearchCV. The aim was to compare the initial performance with the post-optimization performance. The evaluation was performed using 10-fold cross validation, and the metrics used include accuracy, precision, recall, and F1 score. The training results can be seen in Figure 2 and 3.

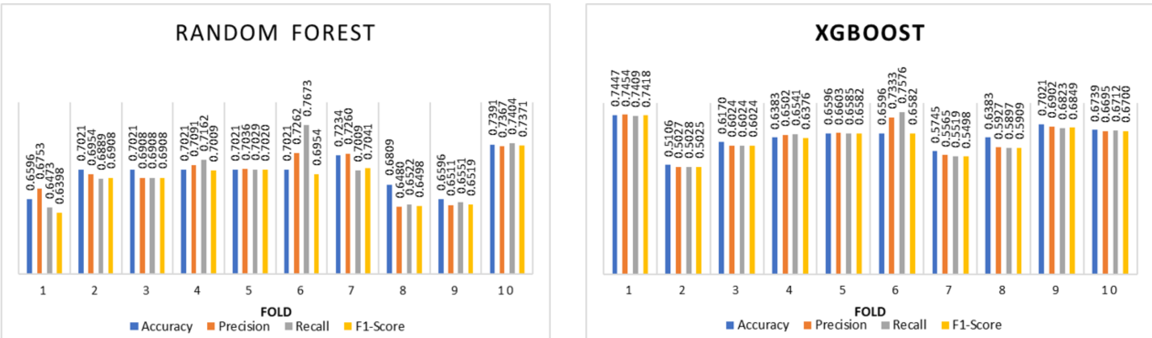

Figure 2. Training result without hyperparameter tuning.

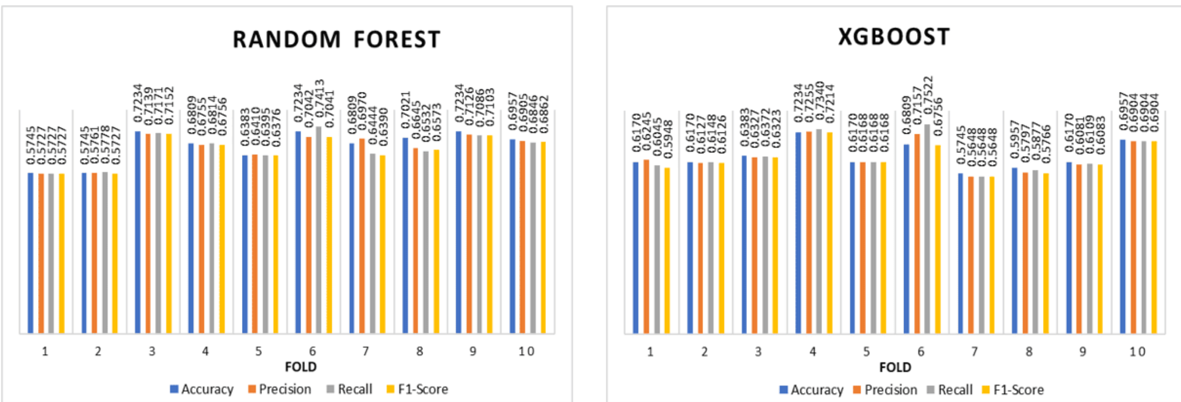

Figure 3. Training result with hyperparameter tuning.

Figures 2 and 3 show the evaluation results of Random Forest and XGBoost on each fold using Accuracy, Precision, Recall, and F1-Score metrics. Figure 2 shows the results without tuning, while Figure 3 illustrates the results after tuning with GridSearchCV. Overall, Random Forest performed more stably and outperformed XGBoost. Tuning does not always improve performance significantly; at some folds, it even degrades, especially for the XGBoost model. The average training results for each model can be seen in Table 11.

 | doi: 10.23960/pepadun.v6i3.295

Table 11. Training result.

Training	Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Without Tuning	Random Forest	69.73	69.62	69.62	68.63
	XGBoost	64.19	64.03	64.11	62.96
With Tuning	Random Forest	67.17	66.48	66.21	65.71
	XGBoost	63.77	63.71	64.13	62.94

In Table 11, without tuning, Random Forest shows the best performance with 69.73% accuracy and 68.63% F1 score, outperforming all other models in every metric. In contrast, tuning had no significant impact on performance, even causing a decrease in results for XGBoost. This shows that the default Random Forest settings are optimal enough to accomplish the classification task on the data used.

In the training phase, the performance values of Random Forest and XGBoost with default parameters were higher than after parameter tuning. This decrease occurred because the model became more stringent in fitting the training data, resulting in lower training results. However, this condition actually shows that the model is not overly dependent on the training data, as evidenced by the improvement in evaluation results after parameter tuning was performed.

4.3 Model Evaluation

Model evaluation was conducted to measure the performance of both models. Model evaluation was conducted using two algorithms: Random Forest and XGBoost. The performance of each model was measured using classification metrics, including Accuracy, Precision, Recall, and F1-Score [25]. The following are the evaluation results for both models.

Table 12. Model evaluation results.

Description	Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Without Tuning	Random Forest	68.16	68.69	68.16	68.32
	XGBoost	65.67	65.86	65.67	65.75
With Tuning	Random Forest	71.64	71.52	71.64	70.90
	XGBoost	67.66	69.81	67.66	67.85

Table 12 shows the results of evaluating both models before and after hyperparameter tuning. In the default configuration, Random Forest performed better than XGBoost, with an Accuracy of 68.16%, Precision of 68.69%, Recall of 68.16%, and F1-Score of 68.32%. These values reflect fairly accurate and balanced predictive capabilities. In contrast, XGBoost only achieved an Accuracy of 65.67%, Precision of 65.86%, Recall of 65.67%, and F1-Score of 65.75%, so its performance was still below that of Random Forest.

After hyperparameter tuning, the performance of both models improved, but Random Forest remained superior with an Accuracy of 71.64%, Precision of 71.52%, Recall of 71.64%, and F1-Score of 70.90%. Meanwhile, XGBoost achieved an Accuracy of 67.66%, Precision of 69.81%, Recall of 67.66%, and an F1-Score of 67.85%. These results confirm that while tuning can improve performance, Random Forest consistently outperforms XGBoost across all evaluation metrics.

4.4 Model Comparison

The purpose of this test is to evaluate the extent to which the trained model can effectively classify data. The outcomes demonstrate the model's capacity to identify novel patterns and exhibit its degree of generalization. The outcomes of the model comparison are presented in Table 13.

Vol. 6 No. 3, 2025, pp. 264-275
Jurnal Pepadun
 | doi: 10.23960/pepadun.v6i3.295

Table 13. Prediction results.

Model	Without Hyperparameter Tuning		With Hyperparameter Tuning	
	True	**False**	**True**	**False**
Random Forest	137	64	144	57
XGBoost	132	69	136	65

In Tables 12 and 13, Random Forest performs better with 68.16% Accuracy and 68.32% F1-Score, successfully classifying 137 data points correctly and 64 incorrectly. In contrast, XGBoost performed lower with an accuracy of 65.67% and F1 score of 65.75%, resulting in 132 correct and 69 incorrect classifications. After tuning, both models showed improved performance. Random Forest achieved 144 correct classifications and 57 incorrect ones, with an accuracy of 71.64% and an F1 score of 70.90%. Meanwhile, XGBoost successfully classified 136 data points correctly and 65 incorrectly, with an accuracy of 67.66% and an F1 score of 67.85%.

4.5 WordCloud

For ease of interpretation, the analysis results were visualized in a word cloud format, displaying the dominant words based on their size and frequency of occurrence. This visualization helps illustrate the tendency of each model to recognize certain words as sentiment indicators. The following is a representation of the word sets from the Random Forest and XGBoost models based on positive and negative classes.

Figure 4. WordCloud Random Forest.

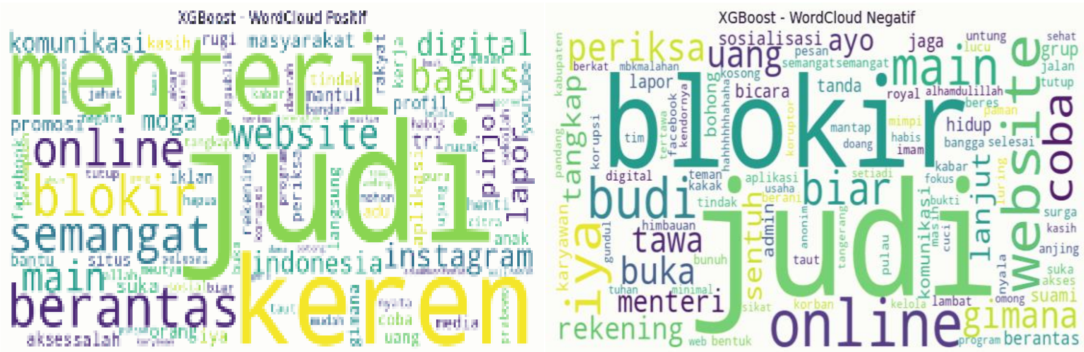

Figure 5. WordCloud XGBoost.

In Figures 4 and 5, there are words that appear in both classes. The presence of the same words in both classes suggests that these words are lexically neutral, and their meaning is dependent on the context of the sentence. For instance, the term "gambling" can be used in a positive context, such as to support efforts to eradicate the problem, or in a negative context, such as to discuss its adverse effects. This finding indicates that the model takes into account the entire sentence structure, rather than solely the meaning of individual words, when assessing sentiment.

4. CONCLUSIONS

The results of the study indicate that the Random Forest and XGBoost algorithms can be effectively applied to classify public sentiment regarding the Ministry of Communication and Digital's performance in addressing online gambling issues. The classification process is comprised of multiple stages, including manual data labeling, text preprocessing, data balancing, model training, and evaluation using accuracy, precision, recall, and F1-score metrics. Random Forest reached an F1-score of 70% with 144 data points accurately classified, whereas XGBoost attained an F1-score of 68% with 136 data points accurately classified. This finding indicates that Random Forest demonstrates superior efficacy in sentiment classification in this study following parameter optimization. However, it should be noted that both algorithms retain their respective advantages. Future research should use a larger and more diverse data set to improve the generalization of the model. In addition, exploration of other methods such as deep learning or the use of word insertion such as Word2Vec and BERT can be considered to achieve more optimal classification results. The addition of neutral sentiment classes and multi category analysis can also enrich future studies.

LITERATURE

[1] Kemkomdigi, "Transaksi Judi Online Kuartal Pertama 2025 Turun Hingga 80 persen," *KEMKOMDIGI*, 2025. https://www.komdigi.go.id/berita/siaran-pers/detail/transaksi-judi-online-kuartal-pertama-2025-turun-hingga-80-persen (accessed Jul. 23, 2025).

[2] N. Julius, "Data Jumlah Pengguna Instagram di Indonesia 2025Bagaimana data demografis pengguna Instagram di tingkat global?Data Jumlah Pengguna Instagram di Indonesia 2025," *Upgraded.id*, 2025. https://upgraded.id/data-jumlah-pengguna-instagram-di-indonesia (accessed Jul. 23, 2025).

[3] E. Salim and M. Syafrullah, "Jakarta Barat Menggunakan Algoritme K-Nearest Neighbor," vol. 20, no. 1, pp. 58–65, 2023, [Online]. Available: https://kemsalim.space/ulasan_dukcapil/

[4] R. Obiedat, D. Al-Darras, E. Alzaghoul, and O. Harfoushi, "Arabic Aspect-Based Sentiment Analysis: A Systematic Literature Review," *IEEE Access*, vol. 9, pp. 152628–152645, 2021, doi: 10.1109/ACCESS.2021.3127140.

[5] N. Anggraini, S. J. Putra, L. K. Wardhani, F. D. U. Arif, N. Hakiem, and I. M. Shofi, "A Comparative Analysis of Random Forest, XGBoost, and LightGBM Algorithms for Emotion Classification in Reddit Comments," *J. Tek. Inform.*, vol. 17, no. 1, pp. 88–97, 2024, doi: 10.15408/jti.v17i1.38651.

[6] I. Afdhal, R. Kurniawan, I. Iskandar, R. Salambue, E. Budianita, and F. Syafria, "Penerapan Algoritma Random Forest Untuk Analisis Sentimen Komentar Di YouTube Tentang Islamofobia," *J. Nas. Komputasi dan Teknol. Inf.*, vol. 5, no. 1, pp. 122–130, 2022, [Online]. Available: http://ojs.serambimekkah.ac.id/jnkti/article/view/4004/pdf

[7] I. S. K. Idris, Y. A. Mustofa, and I. A. Salihi, "Analisis Sentimen Terhadap Penggunaan Aplikasi Shopee Mengunakan Algoritma Support Vector Machine (SVM)," *Jambura J. Electr. Electron. Eng.*, vol. 5, no. 1, pp. 32–35, 2023, doi: 10.37905/jjeee.v5i1.16830.

[8] K. C. Astuti, A. Firmansyah, and A. Riyadi, "Implementasi Text Mining Untuk Analisis Sentimen Masyarakat Terhadap Ulasan Aplikasi Digital Korlantas Polri pada Google Play Store," *REMIK Ris. dan E-Jurnal Manaj. Inform. Komput.*, vol. 8, no. 1, pp. 383–394, 2024.

[9] X. Zhang, R. Gao, Z. Xiao, K. Wang, T. Liu, M. Liang, and J. Zhan., "Natural language processing and text mining in transportation: Current status, challenges, and future roadmap," *Expert Syst. Appl.*, vol. 296, no. November 2024, 2026, doi: 10.1016/j.eswa.2025.129050.

[10] R. Blum, M. Hiabu, E. Mammen, and J. T. Meyer, "Pure interaction effects unseen by Random Forests," *Comput. Stat. Data Anal.*, vol. 212, no. July, p. 108237, 2025, doi: 10.1016/j.csda.2025.108237.

[11] X. Wei, Y. Xu, X. Li, G. Fan, X. Cheng, T. Yu, and B. Jiang, "Study on prediction model of nitrogen

oxide concentration in reprocessing plant based on random forest," *Int. J. Adv. Nucl. React. Des. Technol.*, vol. 7, no. 2, pp. 63–69, 2025, doi: 10.1016/j.jandt.2025.04.011.

[12] Y. Jiang, G. Tong, H. Yin, and N. Xiong, "A Pedestrian Detection Method Based on Genetic Algorithm for Optimize XGBoost Training Parameters," *IEEE Access*, vol. 7, pp. 118310–118321, 2019, doi: 10.1109/ACCESS.2019.2936454.

[13] J. Zhang and Z. Zhao, "Corporate ESG rating prediction based on XGBoost-SHAP interpretable machine learning model," *Expert Syst. Appl.*, vol. 295, no. July 2025, p. 128809, 2026, doi: 10.1016/j.eswa.2025.128809.

[14] Z. Mustaffa and M. H. Sulaiman, "Advanced forecasting of building energy loads with XGBoost and metaheuristic algorithms integration," *Energy Storage Sav.*, pp. 0–55, 2025, doi: 10.1016/j.enss.2025.03.005.

[15] S. Yadav and S. Shukla, "Analysis of k-Fold Cross-Validation over Hold-Out Validation on Colossal Datasets for Quality Classification," *Proc. - 6th Int. Adv. Comput. Conf. IACC 2016*, no. Cv, pp. 78–83, 2016, doi: 10.1109/IACC.2016.25.

[16] A. Shebl, D. Abriha, M. Dawoud, M. Ali Hussein Ali, and Á. Csámer, "PRISMA vs. Landsat 9 in lithological mapping − a K-fold Cross-Validation implementation with Random Forest," *Egypt. J. Remote Sens. Sp. Sci.*, vol. 27, no. 3, pp. 577–596, 2024, doi: 10.1016/j.ejrs.2024.07.003.

[17] M. T. R. Mahesh, V. V. Kumar, V. D. Kumar, O. Geman, M. Margala, and M. Guduri, "The stratified K-folds cross-validation and class-balancing methods with high-performance ensemble classifiers for breast cancer classification," *Healthc. Anal.*, vol. 4, no. August, pp. 1–10, 2023, doi: 10.1016/j.health.2023.100247.

[18] G. S. K. Ranjan, A. Kumar Verma, and S. Radhika, "K-Nearest Neighbors and Grid Search CV Based Real Time Fault Monitoring System for Industries," *2019 IEEE 5th Int. Conf. Converg. Technol. I2CT 2019*, pp. 9–13, 2019, doi: 10.1109/I2CT45611.2019.9033691.

[19] S. M. Malakouti, M. B. Menhaj, and A. A. Suratgar, "The usage of 10-fold cross-validation and grid search to enhance ML methods performance in solar farm power generation prediction," *Clean. Eng. Technol.*, vol. 15, no. July, 2023, doi: 10.1016/j.clet.2023.100664.

[20] I. S. Mangkunegara and P. Purwono, "Analysis of DNA Sequence Classification Using SVM Model with Hyperparameter Tuning Grid Search CV," *Proc. - 2022 IEEE Int. Conf. Cybern. Comput. Intell. Cybern. 2022*, pp. 427–432, 2022, doi: 10.1109/CyberneticsCom55287.2022.9865624.

[21] Imamah and F. H. Rachman, "Twitter sentiment analysis of Covid-19 using term weighting TF-IDF and logistic regresion," *Proceeding - 6th Inf. Technol. Int. Semin. ITIS 2020*, pp. 238–242, 2020, doi: 10.1109/ITIS50118.2020.9320958.

[22] B. A. Kuncoro and B. H. Iswanto, "TF-IDF method in ranking keywords of Instagram users' image captions," *2015 Int. Conf. Inf. Technol. Syst. Innov. ICITSI 2015 - Proc.*, pp. 1–5, 2016, doi: 10.1109/ICITSI.2015.7437705.

[23] C. Yang, E. A. Fridgeirsson, J. A. Kors, J. M. Reps, and P. R. Rijnbeek, "Impact of random oversampling and random undersampling on the performance of prediction models developed using observational health data," *J. Big Data*, vol. 11, no. 1, 2024, doi: 10.1186/s40537-023-00857-7.

[24] N. G. Pedrajas, "Partial random under/oversampling for multilabel problems," *Knowledge-Based Syst.*, vol. 302, no. 112355, pp. 1–17, 2024, doi: https://doi.org/10.1016/j.knosys.2024.112355.

[25] A. Arifuddin, G. S. Buana, R. A. Vinarti, and A. Djunaidy, "Performance Comparison of Decision Tree and Support Vector Machine Algorithms for Heart Failure Prediction," *Procedia Comput. Sci.*, vol. 234, pp. 628–636, 2024, doi: 10.1016/j.procs.2024.03.048.

[26] A. Luque, A. Carrasco, A. Martín, and A. de las Heras, "The impact of class imbalance in classification performance metrics based on the binary confusion matrix," *Pattern Recognit.*, vol. 91, pp. 216–231, 2019, doi: 10.1016/j.patcog.2019.02.023.