Journal of Robotics and Control (JRC)
Volume 6, Issue 1, 2025
ISSN: 2715-5072, DOI: 10.18196/jrc.v6i1.24385 

# HyVADSVM: Hybrid VADER-SVM and GridSearchCV Optimization for Enhancing Cyberbullying Detection

Siti Ernawati [1*], Frieyadie [2], Eka Rini Yulia [3]
[1, 2, 3] Faculty of Information and Technology, Universitas Nusa Mandiri, Jakarta, Indonesia
Email: [1] siti.ste@nusamandiri.ac.id, [2] frieyadie@nusamandiri.ac.id, [3] eka.erl@nusamandiri.ac.id
*Corresponding Author

***Abstract*—Cyberbullying detection is becoming increasingly crucial in today's digital era, as many individuals suffer from online harassment. The main challenge lies in accurately identifying patterns of harassment in social media texts, which often use informal languages, slang, and sarcasm. Existing methods struggle to capture emotional context owing to the vast amount of data and rapid digital interactions. This study aims to improve the detection accuracy by combining advanced sentiment analysis using VADER and parameter tuning with GridSearchCV. Data were collected from Instagram, Twitter, and YouTube, with TF-IDF employed for feature extraction. Multiple machine-learning classifiers (SVM, K-NN, NB, LR, DT, and RF) were tested to determine the best-performing model. VADER was selected for its reliability in processing social media texts rich in informal contexts, effectively capturing emotional nuances, such as sarcasm and varying sentiment intensities. This makes it well suited for complex language patterns typical of cyberbullying scenarios, enhancing data labeling and analysis accuracy. Using 10-fold cross-validation for reliable testing, performance metrics (accuracy, precision, recall, and F1-Score) were evaluated using a confusion matrix. The findings highlight SVM as the most effective model when optimized with GridSearchCV, achieving accuracy (98.83%), precision (98.78%), recall (98.83%), and F1-Score (98.62%) with kernel =linear, C=1, and gamma=scale. This optimized model, HyVADSVM model has significant potential in cyberbullying detection, contributing to academic research and serving as an effective tool to prevent online harassment. Future work could integrate this model into real-time systems, improve user safety, and support digital policymaking.**

***Keywords*—Cyberbullying Detection; Hyperparameter Tuning; Machine Learning; Sentiment Analysis; Social Media Analysis.***

## I. INTRODUCTION

The Internet has become one of the greatest innovations in human history, changing how we communicate, work, study, and live our daily lives. The Internet has also become a platform for extensive social interactions. We can quickly connect with friends, family, and the community through social media, online forums, and messaging applications, regardless of distance. In the rapidly evolving digital era, social media has become a leading platform for social interaction, information sharing, and self-expression. With the increase in the number of social media users, cyberbullying has become an increasingly severe problem. Cyberbullying is a prevalent and ever-increasing global problem, characterized by abusive behavior that infiltrates a victim's personal life [1][3]. Cyberbullying often involves the use of fake identities and ambiguous language, making it difficult to detect automatically, especially across social media platforms with varying user characteristics [2][5]. Current cyberbullying detection methods struggle with the complexity of the social media language, including slang and sarcasm, highlighting the need for advanced sentiment analysis and machine learning techniques to improve accuracy and adaptability. Cyberbullying is a mistake made by unwise users of information technology, who can hurt or harass others repeatedly without thinking about the impact and can cause psychological trauma. Perpetrators often use fake identities to feel safe from legal consequences without considering the social rules and moral values that should be maintained during the use of social media [6]–[8]. Statistics on the prevalence of cyberbullying have become increasingly alarming. According to a report from the Cyberbullying Research Center, approximately 37% of adolescents in the United States have experienced cyberbullying, and this number continues to rise with the growth of social media usage. In Indonesia, approximately one in four teenagers reported having experienced online harassment [9].

Many studies have revealed that parents worldwide believe that their children are exposed to cyberbullying [10]. Bullying negatively impacts both physical and psychological effects on perpetrators and victims, and is a widespread social problem among adolescents worldwide [11]. Cyberbullying is a negative aspect of social media [12]–[14]. This includes behaviors such as addictive users, trolling, online witch hunts, spreading fake news, and privacy violations. The adverse effects of cyberbullying include anxiety, depression, self-harm, suicide attempts, mental health problems, and emotional and economic difficulties [15].

Identifying cyberbullying on social media is often difficult because of the large volume of data generated daily. Big data derived from social media are essential in research because online activities generate large volumes of data [16]. The vast amount of data generated from online activities requires sophisticated analytical approaches to filter the relevant information and identify meaningful patterns. With the abundance of data, there are new challenges such as how to analyze data effectively, which requires the ability to filter relevant information, identify meaningful patterns, and make accurate predictions.

The rise of cyberbullying has emphasized the need for effective detection methods given its detrimental impact on mental health [10]. Identifying and dealing with cyberbullying effectively requires a deep understanding of the sentiments expressed in online texts. Sentiment analysis plays an essential role in detecting and classifying language-containing bullying. Detecting and addressing cyberbullying cases as quickly as possible can help to protect individuals and reduce the risk of severe consequences.

Machine learning algorithms have the potential to overcome the challenges of detecting cyberbullying on social media [13][12], and play an essential role in processing large and complex data. Machine learning algorithms can be extracted from social media data to support decision making, such as data grouping, sentiment classification, and anomaly detection [8], and machine learning techniques have been used to address problems in various real-life applications [18][19]. A labeling technique can be used to classify sentiments into bullying and non-bullying labels. VADER is also known as a lexicon dictionary-based sentiment analysis method that has proven to be successful in examining natural language texts. Using sentiment analysis, VADER assesses textual content, including reviews and documents, to classify sentiments as favorable, negative, or neutral. To achieve optimal performance, the parameters of the machine-learning model must be carefully regulated.

This study aimed to address the challenges in detecting cyberbullying by developing a hybrid model that integrates VADER and SVM, optimized through GridSearchCV. This study fills a gap in the automatic detection of cyberbullying, which has thus far been less effective in handling complex language variations and user identities. The resulting model was named HyVADSVM (hybrid valence-aware dictionary and sentiment reasoner (VADER) with SVM) and GridsearchCV. In this study, hyperparameters were used to determine the best combination of parameters and improve accuracy. Hyperparameters can be used to optimize parameters to obtain the best performance [20]. This research primarily contributes to the development of the HyVADSVM model, which merges VADER sentiment analysis with an optimized SVM classifier. This model aims to significantly enhance the accuracy and efficiency of cyberbullying detection, outperforming the previous methods. In this study, the cyberbullying classification problem was solved using the classical machine-learning algorithms described above. These algorithms were chosen because they are among the most popular classification algorithms and are fairly common in literature [21]–[23]. This research can make a significant contribution to better recognizing patterns of abusive language and negative expressions, improving the accuracy of cyberbullying detection, and preventing cyberbullying on social media.

## II. Research Method

The proposed HyVADSVM model is a research framework aimed at optimizing cyberbullying detection. It is a hybrid model that integrates VADER and SVM and is further optimized using GridSearchCV to improve accuracy in line with the research objectives, offering numerous benefits and advantages.

First, the labeling process uses the VADER algorithm to calculate the polarity score, which is cheaper, error-prone, and faster than manual labeling. Second, because supervised machine learning is better than unsupervised learning [24], researchers used SVM, RF, NB, SVM, K-NN, and D.T. as machine learning algorithms. Third, the model can classify cyberbullying behavior using a hyperparameter technique, namely, GridsearchCV, to search for the optimal parameters in the model. Fourth, we deployed the best model in a system that automatically detects and classifies cyberbullying behavior. The proposed model is shown in Fig. 1.

Table I presents the pseudocode of the HyVADSVM model, Hybrid Valence-Aware Dictionary and Sentiment Reasoner (VADER) with the SVM. The first step is preprocessing the text data, and then transforming the cleaned text into features using TF-IDF. The sentiment scoring process is carried out using HyVAD to determine the sentiment score; if the score is positive, the text is labeled as "bullying," otherwise, it is labeled as "non-bullying." Subsequently, the dataset was split into training and testing data, and hyperparameter tuning was performed using GridSearchCV by employing a combination of kernel, C, and gamma parameters for the SVM model. The model was trained using the training data, and the evaluation results were assessed using metrics such as accuracy, precision, recall, and F1 score. Finally, the algorithm returns the sentiment label, the total sentiment score, and the model evaluation metrics.

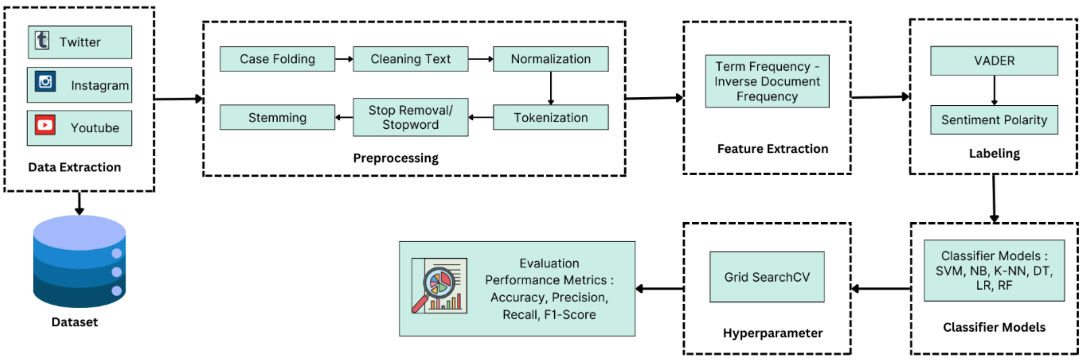



Fig. 1. Proposed HyVADSVM model

TABLE I. PSEUDOCODE OF HYVADSVM MODEL

| No. | **HyVADSVM Algorithm** |
|---|---|
| 1 | **# Input** |
| 2 | INPUT: dataset D |
| 3 | **# Output** |
| 4 | Labeled dataset L |
| 5 | **# Process** |
| 6 | **Preprocessing** |
| 7 | FOR i to D: |
| 8 | **Case Folding**: convert text to lowercase |
| 9 | **Cleaning Text**: remove links, hashtags, symbols, and irrelevant data using RegEx |
| 10 | **Normalization**: normalize text (e.g., handle slang words, typos) |
| 11 | **Tokenization**: split the text into individual tokens (words) |
| 12 | **Stopword Removal**: remove common stopwords from the text |
| 13 | **Stemming**: reduce words to their base or root form |
| 14 | **Feature Extraction** |
| 15 | APPLY TF-IDF to the preprocessed text |
| 16 | **Sentiment Scoring** |
| 17 | calculate the sentiment_score using HyVAD (Hybrid Valence Aware Dictionary) |
| 18 | IF sentiment_score > 0: |
| 19 | SET sentiment_label = "Bullying" |
| 20 | ELSE: |
| 21 | SET sentiment_label = "Non Bullying" |
| 22 | **# Model Training with Hyperparameter Tuning** |
| 23 | SPLIT dataset into training and test sets |
| 24 | DEFINE hyperparameters for tuning: |
| 25 | Kernels, C_values, gamma_values |
| 26 | INITIALIZE GridSearchCV with SVM: |
| 27 | grid = GridSearchCV(SVM(), param_grid={'kernel': kernels, 'C': C_values, 'gamma': gamma_values}, cv=10) |
| 28 | TRAIN SVM model using training data |
| 29 | **# Model Evaluation** |
| 30 | EVALUATE model using metrics (e.g., accuracy, precision, recall, F1 score) |
| 31 | RETURN sentiment_label, total_sentiment_score, model_evaluation_metrics |

## A. Dataset

This study uses sentiment data from 19,377 participants. Data were collected randomly on social media in the form of sentiments about cyberbullying from Twitter, Instagram, and YouTube, with details as follows: 1,554 from Twitter, 4,845 from Instagram, and 12,978 from YouTube, the amount of data can be seen in Fig. 2. The data collection period was from Jan. 2024 to Jun. 2024. Subsequently, the collection of sentiments was integrated into a dataset in the form of CSV. Next, the data were processed using the Python software.

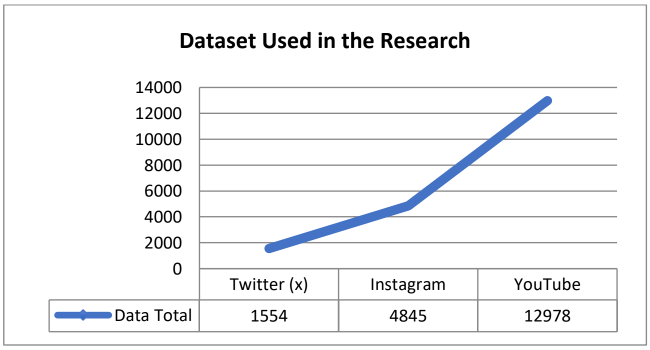



Fig. 2. Dataset used in the research

## B. Data Preprocessing

In text analysis, preprocessing is critical for preparing raw data collected from social media sites (Twitter, Instagram, YouTube) for machine-learning models. Measuring unstructured and noisy text is very complex and involves many methodological challenges, such as processing the data and optimizing the algorithms used to analyze the text [18][25][26]. Each step is illustrated in a diagram. Preprocessing, so case Folding: All texts are in lowercase. In the above example, "Hello" will be transformed to "hello." We performed a similar operation here, so the analysis will not differentiate between the upper and lower cases; thus, we should not count the same word as different.

- Cleaning text: This eliminates things we do not need in our text, such as punctuation, numbers, URL links, emojis, and special characters [22][27]. The aim was to eliminate text coverage that did not contribute to the analysis.
- Normalization: Convert non-standard words or abbreviations into standard, typical forms. For example, u => you and thx =>. Treating different words with equal meanings.
- Tokenization refers to the process of breaking down text into small units [28], which are called "tokens," and most often, it is words or sequences_end A sentence like "I love A.I." breaks down to tokens I, love, and Ai, and can further describe each of these tokens.
- Stopword Removal/Stopword is Removing common words that appear frequently but usually do not have an essential meaning in the analysis, such as "the," "is," "in," "and," etc. Removing stop words helps the model to focus on more meaningful and relevant words.
- Stemming reduces words to their basic form [29]. For example, words like "running," "runner," and "ran" will be changed to the primary form of "run." Stemming helps unify the morphological variations of words so that the model can focus more on the word's primary meaning.

## C. Feature Extraction

Clean the data by preprocessing and convert the raw text into a format that is easier for machine learning algorithms to process and analyze. The results of *preprocessing* in this research used several stages, namely case folding, text cleaning, normalization, tokenization, stop removal, *and stemming*. The feature extraction technique used was TF-IDF. TF-IDF is beneficial for identifying essential words in a document and can help process text [30][31].

TF-IDF is one of the techniques used in text processing to assign weights to words in a document. In practical applications, TF-IDF has been shown to improve the performance of various machine-learning models [32][33]. For instance, in the context of text classification, TF-IDF serves as a robust feature extraction method that enhances the accuracy of classifiers by providing a clear representation of text data [32][35]. In addition, TF-IDF has been adapted for use in specialized domains such as biomedical text analysis, where it aids in identifying relevant terms and improving the retrieval of domain-specific information [36].

TF-IDF aims to identify the most essential words in a document or document collection. *Term Frequency* (T.F.) is the frequency of the occurrence of a word in a document. However, this T.F. value does not provide information regarding the importance of a word in a document. Words that appear frequently in a document, such as conjunctions or common words, may have a high T.F. value but do not have a significant meaning. Therefore, another technique, Inverse Document Frequency (IDF), is required, which gives weight to words that appear infrequently throughout the document.

$$TF\ (t,d) = \frac{\text{Number of times terms } t \text{ appears in document } d}{\text{Total number of terms in document } d} \tag{1}$$

$$IDF\ (t,d) = \log\left(\frac{N}{1+DF(t)}\right) \tag{2}$$

$$TF - IDF(t,d,D) = TF(t,d)\ x\ IDF(t,D) \tag{3}$$

After obtaining the T.F. and IDF values, the TF-IDF value was calculated by multiplying the T.F. value by the *IDF* value. $N$ represents the number of documents, and $DF(t)$ is the number of documents containing the word $t$ represents the number of documents. Words with a high TF-IDF score are considered essential and contribute more to determining the topic of a document or a collection of documents.

## D. Labeling with VADER

The dataset collected does not have labelled output; therefore, it is labelled as bullying or non-bullying to train the supervised classifiers. Therefore, the VADER method was used to label the dataset. VADER determines the negative, neutral, positive, and combined polarity scores for each sentiment [37][38][39]. Considered as negative polarity a combined value less than or equal to -0.05 is considered negative polarity, while a value greater than 0.05 is regarded as positive polarity. Values between 0.05 and −0.05 are considered neutral [16].

The VADER technique was chosen because it is effective at capturing negative sentiments in informal social media texts [Reference]. In this study, the sentiment scoring process was conducted by assigning labels based on sentiment scores. If the positive score exceeded the threshold value (>0), the sentiment was labeled as non-bullying, whereas if it was less than 0, the sentiment was labeled as bullying. Non-bullying sentiment refers to text that does not contain elements of bullying but instead includes content that is neutral or positive, whereas bullying sentiment refers to text or messages that contain negative or aggressive elements intended to demean, harm, or intimidate individuals or groups. VADER was integrated into the model following the pre-processing step.

## E. Classifier Models

Classification in supervised learning involves training a machine-learning model using labelled data, allowing the model to learn the relationship between input features and output labels. After training, the model was used to classify new unlabeled data. Some classification algorithms in machine learning include N.B., K-NN, SVM, L.R., R.F., and D.T. Therefore, the reason for choosing the six algorithms is that they are included in the supervised learning group, which many researchers have used and developed. In this learning, the machine is trained using labeled data, and after training, it can predict the correct results for new data [40]–[42].

### 1) Naïve Bayes (NB)

Naive Bayes is a classification algorithm based on Bayes' theorem. When applied to large datasets, the Proven Bayesian classification has high accuracy and speed. One type of Naive Bayes classification commonly used for text classification, such as sentiment analysis, is Multinomial Naive Bayes (MNB) [43][44]. Multinomial Naive Bayes (MNB) is suitable for classifying text with many features (e.g., words in a document) using probability methods [45]. This algorithm calculates the probability of each word's occurrence in each class and then uses these probabilities to predict the new class of a text. In, emails were classified as spam or non-spam using Multinomial Naive Bayes implementation, document classification into specific topics, or sentiment classification in text. The code used during the experiment involved importing the MultinomialNB class from the naive_Bayes module in the scikit-learn library. Using MultinomialNB, training and prediction of text data can be performed quickly.

The probability of a document in each available category was calculated using the Multinomial Naive Bayes formula. We then predict the category that best matches the document using the probability. Equation (4) is the basic formula for Multinomial Naive Bayes.

$$P\left(\frac{c}{d}\right) = P(c) * \frac{P\left(\frac{d}{c}\right)}{P(d)} \tag{4}$$

$P = \left(\frac{c}{d}\right)$ is the probability that document $d$ belongs to category $c$. $P = (c)$ is the prior probability of category $c$. $P = \left(\frac{d}{c}\right)$ is the probability of the feature (word or term) occurring in document $d$ given that document $d$ belongs to category $c$. $P = (d)$ The probability of document $d$ was evaluated. To calculate the probability $P(d|c)$, the following Equation (5).

$$P\big((d|c)\big) = \prod (P(t|c)^{nt}) \tag{5}$$

$P = \left(\frac{t}{c}\right)$ is the probability of the occurrence of feature $t$ (word or term) in category $c$, and $nt$ is the number of times feature $t$ appears in document $d$. To calculate the probabilities $P(c)$ and $P(d)$, using the following formulas:

$$P(c) = \frac{Nc}{N} \tag{6}$$

Where $Nc$ denotes the number of documents in Category $c$. $N$ is the total number of records. $P(d)$ can be calculated in the same way as $P(d|c)$, assuming that the document could belong to all available categories. After obtaining all the probabilities, the next step is to classify the documents into the category with the highest probability. The category with the highest probability was considered to be the most likely category for the document.

*2) K-NN*

The K-NN is a supervised learning classification algorithm known as a distance-based method. The k-NN algorithm is a robust non-parametric classifier that assigns an unclassified pattern to the class represented by most of its k-nearest neighbors [46], [47]. K-NN uses the same features and classifies the data points based on their proximity to neighbors. In K-NN classification, an unknown pattern is assigned to the most dominant class among its nearest neighbors [48]. In K-NN, the only unknown parameter is K. K is the number of nearest neighbors considered for assigning a label to the current point [49]. Choosing the correct value of K is known as parameter tuning and is crucial for achieving better accuracy. If K is too small, there is a risk of overfitting. However, if K is excessively large, the algorithm becomes computationally expensive. Another commonly used formula for selecting K is $k = \sqrt{n}$, where $n$ is the total number of data points.

*3) SVM*

The Support Vector Machine (SVM) is a fundamental and essential technique in machine learning and is widely used machine learning technique for detecting the polarity of text [50]. The SVM performs well in both classification and regression tasks. To implement the model's performance, researchers will use the sklearn library, specifically SVC, which is the Support Vector Classification implementation from the libsvm library. SVM is an important machine learning algorithm with state-of-the-art performance for many classification problems [22]. The advantage of the SVM is its ability to handle high-dimensional data and provide larger margins for more accurate classification, although it may require longer training times[51].

An SVM is a classification model that determines the best hyperplane for separating classes with the most significant margin [52]–[54]. The SVM first identifies the support vectors for each class. The support vectors are the samples from each class closest to the samples of the other classes. Once the support vectors have been determined, SVM calculates the margin. Think of the margin as the space separating the two classes. It is created based on the support vectors, where these vectors act as the edge boundaries of the margin, often referred to as the "road shoulders." The SVM seeks the most significant margin or comprehensive road, separating the two classes.

Based on Fig. 3, the SVM will choose the margin on the right because the 'road' or margin on the right side is wider than that on the left side. Therefore, the image on the right has a large margin, whereas that on the left has a small margin [55]. The margin is the distance between the hyperplane and nearest data points from each class. A hyperplane is a decision boundary that separates the data from different classes.

$$w.x + b = 0 \tag{7}$$

where $w$ is the weight vector that determines the orientation of the hyperplane; $x$ is the feature vector of the data; and $b$ is the bias that determines the distance of the hyperplane from its origin.

For nonlinear problems, SVM can use kernel functions to map data to higher dimensions. Commonly used kernels include linear, polynomial, Radial Basis Function (RBF), and sigmoid kernels [56], [57]. The selection of an appropriate kernel and tuning parameters often requires experimentation and model validation to determine the optimal configuration.

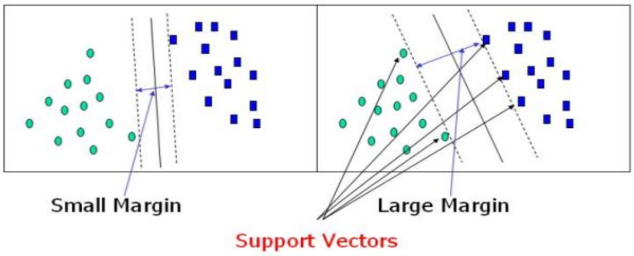



Fig. 3. Margin in SVM

*4) Logistic Regression*

Logistic Regression is a classification algorithm in Machine Learning used to predict the probability of a categorical dependent variable. In logistic regression, the dependent variable is binary and contains data encoded as one or zero [42]. This generalized linear regression technique is used to learn mapping from a set of numerical variables to a binary or probabilistic variable [58].

Logistic Regression can also classify an observation into one of two classes or into one of many classes [59]. Logistic Regression aims to model the probability that a given input instance belongs to a particular class. Logistic Regression is used to estimate this probability by fitting a logistic (sigmoid) function to the input features. The hypothesis function in logistic regression uses a sigmoid function to map the linear output to the range [0, 1].

$$h_0(x) = \frac{1}{1 + e^{-0\,T_x}} \tag{8}$$

Where, $h_o(x)$ represents the predicted probability, $-0\,T_x$ it is the linear combination of the model parameters 0 and the input features $x$, and $e$ is the natural logarithm base.

*5) Random Forest*

Random Forest is an ensemble algorithm that combines multiple decision trees to obtain accurate and stable predictions [60]. Random Forest (RF) is an ensemble method that uses decision trees as individual predictors [61]. Techniques, such as bagging, are based on the random forest method, randomization outputs, and random subspaces without boosting. RF is the most powerful classification algorithms [62]. Each tree was built in the forest using a random subset of training data and a random subset of features. The final prediction of the Random Forest is the average (for regression) or majority vote (for classification) of all trees.

$$y = \frac{1}{N}\sum_{i=1}^{N} y_i \quad (9)$$

Where $N$ is the number of trees in the forest and $y_i$ is the prediction from the $ith$ tree.

*6) Decision Tree*

Decision tree is a powerful method used in various fields such as machine learning, pattern identification, and image processing [63]. A decision tree is a classification model that maps decisions based on data features. Decision trees are widely used both as independent classifiers and as base models in classification methods, primarily because they are easy to understand and explain [64]. Each tree consists of nodes and branches. Each node represents a feature in the category to be classified and each subset defines the values that the node can take [65]. Owing to their simple analysis and accuracy with various data types, decision trees have several applications. Fig. 4 illustrates the root nodes of the decision tree.

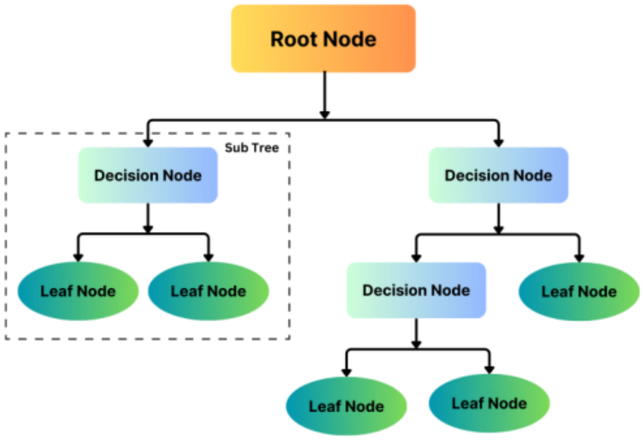



Fig. 4. Root node in a decision tree

## F. Hyperparameter Tuning

Hyperparameter tuning is critical for optimizing machine-learning models [66][67]. This involves finding the best parameters to maximize model performance [68]. However, this process can be challenging and requires a tradeoff between time and performance [69]. Hyperparameter tuning methods often demand extensive computing resources and may be task dependent [70]. Despite these challenges, hyperparameter tuning significantly affects the performance of machine learning models [71].

In this study, hyperparameter adjustment using GridSearchCV is a powerful tool in sentiment analysis, aiding the optimization of machine learning models for sentiment classification tasks. By exhaustively searching through a specified grid of hyperparameters, GridSearchCV helps identify the best combination of parameters that maximizes the model's performance in sentiment analysis [72]–[74]. This process involves systematically evaluating the model with different hyperparameter combinations and selecting the model that yields the highest accuracy or other evaluation metrics [72][73].

Various machine learning algorithms have been utilized in the context of sentiment analysis, such as support vector machines (SVM), Naïve Bayes, and deep learning techniques, to classify sentiments in textual data [75][76]. In particular, it has been widely employed for sentiment analysis owing to its effectiveness in handling high-dimensional data and binary-classification tasks [76]. In addition, the use of deep learning models such as recurrent neural networks (RNNs) and convolutional neural networks (CNNs) has shown promising results in sentiment analysis [77].

Researchers have explored various optimization techniques, such as genetic algorithms, particle swarm optimization, and randomized search optimization, to enhance the performance of sentiment analysis models [78][79]. These optimization methods aim to improve the efficiency and accuracy of sentiment classification by fine-tuning the hyperparameters of the model [78][79]. Moreover, the integration of sentiment lexicons and domain-specific features has been proposed to further enhance sentiment-analysis tasks [80].

GridSearchCV works with the basic concepts of cross-validation and parameter grid exploration and some of the underlying mathematical elements of how GridSearchCV works. Suppose there is a parameter that you want to tune, and each parameter has multiple value options. Where is the number of possible values for the i-th parameter $nP_i$.

$$Total\ Combinations = P_1\ x\ P_2\ x\ P_{...}\ x\ P_n \quad (10)$$

For each combination of parameters, GridSearchCV performs an iteration in which the model is trained on the fold and tested on the remaining fold $k - 1$.

$$Average\ Score = \frac{1}{k}\sum_{i=1}^{k} Score_i \quad (11)$$

where the score $i$ is the evaluation score of the ith fold.

Table II represents the parameter tuning for each machine learning algorithm used in this study.

TABLE II. THE PARAMETER TUNING FOR EACH ALGORITHM

| ML Algorithm | Parameters |
|---|---|
| SVM | C : [0.1, 1, 10, 15, 20]<br>Kernel : ['linear', 'rbf', 'poly']<br>Gamma : ['scale', 'auto'] |
| KNN | n_neighbors : [3, 5, 7, 9, 11]<br>weights : ['uniform', 'distance']<br>metric : ['euclidean', 'manhattan', 'minkowski'] |
| NB | Alpha : [0.5, 1.0, 1.5, 2.0] |
| LR | C : [0.001, 0.01, 0.1, 1, 10, 100]<br>Penalty : ['l2'] |
| DT | max_depth : [None, 10, 20, 30, 40, 50]<br>min_samples_split : [2, 5, 10]<br>min_samples_leaf : [1, 2, 4]<br>max_features : ['sqrt', 'log2'] |
| RF | n_estimators : [100, 200]<br>max_depth : [None, 10, 20, 30, 40, 50]<br>min_samples_split : [2, 5, 10]<br>min_samples_leaf : [1, 2, 4]<br>max_features : ['sqrt', 'log2'] |

### G. Evaluation

The dataset was pre-processed before being applied to the machine learning algorithm. Removing sentiments with neutral values. Only sentiments with bullying and non-bullying labels were processed and used for the machine learning algorithms. The data were split into training and testing sets. The model used training data to learn from the patterns and characteristics of the data. The test data evaluated the model's ability to generalize fresh, previously unexplored data unseen by the model during training. The performance of each model was evaluated using a confusion matrix including accuracy, Precision, Recall, and F1-score [22], [22]. The confusion matrix is a 2×2 matrix for binary classification with actual values on one axis and predicted values on the other. Fig. 5 illustrates the confusion matrix.

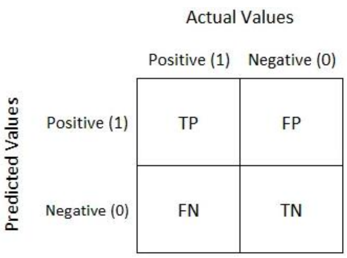



Fig. 5. Confusion matrix

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (12)$$

$$Precision = \frac{TP}{TP + FP} \quad (13)$$

$$Recall = \frac{TP}{TP + FN} \quad (14)$$

$$F1 - Score = 2\,x\,\frac{precision\ x\ recall}{precision + recall} \quad (15)$$

Where, True Positive (T.P.) is the number of positive data predicted to be true. True Negative (T.N.) is the number of negative data predicted to be true. False Positive (F.P.) is the number of negative data predicted to be positive (type I error). False Negative (F.N.) is the number of positive data predicted to be negative (type II error).

This research only produced a model that will be used as the best model for detecting cyberbullying actions. In future developments, the model can be integrated into platforms such as web-based applications or APIs to process incoming data. Additionally, testing and evaluation will be conducted after integration to ensure that the system can operate reliably and accurately in detecting cyberbullying in real time.

## III. RESULTS AND DISCUSSION

In this section, we present the results and a discussion of the proposed HyVADSVM model for cyberbullying detection. The findings are discussed in detail, emphasizing the model's performance, the impact of the various techniques used, and its effectiveness in achieving the research objectives. Additionally, we explored key aspects, such as feature extraction and model optimization, to provide a comprehensive understanding of the results obtained.

### A. Preprocessing and Feature Extraction

Preprocessing was performed before VADER was applied. Preprocessing is an essential stage in data analysis that aims to prepare raw data so that they are ready for use by machine learning algorithms. The preprocessing steps include data cleaning, such as removing punctuation, stop words, and symbols, and normalizing text by converting all letters to lowercase letters. Tokenization, stemming, and lemmatization are often used to break down texts into smaller units and unify words. This process ensures that the processed data are clean, consistent, and in the correct format to improve the accuracy and efficiency of the applied machine-learning model. Table III presents a sample of the preprocessing results for sentiments.

TABLE III. SAMPLE OF THE PREPROCESSING RESULTS

| Sentiment | Cleaning | Tokenization | Normalization |
|---|---|---|---|
| beautiful in her own style and very confident | beautiful style very confident | ['beautiful', 'style', 'very', 'confident'] | beautiful style very confident |
| the beautiful language that comes out of his mouth astaghfirullah oh god... | beautiful language that comes mouth astaghfirullah | ['beautiful', 'language', 'that', 'comes', 'mouth', 'astaghfirullah'] | beautiful language that come mouth astaghfirullah |
| "Your voice is so melodious, your appearance never disappoints." | your voice melodious your appearance never disappoints | ['your', 'voice', 'melodious', 'your', 'appearance', 'never', 'disappoints'] | your voice melodious your appearance never disappoints |
| May you soon receive punishment from Allah SWT | soon receive punishment from allah | ['soon', 'receive', 'punishment', 'from', 'allah'] | soon receive punishment from allah |

Feature extraction was then performed on the dataset using the TF-IDF technique. At this stage, data in the form of text are converted into numbers. Fig. 6 shows the results of feature extraction processing using the TF-IDF technique with the Python programming language. Fig. 6 shows that code 1 contains the line numbers of each processed data point. Code 2 is a unique integer for each word in a line. Code 3 results from the weighting (score) calculated using the TF-IDF technique.

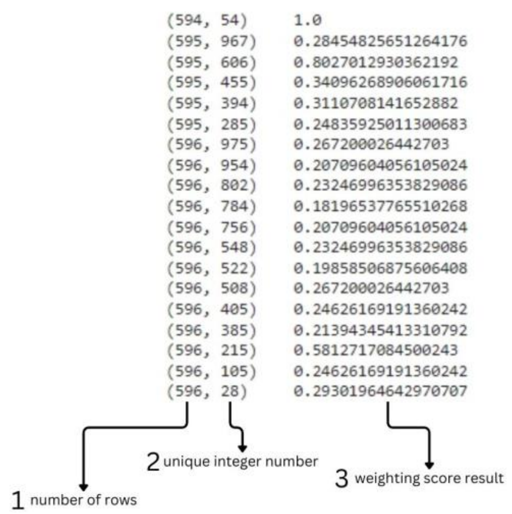



Fig. 6. TF-IDF weighting results

### B. Labeling with VADER

The results of the classification using VADER with 19,377 data points obtained as much as 98% with bullying and non-bullying labels as much as 2%, as shown in Fig. 7. Table IV presents a sample of the labeling results obtained using VADER.

TABLE IV. SAMPLE LABELING RESULTS USING VADER

| Sentiment | Compound Score | Polarity | Label |
|---|---|---|---|
| beautiful in her own style and very confident | 0.5859 | 1 | non-bullying |
| the beautiful language that comes out of his mouth astaghfirullah oh god... | -0.7506 | 0 | bullying |
| Your voice is so melodious, your appearance never disappoints | 0.5106 | 1 | non-bullying |
| May you soon receive punishment from Allah SWT | -0.7506 | 0 | bullying |

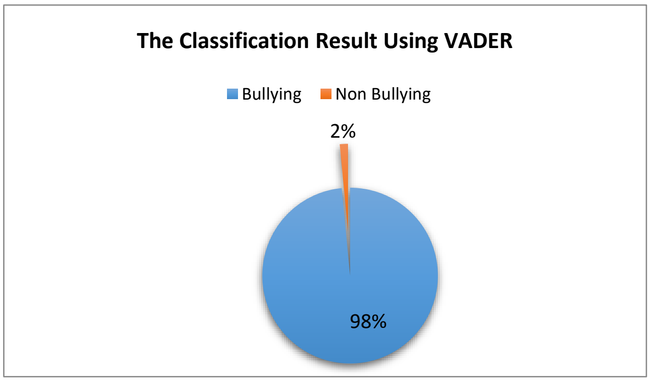



Fig. 7. The classification result using VADER

### C. Hyperparameter Tuning

This process seeks the optimal combination of hyperparameter values to improve model performance. Use GridSearchCV from Python's sci-kit-learn library to find the best combination of hyperparameters. For the HyVADSVM model, the algorithm with the highest score is the SVM. Table V lists the parameters, ranges, and types of values used in the optimization.

TABLE V. DESCRIPTION OF PARAMETERS AND VALUE RANGES

| Parameter | Range/ Type | Explanation |
|---|---|---|
| Kernel | [Linear, RBF, Poly, Sigmoid] | A transform function maps data from the Input to the higher feature space. |
| C | [0.1 – 20] | Controlling the trade-off between maximising margin and minimising misclassification |
| Gamma | [Scale, Auto] | Determine how far a single training instance can affect |

### D. Classification Results

In this section, we present the classification results obtained from the application of the developed model for cyberbullying detection. The results were divided into classifications before and after parameter tuning. These results provide an overview of the model's performance in identifying content containing cyberbullying from previously processed data. The evaluation used performance metrics, namely accuracy, precision, recall, and F1-Score. These metrics were chosen to evaluate the model's ability to accurately classify by measuring the percentage of correct predictions from the total data tested.

#### 1) Classification Before Tuning Parameter

In testing cyberbullying sentiment for processing the data using the Python programming language, it were obtained from various social media platforms, totalling 19,377 entries. Using VADER, data were classified into bullying and non-bullying sentiments. Next, we pre-processed the data. The data were split into training and testing sets in a ratio of 80:20.

A 10-fold cross-validation technique was used for model testing by randomly forming each fold. The dataset consisted of ten equal-sized subsets or folds. The division was randomly performed. This division was performed randomly, with each subgroup used as the validation data and the remaining subsets as the training data. the model ten times. In each iteration, nine folds were used to train the model and one fold was used to test the model. Fig. 1 shows the 10-fold cross-validation technique for the HyVADSVM model.

Next, we performed word weighting using TF-IDF. Subsequently, machine-learning algorithms, including SVM, K-NN, NB, DT, LR, and RF, were applied to the data that had completed preprocessing and weighting. The evaluation process uses a Confusion Matrix to visualize the performance results of the algorithm. Table VI presents the results of the confusion matrix before parameter tuning. The SVM algorithm achieved the highest values, with an accuracy of 98.80%, precision of 98.60%, recall of 98.40%, and F1-Score of 98.60%.

TABLE VI. RESULTS OF CONFUSION MATRIX EVALUATION USING MACHINE LEARNING ALGORITHM BEFORE TUNING PARAMETER

| ML Algorithm | Evaluation Metrics Before Tuning Parameters (%) | | | |
|---|---|---|---|---|
| | Accuracy | Precision | Recall | F1-Score |
| **SVM** | **98.80** | **98.60** | **98.40** | **98.60** |
| KNN | 98.23 | 97.82 | 98.23 | 97.81 |
| NB | 98.15 | 97.96 | 98.15 | 97.41 |
| LR | 98.15 | 98.19 | 98.15 | 97.37 |
| DT | 98.46 | 98.25 | 98.46 | 98.30 |
| RF | 98.78 | 98.70 | 98.78 | 98.56 |

#### 2) Classification After Tuning Parameter

In this section, we describe the experiments conducted to adjust the parameters of each algorithm to obtain the highest values from the confusion matrix. Table VII presents the confusion matrices obtained after tuning the parameters of each algorithm. Where it can be seen that the highest value was found in the SVM algorithm, followed by RF in second place, RF again in third place, DT in fourth place, KNN in fifth place, and NB in sixth place.

Based on Table VII, it is evident that the results show improved performance compared with the model before tuning. As intended, parameter tuning allows the model to determine the optimal combination of parameters, thereby maximizing accuracy and reducing prediction errors. The classification results obtained after tuning indicated that the SVM algorithm achieved the highest values among the five other algorithms, with an accuracy of 98.83%, an increase of 0.03%, a precision of 98.78%, an indication of 0.18%, a recall of 98.83%, an increase of 0.43%, an F1-score of 98.62%, and an increase of 0.02%. This demonstrated that the model was better at recognizing patterns and making more accurate predictions of the test data.

TABLE VII. RESULTS OF CONFUSION MATRIX EVALUATION USING MACHINE LEARNING ALGORITHM AFTER TUNING PARAMETER

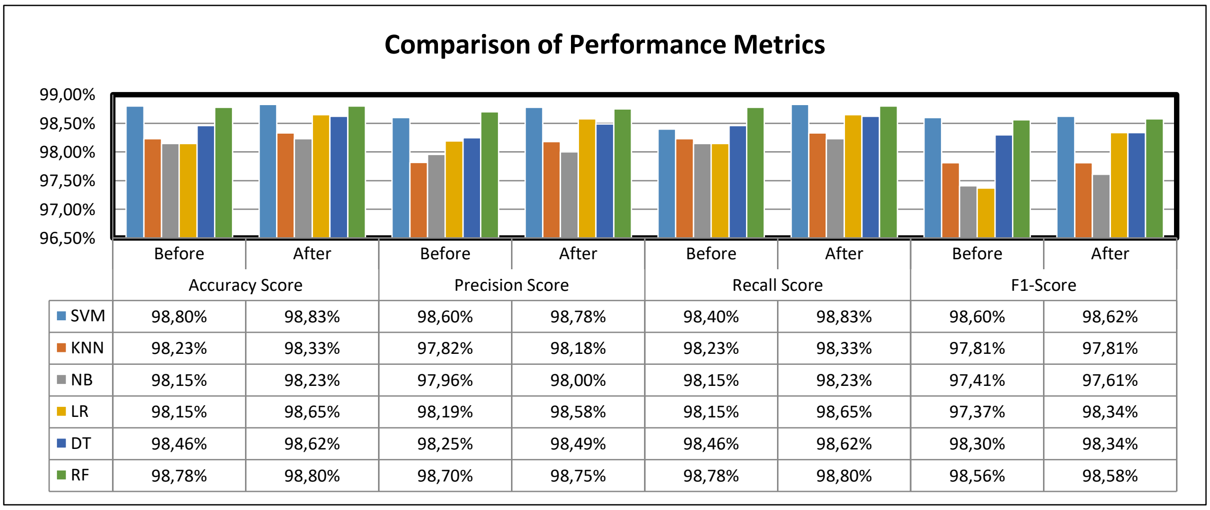
| ML Algorithm | Evaluation Metrics After Tuning Parameters (%) | | | |
|---|---|---|---|---|
| | Accuracy | Precision | Recall | F1-Score |
| **SVM** | **98.83** | **98.78** | **98.83** | **98.62** |
| KNN | 98.33 | 98.18 | 98.33 | 97.81 |
| NB | 98.23 | 98.00 | 98.23 | 97.61 |
| LR | 98.65 | 98.58 | 98.65 | 98.34 |
| DT | 98.62 | 98.49 | 98.62 | 98.34 |
| RF | 98.80 | 98.75 | 98.80 | 98.58 |

The parameters adjusted in the SVM algorithm were the Kernel, C, and Gamma. The best parameter adjustments for the SVM were Kernel=linear, C=1, and gamma =scale. Table VIII lists the tuning values applied for the adjustment. Fig. 8 show a comparison of the model evaluations from the six algorithms, demonstrating improvements after parameter tuning using GridSearchCV. Therefore, based on these figures, parameter tuning using the GridSearchCV technique proved to be effective in enhancing classification performance.

TABLE VIII. GRIDSEARCHCV PARAMETER TUNING FOR SVM

| Parameter | Best Parameter |
|---|---|
| Kernel | linear |
| C | 1 |
| Gamma | scale |

Table IX presents a comparison of various machine learning models used by several researchers in their studies. Several studies do not comprehensively report their evaluation results, such as precision, recall, and F1-Score. They only mention accuracy. From the analysis conducted, the best algorithms employed by each researcher were displayed along with their accuracy levels. The results indicate that the proposed HyVADSVM model achieved the highest accuracy of 98.83%, making it superior to the other models examined. Most other researchers showed an accuracy below 97%, with logistic regression (LR) being the best choice. The advantage of this model is due to the combination of the appropriate algorithm choice SVM with parameter tuning optimized using GridSearchCV, as well as the use of VADER sentiment analysis, which effectively captures emotional nuances in social media texts. This demonstrates the effectiveness of the HyVADSVM model in addressing similar issues and proves its significant potential for application in the field of machine learning.




| | Accuracy Score Before | Accuracy Score After | Precision Score Before | Precision Score After | Recall Score Before | Recall Score After | F1-Score Before | F1-Score After |
|---|---|---|---|---|---|---|---|---|
| SVM | 98,80% | 98,83% | 98,60% | 98,78% | 98,40% | 98,83% | 98,60% | 98,62% |
| KNN | 98,23% | 98,33% | 97,82% | 98,18% | 98,23% | 98,33% | 97,81% | 97,81% |
| NB | 98,15% | 98,23% | 97,96% | 98,00% | 98,15% | 98,23% | 97,41% | 97,61% |
| LR | 98,15% | 98,65% | 98,19% | 98,58% | 98,15% | 98,65% | 97,37% | 98,34% |
| DT | 98,46% | 98,62% | 98,25% | 98,49% | 98,46% | 98,62% | 98,30% | 98,34% |
| RF | 98,78% | 98,80% | 98,70% | 98,75% | 98,78% | 98,80% | 98,56% | 98,58% |

Fig. 8. Comparison of the model evaluations for the six machine learning algorithms


Fig. 9. 10-fold Cross-validation Technique for the HyVADSVM Model

TABLE IX. COMPARISON OF VARIOUS MACHINE LEARNING MODELS

| Authors | ML Algorithms | Best ML Algorithm | Accuracy (%) | Precision (%) | Recall (%) | F1-Score (%) |
|---|---|---|---|---|---|---|
| K. Alam et.al [30] | NB, LR, DT, SVM | LR | 94.00 | - | - | 94.00 |
| Qi. Yuxing et.al [39] | RF, NB, SVM | SVM | 71.00 | 79.00 | 73.00 | 76.00 |
| Poornima et.al [26] | NB, SVM, LR | LR | 86.00 | - | - | - |
| Satvik Garg [33] | LR, NB, SVM, SGD, RC | SVM | 93.00 | 94.00 | 96.00 | 95.00 |
| Elgeldawi et.al [67] | LR, RC, SVM, DT, RF, NB | SVM | 95.62 | - | - | - |
| Vashisht, et.al [81] | SVM | SVM | 81.00 | - | - | - |
| Thomas Renault [82] | NB, ME, SVM, RF, MP | ME | 74.45 | - | - | - |
| Mardjo et.al [61] | SVM, RF, NB, DT | RF | 75.29 | 70.22 | 87.70 | 78.00 |
| Vaidya et.al [83] | KNN, DT, LR, NB | DT | 57.69 | - | - | - |
| Priya et.al [84] | DT, RF, NB, SVM, KNN | NB | 78.00 | 82.20 | 85.00 | 83.60 |
| Shah et.al [58] | LR, RF, KNN | LR | 97.00 | 97.00 | - | 97.00 |
| **Proposed Model (HyVADSVM)** | **SVM, KNN, NB, LR, DT, RF** | **SVM** | **98.83** | **98.78** | **98.83** | **98.62** |

ML: Machine Learning; LR: Logistic Regression; SVM: Support Vector Machine; RF: Random Forest; NB: Naïve Bayes; SGD: Stochastic Gradient Descent; DT: Decision Tree; RC: Ridge Classifier; ME : Maximum Entropy; MP : Multilayer Perceptron; KNN: K-Nearest Neighbor.

## IV. CONCLUSION

In accordance with the research objective, which is to optimize cyberbullying detection by developing the HyVADSVM model, a hybrid model integrating VADER and SVM, and optimized through GridSearchCV to improve accuracy, it has been proven that this model can enhance accuracy and make accurate predictions by adjusting SVM parameters, namely Kernel, C, and Gamma. These three parameters are crucial for SVM performance, because selecting the appropriate kernel affects the model's ability to capture complex relationships within the data. Optimizing the C parameter is essential to avoid underfitting (model that is too simple) or overfitting (model that is too complex). Choosing the optimal gamma is also vital for balancing the accuracy of the training data with generalization to unseen data, ensuring that the model performs well both in training and when applied to new datasets. The sentiment labeling results using VADER showed that the percentage of bullying sentiments reached 98%, much higher than the 2% for non-bullying sentiments. The classification results obtained after parameter adjustment showed that the SVM algorithm achieved the highest value among the five algorithms. Before the parameter adjustment, the SVM model showed an accuracy of 98.80%, which increased to 98.83% after adjustment, reflecting an improvement of 0.03%. The precision increased from 98.60% to 98.78% (an increase of 0.18%), the recall from 98.40% to 98.83% (an increase of 0.43%), and the F1-Score from 98.60% to 98.62% (an increase of 0.02%).

The implications of this research cover several aspects, namely Implications for System Aspects, which show that integrating VADER sentiment analysis with SVM optimized using GridSearchCV can improve accuracy in detecting cyberbullying more accurately. By adjusting parameters such as kernel, C, and gamma in the SVM, the HyVADSVM model can effectively classify cyberbullying content with high accuracy, making it a relevant method for detecting language patterns containing bullying elements. Implications for Managerial Aspects: The research model can be applied by technology companies or social media platforms in their efforts to enhance user security and detect cyberbullying behavior. Using this model, companies can make strategic decisions regarding content moderation policies more effectively based on sentiment analysis of user interactions on their platforms. Additionally, this research can be applied by educational institutions or other organizations to monitor online activities to prevent harmful behaviors among young users or specific communities. With more accurate analysis, institutions can identify negative trends in online interactions and implement preventive measures more quickly. Implications for Future Research open opportunities for further development in cyberbullying detection, including the development of other hybrid models with more advanced natural language processing (NLP) methods or deep learning.

Moreover, future research can focus on testing larger and more diverse datasets, including informal languages or slang commonly used on social media. Adding features such as user feedback to model predictions can also improve the accuracy and adaptability of future models. Thus, the implications of this research include not only the technical aspects of developing detection systems, but also the social aspects of creating a safer digital environment for users. One limitation of this research is the handling of ambiguous language, where text containing double meanings or unclear sentences may sometimes reduce detection accuracy. Therefore, further research is required to address this limitation.

## ACKNOWLEDGMENT

The authors extend their deepest gratitude to Direktorat Riset, Teknologi, dan Pengabdian kepada Masyarakat (DRTPM), Kementerian Pendidikan, Kebudayaan, Riset, dan Teknologi, for the funding support that made this research possible. Thanks are also extended to our research colleagues and the entire team, who provided guidance and support throughout the research process. Appreciation goes to family and friends for their moral support as well as to everyone who contributed to the completion of this research.

## REFERENCES


[1] Á. Denche-zamorano *et al.*, “Science Mapping : A Bibliometric Analysis on Cyberbullying and the Psychological Dimensions of the Self,” *Int. J. Environ. Res. Public Health*, vol. 20, no. 209, pp. 2–14, 2023, doi: 10.3390/ ijerph20010209.

[2] A. Ollagnier, E. Cabrio, S. Villata, and S. Tonelli, “BiRDy: Bullying Role Detection in Multi-Party Chats,” in *Proceedings of the 37th AAAI Conference on Artificial Intelligence, AAAI 2023*, vol. 37, 2023, doi: 10.1609/aaai.v37i13.27080.

[3] F. B. Shaikh, M. Rehman, S. Member, M. A. Hashmani, and S. Member, “Cyberbullying Behaviour : A Study of Undergraduate University Students,” *IEEE Access*, vol. 9, pp. 92715–92734, 2021, doi: 10.1109/ACCESS.2021.3086679.

[4] N. A. Abdul Rahman, N. Hussein, S. D. Rusdi, and Z. D. Abd Aziz, “Factors Influencing Cyberbullying Behavior among Malaysian Tertiary Students,” *Adv. Bus. Res. Int. J.*, vol. 6, no. 2, 2020, doi: 10.24191/abrij.v6i2.10594.

[5] A. G. Philipo, D. S. Sarwatt, J. Ding, M. Daneshmand, and H. Ning, "Cyberbullying Detection: Exploring Datasets, Technologies, and Approaches on Social Media Platforms," *arXiv preprint arXiv:2407.12154*, 2024.

[6] C. Slamet, A. Krismunandar, D. S. Maylawati, Jumadi, A. S. Amin, and M. A. Ramdhani, "Deep Learning Approach for Bullying Classification on Twitter Social Media with Indonesian Language," *2020 6th International Conference on Wireless and Telematics (ICWT)*, pp. 1-5, 2020, doi: 10.1109/ICWT50448.2020.9243653.

[7] M. Alfi, S. Nasution, and E. B. Setiawan, “Enhancing Cyberbullying Detection on Indonesian Twitter : Leveraging FastText for Feature Expansion and Hybrid Approach Applying CNN and BiLSTM,” *Int. Inf. Eng. Technol. Assoc.*, vol. 37, no. 4, pp. 929–936, 2023, doi: 10.18280/ria.370413.

[8] I. A. Asqolani and E. B. Setiawan, “Hybrid Deep Learning Approach and Word2Vec Feature Expansion for Cyberbullying Detection on Indonesian Twitter,” *Int. Inf. Eng. Technol. Assoc.*, vol. 28, no. 4, pp. 887–895, 2023, doi: 10.18280/isi.280410.

[9] W. Frensh, M. Ablisar, and M. Mulyadi, “Criminal Policy on Cyberbullying of Children in Indonesia,” *Int. J. Cyber Criminol.*, vol. 15, pp. 44–59, 2021, doi: 10.5281/zenodo.4766542.

[10] L. H. Collantes, Y. Martafian, S. N. Khofifah, T. Kurnia Fajarwati, N. T. Lassela, and M. Khairunnisa, "The Impact of Cyberbullying on Mental Health of the Victims," *2020 4th International Conference on Vocational Education and Training (ICOVET)*, pp. 30-35, 2020, doi: 10.1109/ICOVET50258.2020.9230008.

[11] E. J. ALjohani, W. M. S. Yafooz, and A. Alsaeedi, "Cyberbullying Detection Approaches: A Review," *2023 5th International Conference on Inventive Research in Computing Applications (ICIRCA)*, pp. 1310-1316, 2023, doi: 10.1109/ICIRCA57980.2023.10220688.

[12] S. Utku and E. Nasibov, “Cyberbullying Detection : Utilizing Social Media Features,” *Expert Syst. Appl.*, vol. 179, pp. 1–12, 2021, doi: 10.1016/j.eswa.2021.115001.

[13] T. K. Balaji, C. Sekhara, R. Annavarapu, and A. Bablani, “Machine Learning Algorithms for Social Media Analysis : A Survey,” *Comput. Sci. Rev.*, vol. 40, no. 100395, pp. 1–32, 2021, doi: 10.1016/j.cosrev.2021.100395.

[14] N. I. Aliyu, M. D. Abdulrahaman, F. O. Ajibade, and T. Abdurauf, “Analysis of Cyber Bullying on Facebook Using Text Mining,” *J. Appl. Artif. Intell.*, vol. 1, pp. 1–12, 2020, doi: 10.48185/jaai.v1i1.30.

[15] A. Muneer, “A Comparative Analysis of Machine Learning Techniques for Cyberbullying Detection on Twitter,” *Electron. J.*, vol. 10, no. 22, pp. 1–20, 2020, doi: 10.3390/fi1211018.

[16] M. Mohamad and A. Al-quran, “Sentiment Analysis Technique and Neutrosophic Set Theory for Mining and Ranking Big Data From Online Reviews,” *IEEE Access*, vol. 9, no. 3, pp. 47338–47353, 2021, doi: 10.1109/ACCESS.2021.3067844.

[17] T. Hasan, A. E. Hossain, S. H. Mukta, A. Akter, M. Ahmed, and S. Islam, “A Review on Deep-Learning-Based Cyberbullying Detection,” *Future Internet*, vol. 15, no. 5, p. 179, 2023, doi: 10.3390/fi15050179.

[18] P. Kumar, R. Pamula, and G. Srivastava, “A Systematic Literature Review On Machine Learning Applications For Consumer Sentiment Analysis Using Online Reviews,” *Elsevier - Comput. Sci. Rev.*, vol. 41, pp. 1–12, 2021, doi: 10.1016/j.cosrev.2021.100413.

[19] S. Kumar, M. Gahalawat, P. P. Roy, D. P. Dogra, and B.-G. Kim, “Exploring Impact of Age and Gender on Sentiment Analysis Using Machine Learning,” *Electronics*, vol. 9, no. 374, pp. 1–14, 2020, doi: 10.3390/electronics9020374.

[20] K. Thavasimani and N. K. Srinath, “Hyperparameter Optimization Using Custom Genetic Algorithm For Classification Of Benign And Malicious Traffic On Internet Of Things – 23 Dataset,” *Int. J. Electr. Comput. Eng.*, vol. 12, no. 4, pp. 4031–4041, 2022, doi: 10.11591/ijece.v12i4.pp4031-4041.

[21] P. Das and D. H. Mazumder, “An Extensive Survey on The Use of Supervised Machine Learning Techniques in The Past Two Decades for Prediction of Drug Side Effects,” *Artif. Intell. Rev.*, vol. 56, no. 9, pp. 9809–9836, 2023, doi: 10.1007/s10462-023-10413-7.

[22] P. C. Sen, M. Hajra, and M. Ghosh, “Supervised Classification Algorithms in Machine Learning: A Survey and Review,” in *Advances in Intelligent Systems and Computing*, vol. 937, pp. 99–111, 2020, doi: 10.1007/978-981-13-7403-6.

[23] A. Ma and D. Rifaldi, “Application of Machine Learning in Healthcare and Medicine : A Review,” *J. Robot. Control*, vol. 4, no. 5, pp. 621–631, 2023, doi: 10.18196/jrc.v4i5.19640.

[24] A. A. Aziz and A. Starkey, “Predicting Supervise Machine Learning Performances for Sentiment Analysis Using Contextual-Based Approaches,” *IEEE Access*, vol. 8, pp. 17722–17733, 2020, doi: 10.1109/ACCESS.2019.2958702.

[25] N. Nandal, R. Tanwar, and J. Pruthi, “Machine Learning Based Aspect Level Sentiment Analysis For Amazon Products,” *Springer - Spat. Inf. Res.*, vol. 28, pp. 601–607, 2020, doi: 10.1007/s41324-020-00320-2.

[26] A. Poornima and K. S. Priya, “A Comparative Sentiment Analysis Of Sentence Embedding Using Machine Learning Techniques,” in *2020 6th International Conference on Advanced Computing & Communication Systems (ICACCS)*, pp. 493–496, 2020, doi: 10.1109/ICACCS48705.2020.9074312.

[27] N. F. Baarir and A. Djeffal, "Fake News detection Using Machine Learning," *2020 2nd International Workshop on Human-Centric Smart Environments for Health and Well-being (IHSH)*, pp. 125-130, 2021, doi: 10.1109/IHSH51661.2021.9378748.

[28] M. Al-hashedi, L. Soon, H. Goh, A. H. L. Lim, and E.-G. Siew, “Cyberbullying Detection Based on Emotion,” *IEEE Access*, pp. 1–9, 2023, doi: 10.1109/ACCESS.2023.3280556.

[29] S. Gadde, A. Lakshmanarao, and S. Satyanarayana, "SMS Spam Detection using Machine Learning and Deep Learning Techniques," *2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS)*, pp. 358-362, 2021, doi: 10.1109/ICACCS51430.2021.9441783.

[30] K. S. Alam, S. Bhowmik, and P. R. K. Prosun, “Cyberbullying Detection : An Ensemble Based Machine Learning Approach,” in *Proceedings of the Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV 2021)*, pp. 710–715, 2021, doi: 10.1109/ICICV50876.2021.9388499.

[31] S. Jain, S. K. Jain, and S. Vasal, "An Effective TF-IDF Model to Improve the Text Classification Performance," *2024 IEEE 13th International Conference on Communication Systems and Network Technologies (CSNT)*, pp. 1-4, 2024, doi: 10.1109/CSNT60213.2024.10545818.

[32] M. Khairy, T. M. Mahmoud, and T. Abd-El-Hafeez, “The effect of rebalancing techniques on the classification performance in cyberbullying datasets,” *Neural Comput. Appl.*, vol. 36, no. 3, pp. 1049–1065, 2024, doi: 10.1007/s00521-023-09084-w.

[33] S. Garg, “Drug Recommendation System based on Sentiment Analysis of Drug Reviews using Machine Learning,” in *International Conference on Cloud Computing, Data Science & Engineering*, pp. 175–181, 2021, doi: 10.1109/confluence51648.2021.9377188.

[34] R. Phann, C. Soomlek, and P. Seresangtakul, “Multi-Class Text Classification on Khmer News Using Ensemble Method in Machine Learning Algorithms,” *Acta Inform. Pragensia*, vol. 12, no. 2, pp. 1–17, 2023, doi: 10.18267/j.aip.210.

[35] A. Radhakrishnan, G. S. Member, and D. Mahapatra, “Consumer Document Analytical Accelerator Hardware,” *IEEE Access*, vol. 11, pp. 5161–5167, 2023, doi: 10.1109/ACCESS.2023.3237463.

[36] Z. Zhang and Z. Wu, "Improved TF-IDF Algorithm Combined with Multiple Factors," *2021 3rd International Conference on Applied Machine Learning (ICAML)*, pp. 492-495, 2021, doi: 10.1109/ICAML54311.2021.00109.

[37] R. Annamalai, S. A. Rasool, D. S, K. Venkatraman, and S. Y, "Sentiment Analysis using VADER: Unveiling Customer Sentiment and Predicting Buying Behavior in the Airline Industry," *2024 IEEE International Conference for Women in Innovation, Technology & Entrepreneurship (ICWITE)*, pp. 277-282, 2024, doi: 10.1109/ICWITE59797.2024.10503389.

[38] D. A. Kristiyanti, S. A. Sanjaya, V. C. Tjokro, and J. Suhali, “Dealing Imbalance Dataset Problem in Sentiment Analysis of Recession in Indonesia,” *IAES Int. J. Artif. Intell.*, vol. 13, no. 2, pp. 2060–2072, 2024, doi: 10.11591/ijai.v13.i2.pp2060-2072.

[39] Y. Qi and Z. Shabrina, “Sentiment analysis using Twitter data : a comparative application of lexicon - and machine - learning - based approach,” *Soc. Netw. Anal. Min.*, vol. 13, no. 1, pp. 1–14, 2023, doi: 10.1007/s13278-023-01030-x.

[40] M. Suyal and P. Goyal, “A Review on Analysis of K-Nearest Neighbor Classification Machine Learning Algorithms based on Supervised Learning,” *Int. J. Eng. Trends Technol.*, vol. 70, no. 7, pp. 43–48, 2022, doi: 10.14445/22315381/IJETT-V70I7P205.

[41] N. Burkart and M. F. Huber, “A Survey on the Explainability of Supervised Machine Learning,” *J. Artif. Intell. Res.*, vol. 70, pp. 245–317, 2021, doi: 10.1613/jair.1.12228.

[42] Y. Zheng, T. Deng, and Y. Wang, “Autism Classification Based On Logistic Regression Model,” in *2nd International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering (ICBAIE 2021) Autism*, pp. 579–582, 2021, doi: 10.1109/ICBAIE52039.2021.9389914.

[43] T. Wong and H. Tsai, “Knowledge-Based Systems Multinomial naïve Bayesian classifier with generalized Dirichlet priors for high-dimensional imbalanced data,” *Knowledge-Based Syst.*, vol. 228, p. 107288, 2021, doi: 10.1016/j.knosys.2021.107288.

[44] P. J. B. Pajila, B. G. Sheena, A. Gayathri, J. Aswini, M. Nalini, and S. S. R, "A Comprehensive Survey on Naive Bayes Algorithm: Advantages, Limitations and Applications," *2023 4th International Conference on Smart Electronics and Communication (ICOSEC)*, pp. 1228-1234, 2023, doi: 10.1109/ICOSEC58147.2023.10276274.

[45] E. Hossain, O. Sharif, and M. M. Hoque, “Sentiment Polarity Detection on Bengali Book Reviews Using Multinomial Naïve Bayes,” in *Proceedings of ICACIE 2020*, pp. 281–292, 2020, doi: 10.1007/978-981-33-4299-6.

[46] T. A. Assegie, “An optimized K-Nearest Neighbor based breast cancer detection,” *J. Robot. Contro;*, vol. 2, no. 3, pp. 115–118, 2020, doi: 10.18196/jrc.2363.

[47] S. Zhang, S. Member, and J. Li, “KNN Classification With One-Step Computation,” *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 3, pp. 2711–2723, 2023, doi: 10.1109/TKDE.2021.3119140.

[48] A. Lu, Z. Fang, N. Farahpour, and L. Shannon, "CHIP-KNN: A Configurable and High-Performance K-Nearest Neighbors Accelerator on Cloud FPGAs," *2020 International Conference on Field-Programmable Technology (ICFPT)*, pp. 139-147, 2020, doi: 10.1109/ICFPT51103.2020.00027.

[49] S. Zhang and S. Member, “Challenges in KNN Classification,” *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 10, pp. 4663–4675, 2022, doi: 10.1109/TKDE.2021.3049250.

[50] A. Sachdeva and D. I. Kashyap, “Empirical Analysis of Support Vector Machine and Multinomial Naive Bayes,” *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 10, no. 5, 2022, doi: 10.22214/ijraset.2022.42009.

[51] S. Shi, L. Ye, H. Xiao, and T. Luo, "A Comparative Study of Mainstream Machine Learning Algorithms for Spine Health Prediction," *2022 16th ICME International Conference on Complex Medical Engineering (CME)*, pp. 349-355, 2022, doi: 10.1109/CME55444.2022.10063325.

[52] D. Chaerul, E. Saputra, A. Ma, and K. Sunat, “Optimizing Predictive Performance : Hyperparameter Tuning in Stacked Multi-Kernel Support Vector Machine Random Forest Models for Diabetes Identification,” *J. Robot. Control*, vol. 4, no. 6, pp. 896–904, 2023, doi: 10.18196/jrc.v4i6.20898.

[53] A. Ma, D. Rifaldi, and A. Azzawagama, “Comparison of Convolutional Neural Networks and Support,” *IJRCS-International J. Robot. Control Syst.*, vol. 4, no. 1, pp. 445–462, 2024, doi: 10.31763/ijrcs.v4i1.1375.

[54] V. Piccialli and M. Sciandrone, “Nonlinear Optimization and Support Vector Machines,” *Ann. Oper. Res.*, vol. 16, 2022, doi: 10.1007/s10479-022-04655-x.

[55] J. Alcaraz, M. Labbé, and M. Landete, “Support Vector Machine with feature selection: A multiobjective approach,” *Expert Syst. Appl.*, vol. 204, 2022, doi: 10.1016/j.eswa.2022.117485.

[56] F. B. Bağci and Ö. Karal, “Exploring Efficient Kernel Functions For Support Vector Clustering,” *Mugla J. Sci. Technol.*, pp. 36–42, 2020, doi: 10.22531/muglajsci.703790.

[57] I. S. Al-mejibli, J. K. Alwan, and D. H. Abd, “The Effect Of Gamma Value On Support Vector Machine Performance With Different Kernels,” *Int. J. Electr. Comput. Eng.*, vol. 10, no. 5, pp. 5497–5506, 2020, doi: 10.11591/ijece.v10i5.pp5497-5506.

[58] K. Shah, H. Patel, D. Sanghvi, and M. Shah, “A Comparative Analysis of Logistic Regression , Random Forest and KNN Models for the Text Classification,” *Augment. Hum. Res.*, vol. 5, no. 12, pp. 1–16, 2020, doi: 10.1007/s41133-020-00032-0.

[59] Y. Fan, J. Bai, X. Lei, Y. Zhang, B. Zhang, and K. Li, “Journal of Network and Computer Applications Privacy preserving based logistic regression on big data,” *J. Netw. Comput. Appl.*, vol. 171, p. 102769, 2020, doi: 10.1016/j.jnca.2020.102769.

[60] R. Fezai, K. Dhibiv, M. Mansouri, and M. Trabelsi, “Effective Random Forest-Based Fault Detection and Diagnosis for Wind Energy,” *IEEE Sens. J.*, vol. 21, no. 5, pp. 6914–6921, 2021, doi: 10.1109/JSEN.2020.3037237.

[61] A. Mardjo and C. Choksuchat, “HyVADRF : Hybrid VADER – Random Forest and GWO for Bitcoin Tweet Sentiment Analysis,” *IEEE Access*, vol. 10, pp. 101889–101897, 2022, doi: 10.1109/ACCESS.2022.3209662.

[62] K. Dhibi *et al.*, “Reduced Kernel Random Forest Technique for Fault Detection and Classification in Grid-Tied PV Systems,” *IEEE J. Photovoltaics*, vol. 10, no. 6, pp. 1–8, 2020, doi: 10.1109/JPHOTOV.2020.3011068.

[63] B. T. Jijo and A. M. Abdulazeez, “Classification Based on Decision Tree Algorithm for Machine Learning,” *J. Appl. Sci. Technol. TRENDS*, vol. 2, no. 1, pp. 20–28, 2021, doi: 10.38094/jastt20165.

[64] L. Cañete-sifuentes, R. Monroy, and M. A. Medina-pérez, “A Review and Experimental Comparison of Multivariate Decision Trees,” *IEEE Access*, vol. 9, pp. 110451–110479, 2021, doi: 10.1109/ACCESS.2021.3102239.

[65] Z. He, Z. Wu, G. Xu, Y. Liu, and Q. Zou, “Decision Tree for Sequences,” *IEEE Trans. Knowl. Data Eng.*, vol. 7, no. 1041, pp. 1–15, 2020, doi: 10.1109/TKDE.2021.3075023.

[66] J. A. Ilemobayo *et al.*, “Hyperparameter Tuning in Machine Learning: A Comprehensive Review,” *J. Eng. Res. Reports*, vol. 26, no. 6, pp. 388–395, Jun. 2024, doi: 10.9734/JERR/2024/V26I61188.

[67] 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. 79, pp. 1–21, 2021, doi: 10.3390/informatics8040079.

[68] A. E. Minarno, M. H. C. Mandiri, Y. Munarko, and H. Hariyady, “Convolutional Neural Network with Hyperparameter Tuning for Brain Tumor Classification,” *Kinet. Game Technol. Inf. Syst. Comput. Network, Comput. Electron. Control*, vol. 6, no. 2, pp. 127–132, May 2021, doi: 10.22219/KINETIK.V6I2.1219.

[69] M. Krček and G. Perin, “Autoencoder-enabled Model Portability for Reducing Hyperparameter Tuning Efforts in Side-channel Analysis,” *Journal of Cryptographic Engineering*, vol. 14, no. 3, pp. 475-497, 2024, doi: 10.1007/s13389-023-00330-4.

[70] J. Huang, J. Rojas, M. Zimmer, H. Wu, Y. Guan, and P. Weng, “Hyperparameter auto-tuning in self-supervised robotic learning,” *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 3537–3544, Apr. 2021, doi: 10.1109/LRA.2021.3064509.

[71] S. Mohapatra, S. Sasy, X. He, G. Kamath, and O. Thakkar, “The Role of Adaptive Optimizers for Honest Private Hyperparameter

Selection," *Proc. AAAI Conf. Artif. Intell.*, vol. 36, no. 7, pp. 7806–7813, Jun. 2022, doi: 10.1609/AAAI.V36I7.20749.

[72] A. Chaiyana *et al.*, "Leveraging Remotely Sensed and Climatic Data for Improved Crop Yield Prediction in the Chi Basin, Thailand," *Sustain.*, vol. 16, no. 6, p. 2260, Mar. 2024, doi: 10.3390/SU16062260.

[73] F. Naim, A. E. Cook, and J. Moortgat, "Estimating Compressional Velocity and Bulk Density Logs in Marine Gas Hydrates Using Machine Learning," *Energies*, vol. 16, no. 23, p. 7709, Dec. 2023, doi: 10.3390/EN16237709/S1.

[74] H. Darmawan, M. Yuliana, and M. Z. S. Hadi, "GRU and XGBoost Performance with Hyperparameter Tuning Using GridSearchCV and Bayesian Optimization on an IoT-Based Weather Prediction System," *Int. J. Adv. Sci. Eng. Inf. Technol.*, vol. 13, no. 3, 2023, doi: 10.18517/ijaseit.13.3.18377.

[75] A. Wan *et al.*, "Using Transfer Learning and XGBoost for Early Detection of Fires in Offshore Wind Turbine Units," *Energies*, vol. 17, no. 10, p. 2330, May 2024, doi: 10.3390/EN17102330.

[76] H. Niu, Z. Zhang, and M. Luo, "Evaluation and Prediction of Low-Carbon Economic Efficiency in China, Japan and South Korea: Based on DEA and Machine Learning," *Int. J. Environ. Res. Public Heal.*, vol. 19, no. 19, p. 12709, Oct. 2022, doi: 10.3390/IJERPH191912709.

[77] M. Jha, R. Gupta, and R. Saxena, "Convolutional neural network based detection of lung adenocarcinoma by amalgamating hybrid features," *Int. J. Adv. Technol. Eng. Explor.*, vol. 11, no. 111, pp. 160–176, 2024, doi: 10.19101/IJATEE.2023.10102196.

[78] T. Alshammari, "Using Artificial Neural Networks with GridSearchCV for Predicting Indoor Temperature in a Smart Home," *Eng. Technol. Appl. Sci. Res.*, vol. 14, no. 2, pp. 13437–13443, Apr. 2024, doi: 10.48084/ETASR.7008.

[79] Jumanto *et al.*, "Optimizing Support Vector Machine Performance for Parkinson's Disease Diagnosis Using GridSearchCV and PCA-Based Feature Extraction," *J. Inf. Syst. Eng. Bus. Intell.*, vol. 10, no. 1, pp. 38–50, Feb. 2024, doi: 10.20473/JISEBI.10.1.38-50.

[80] 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," *Mater.*, vol. 15, no. 21, p. 7432, Oct. 2022, doi: 10.3390/MA15217432.

[81] G. Vashisht, "Sentimental study of CAA by location-based tweets," *Int. J. Inf. Technol.*, vol. 13, no. 4, pp. 1555–1567, 2021, doi: 10.1007/s41870-020-00604-8.

[82] T. Renault, "Sentiment Analysis and Machine Learning In Finance: A Comparison Of Methods And Models On One Million Messages," *Springer - Digit. Financ.*, pp. 1–13, 2020, doi: 10.1007/s42521-019-00014-x.

[83] P. M. V. P. Vaidya, R. Korde, O. Chaudhari, P. Chirde, S. Deoda, and D. Kadu, "Twitter Sentiment Analysis Using Machine Learning Algorithm," *Int. J. Ingenious Res. Invent. Dev.*, vol. 3, no. 1, pp. 106–114, 2024, doi: 10.5281/zenodo.10967535.

[84] A. Priya, S. Garg, and N. P. Tigga, "Predicting Anxiety , Depression and Stress in Modern Life using Machine Learning Algorithms," in *Procedia Computer Science - International Conference on Computational Intelligence and Data Science (ICCIDS 2019)*, vol. 167, pp. 1258–1267, 2020, doi: 10.1016/j.procs.2020.03.442.