SHM PUBLISHER
SURYA HIJAU MANFAAT

Journal of Information System Exploration and Research

https://shmpublisher.com/index.php/joiser

p-ISSN 2964-1160 | e-ISSN 2963-6361

Enhancing Abusive Language Detection on Twitter Using Stacking Ensemble Learning

Putri Utami[1]*, Yulizchia Malica Pinkan Tangai[2], Jumanto Unjung[3], Much Aziz Muslim[4]
[1,2,3]Department of Computer Science, Faculty of Mathematics and Natural Science, Universitas Negeri Semarang, Indonesia
[4]Faculty of Technology Management and Business, Universiti Tun Hussein Onn Malaysia, Malaysia

DOI: https://doi.org/10.52465/joiser.v3i2.594
Received 07 July 2025; Accepted 05 August 2025; Available online 19 August 2025

Article Info

Keywords:
Abusive language;
Twitter;
Stacking;
Ensemble learning

Abstract

Detecting abusive language on Twitter is an important step in reducing the prevalence of negative content and harassment. This study aims to improve the accuracy and effectiveness of abusive language detection on Twitter by addressing the limitations of the single model commonly used previously. The stacking method is employed by combining Term Frequency-Inverse Document Frequency (TF-IDF) as the feature extraction method, along with the Naive Bayes and XGBoost algorithms as classification models. Naive Bayes is known for its simplicity in handling text classification, while XGBoost excels in processing complex data and achieving high accuracy. The combination of these two models is expected to improve performance in detecting coarse language. The research results show that the proposed model outperforms the methods in previous studies, with an accuracy of 91.91% and an AUC of 96.76%. These findings demonstrate the effectiveness of the stacking approach in reducing classification errors in coarse language detection. Further research could explore the use of larger datasets or more complex models to improve detection accuracy.

1. Introduction

Twitter is a platform that is widely used to exchange messages with fellow users. There are several features in Twitter, such as tweets, retweets, handles, hashtags, follow, and search [1]. Tweets are short messages with a limit of 140 characters that can contain text, photos, videos, and links [2], while a retweet is the act of reposting without any additional content from the user [3]. When at the beginning of a user's name the '@' symbol is given to refer to other people or other organizations, it is called a handle [3] and when the '#' symbol is followed by one or more words behind it, it is used to group tweets based on certain topics called hashtags [4]. The follow feature is used to receive tweet updates from accounts that users follow in real-time [1], and the search feature allows users to find tweets or content relevant to what they are looking for using keywords or phrases in real-time [1].

* ***Corresponding Author:***

Puri Utami,
Department of Computer Science,
Universitas Negeri Semarang,
Sekaran, Gunungpati, Semarang, Indonesia.
Email: utamiputri575@students.unnes.ac.id

In recent years, Twitter has experienced rapid growth and has become it for various types of communication, including news sharing, discussions on trending topics, and others. However, Twitter's open nature also brings challenges, such as the wide dissemination of information that may include offensive language, which can lead to negative content that can adversely affect users.

Abusive language on Twitter can take many forms, such as insults, profanity, verbal abuse, or hate speech. This language is often directed at attacking a person's identity, such as their race, religion, gender, sexual orientation, or physical condition [5]. Abusive language not only creates an unhealthy communication environment but can also trigger social conflicts, increase psychological pressure on victims, and even affect the mental health of users in general. In a broader context, the spread of abusive language can also damage the reputation of a community and negatively influence public opinion. Therefore, abusive language detection is very important because it helps reduce hate speech and harassment on social media platforms.

Various approaches have been proposed in previous research to address this issue. Some studies used traditional machine learning techniques such as Naive Bayes [6], support vector machines (SVM) [7], [8], and logistic regression [8] to detect coarse language, and provided quite good performance. Although simple and fast, these methods are sometimes suboptimal on complex data. Deep learning techniques such as CNNs [9] and Bi-LSTMs [8], [11] have shown better results in some cases, but require large computational resources and difficult interpretation [11]. Therefore, ensemble-based approaches such as stacking are emerging as a more efficient alternative and can optimize the power of various models.

In this study, we propose a novel approach to detect abusive words on Twitter by applying Term Frequency-Inverse Document Frequency (TF-IDF) as a feature extraction method combined with a stacking model that integrates XGBoost and Naive Bayes classifiers. This stacking technique utilizes the strengths of both models: Naive Bayes is known for its simplicity and effectiveness in handling text classification [12], [14], while XGBoost excels in handling complex data and providing accurate predictions [14]. By combining these two models, we aim to improve accuracy and robustness in coarse language detection.

The novelty of this research lies in the application of a stacking technique that combines two traditional machine learning models by applying TF-IDF for coarse language detection. This approach has not been widely explored in previous studies, which usually only use one model, combine two simple models, or focus on deep learning methods. The use of a combination of XGBoost and Naive Bayes models with TF-IDF as a feature extraction method can provide a more efficient solution. The findings of this research are expected to make a significant contribution to the development of abusive language detection systems on social media.

2. Method

This study consists of importing data from Kaggle, preprocess data, feature extraction using TF-IDF, model building using stacking Naive-Bayes and XGBoost, and model evaluation using accuracy, precision, recall, and f1-score. Figure 1 explains the flow of this research.

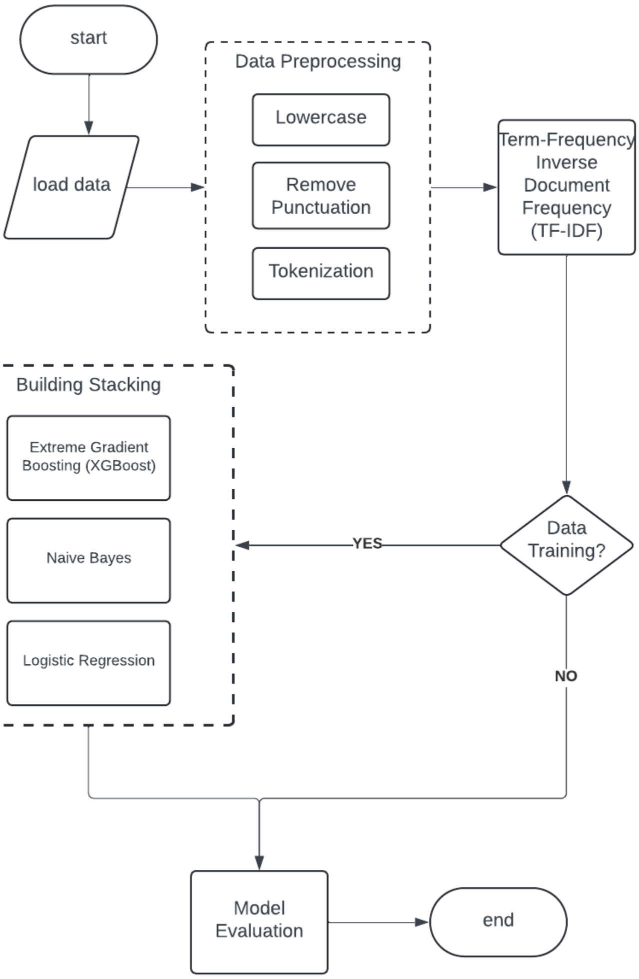

Figure 1. Flowchart of the research

2.1. Dataset

The dataset is acquired from Kaggle, where the author used the dataset in this paper [15]. The author on Kaggle states that the dataset is a re-uploaded version from the original author on GitHub, so all credit goes to the GitHub author [16]. The dataset contains 13 features such as 'Tweet', 'HS', 'Abusive', 'HS_Individual', 'HS_Group', 'HS_Religion', 'HS_Race', 'HS_Physical', 'HS_Gender', 'HS_Other', 'HS_Weak', 'HS_Moderate', 'HS_Strong', and 13.169 data. This research uses only 2 features, which are ‘Tweet’ and ‘Abusive’, to detect abusive words in the dataset. Figure 2 shows the value of each class in the ‘Abusive’ column.

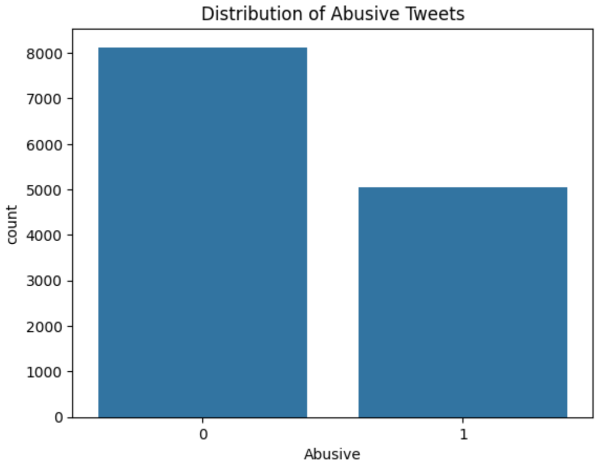

Figure 2. Data count of each class

2.2. Preprocessing Data

The next stage is data preprocessing, which aims to clean and prepare the data [18]. There are several preprocessing stages in this research, namely lowercase, removing punctuation, and tokenization. The lowercase stage is a preprocessing stage that changes all letter sizes to small. The purpose of lowercase is so that there are no differences due to capitalization. Next is the data cleaning process that removes all characters, such as punctuation marks, symbols, and numbers. Then, tokenization, where the data is broken down into tokens. After the cleaning process, the remaining tokens will be recombined into one sentence.

2.3. TF-IDF Feature Extraction

Term-Frequency Inverse Document Frequency (TF-IDF) is an algorithm used to evaluate the significance of the term or word related to larger corpus of the document [18]. TF-IDF calculated based on two things, Term Frequency to assess the importance of a term within document by counting appereance of a term in a document or corpus and Inverse Document Frequency, where it assesses the significance of a term across the entirety of the document collection by calculating the logarithm of the ratio between the total number of documents and the number of documents containing the term [19]. Formula (1), (2), (3) are used to calculate the TF-IDF.

$$TF = \frac{frequency\ of\ the\ term\ in\ a\ word}{total\ word\ in\ corpus} \tag{1}$$

$$IDF = \log \frac{total\ word\ in\ corpus}{total\ word\ in\ corpus\ containing\ the\ term} \tag{2}$$

$$TF - IDF = TF \times IDF \tag{3}$$

2.4. Detection Model Building

After the data is processed, the next step is to divide the data into 2 sets. 80% training set to train the model so that it can "learn" from the data, and 20% test set to evaluate the performance of the model after training. Then this training set will train the proposed model of this research.

2.4.1 Naive-Bayes

A naive Bayes classifier is a probabilistic model that makes predictions based on Bayes' theorem, assuming that the features are independent [20]. This study uses Multinomial Naive-Bayes where a probabilistic classifier that operates under the assumption that the features of the input data are independent of one another. It estimates the probability of each class based on the features [21]. Implementation of Naive-Bayes in this study by using Multinomial Naive-Bayes from the sklearn library with default hyperparameters. Parameter θ_y is estimated using relative frequency calculation which is a refined version of maximum likelihood estimation (MLE). The relative frequency calculation formula is mentioned in formula (4).

$$\hat{\theta}_{yi} = \frac{N_{y_i} + \alpha}{N_y + \alpha n} \tag{4}$$

Where N_{yi} Is the number of occurrences of feature i in samples in class y in the training dataset T. While N_y, is the total number of N_{yi} For class y. To avoid zero probabilities that can cause problems in the model, a smoothing prior $\alpha \geq 0$ is used in the model. There are 2 types of smoothing, Laplace smoothing when $\alpha = 1$ and Lidstone smoothing when $\alpha < 1$.

2.4.2. XGBoost

XGBoost was developed by Chen and Guestrin in 2016 [22] to advance the GB algorithm. This was achieved by incorporating several additional features, including regularisation and tree pruning, which mitigate overfitting [23]. XGBoost Implementation in this study uses XGBClassifier from the XGBoost library. The hyperparameter is then tuned using RandomizedSearchCV with the hyperparameters listed in Table 1.

Table 1. Hyperparameter list for XGBoost

Hyperparameter	Value List
n_estimators	100, 200, 300
max_depth	3, 4, 5
learning_rate	0.01, 0.1, 0.3
reg_alpha	0.01, 0.1, 1
Reg_lambda	0.01, 0.1, 1

2.4.3. Stacking

Stacking is an ensemble method where the outputs of many classifiers are used to create a meta-classifier for final classification [25]. This method works by combining 2 or more models into one meta-classifier to improve model prediction [26]. Stacking is used to reduce misclassification by combining many classifiers into one meta-classifier model, so as to improve prediction accuracy [27]. Stacking in this study is implemented using StackingClassifier from the sklearn library. The estimators included in StackingClassifier are the Naive-Bayes model and the XGBoost model. Before stacking, the prediction results from the XGBoost model were optimised using RandomisedSearchCV to search for parameters randomly and effectively. Logistic regression was used as the final estimator to prevent overfitting [28]. The stacking method of ensemble learning in this study can be seen in Figure 3.

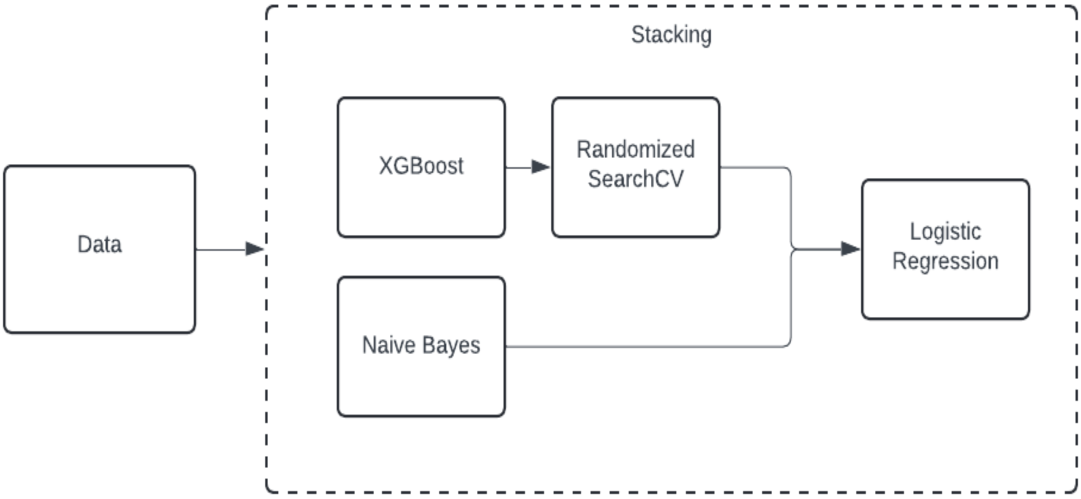

Figure 3. Flowchart of the stacking method

2.5. Model Evaluation

The model will be evaluated using 3 3-step Stratified Cross-validation. Metrics used to evaluate the model are accuracy, precision, recall, and F1-score are mentioned in formulas (5), (6), (7), (8), (9), and (10).

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{5}$$

$$Precision = \frac{TP}{TP + FP} \tag{6}$$

$$Precision = \frac{TP}{TP + FN} \tag{7}$$

$$F1 - Score = 2 \times \frac{(Precision \times Recall)}{(Precision + Recall)} \tag{8}$$

$$TPR = \frac{TP}{TP + FN} \quad (9)$$

$$FPR = \frac{FP}{FP + TN} \quad (10)$$

Where TP is True Positive, where the model correctly predicts the positive class. TN is True Negative, where the model correctly predicts the negative class. FP is False Positive, where the model incorrectly predicts the positive class when the actual class is negative, FN is False Negative, the model incorrectly predicts the negative class when the actual class is positive. TPR and FPR are used to plot the Receiver Operating Characteristic (ROC).

3. Results and Discussion

The results of this research are in the form of a confusion matrix and a classification report. A confusion matrix is used to find out how well the model can distinguish between classes and the possibility of the model making mistakes by displaying the number of true positives, false positives, true negatives, and false negatives. Figure 4 shows the confusion matrix of the model.

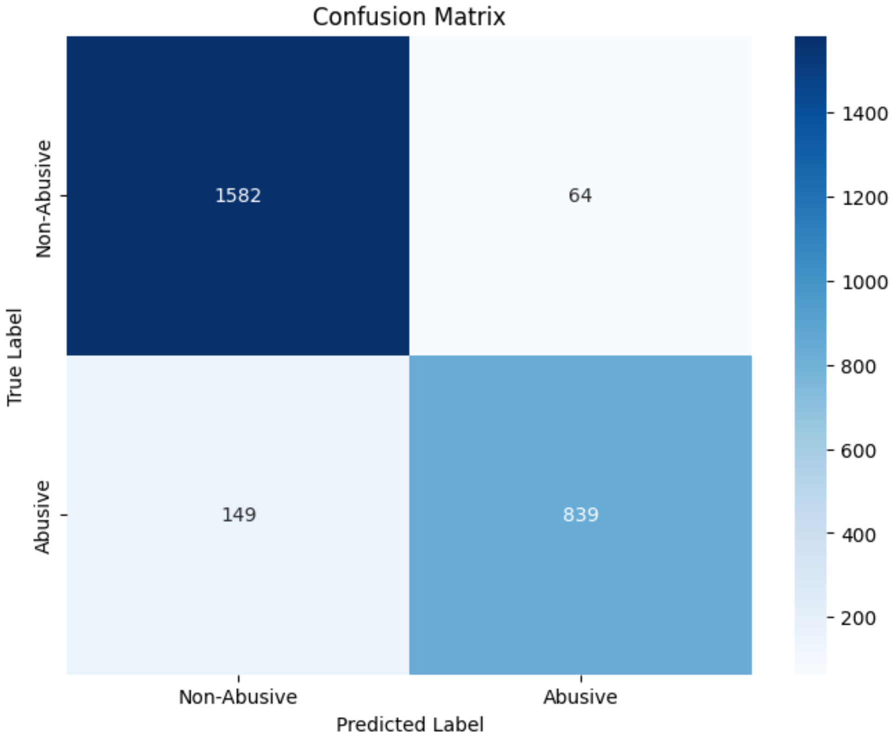

Figure 4. Confusion matrix

Figure 4 above shows that the True Positive 839 data are actually "Abusive" and correctly classified by the model as "Abusive". False Positive, there are 64 data points that are actually "Not Abusive" but wrongly classified by the model as "Abusive". True Negative: 1582 data points are actually "Not Abusive" and have been correctly classified by the model as "Not Abusive". Then False Negative there are 149 data points that are actually "Rough" but have been misclassified by the model as "Not Rough". In the True Positive (TP) and True Negative (TN) sections, there are high numbers. This shows that the model has an overall good performance. However, the False Positive (FP) and False Negative (FN) sections show that the model still makes mistakes. This matrix illustrates that the model can distinguish between the "Abusive" and "Non-Abusive" classes.

In line with the confusion matrix, the Stacking Model successfully classified the test data with high accuracy. Based on the evaluation results, the model misclassified some class 1 data as class 0, where class 1 has a recall of 84.92%. The classification report of this model is shown in Table 2.

Table 2. Classification report

	Precision	**recall**	**F1-score**	**support**
0 (non-abusive)	91.39%	96.11%	93.69%	1646
1 (abusive)	92.91%	84.92%	88.74%	988
Accuracy			91.91%	2634
Macro avg	92.15%	90.52%	91.21%	2634
Weighted avg	91.96%	91.91%	91.83%	2634

Figure 5 shows the Receiver Operating Characteristic (ROC) curve of the Stacking Classifier model prediction results. This ROC curve helps in illustrating the performance of the model at various classification thresholds, where the threshold can determine whether the data is classified as a positive or negative class. If the area under the curve ROC (AUC) is close to 1, then the model has a good performance in class classification [28].

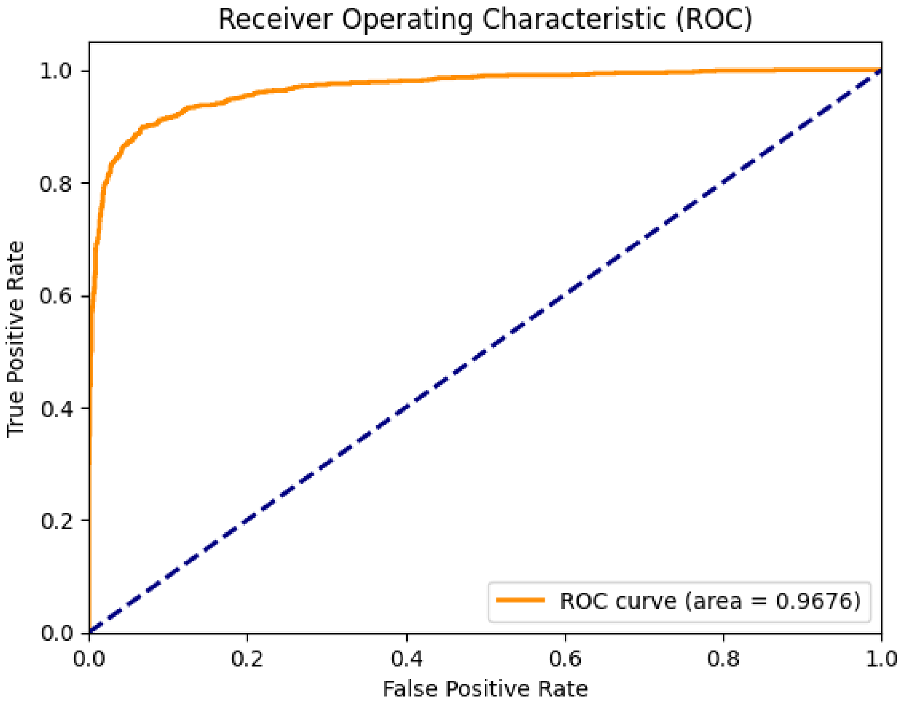

Figure 5. Receiver operating characteristics (ROC)

The model is reliable for classification tasks as it can reduce both positive and negative class misidentification with an AUC of 96.76%. The Stacking Classifier achieves consistent performance in complex classification problems. The results obtained in this study using the Stacking model will be compared with previous research models. The model comparison is shown in Table 3.

Table 3. Comparison with previous research

Author	Method	Accuracy	F1-score	AUC
Reinaldo Yosafat, et al [30]	CNN	91.31%	91.31%	-
Rahmat Hendrawan, et al [31]	RFDT	82.36%	-	-
Muhammad Razi Mahardika, et al [32]	BERT	88%	-	96%
Proposed Method	**Stacking Classifier**	**91.91%**	**93.69%**	**96.76%**

In research [30] and [31] used the same dataset was used but with different methods. Penelitian oleh Reinaldo Yosafat dkk menggunakan CNN yang mencapai akurasi dan skor F1 sebesar 91,31%. Penelitian oleh Hendrawan dkk mengusulkan Random Forest Decision Tree (RFDT) yang mencapai akurasi sebesar 82,36%. Another study conducted by Muhammad Razi Mahardika, et al, using BERT achieved an accuracy of 88% and an AUC of 96%. Meanwhile, this study proposes the use of Stacking classification, which combines Naive Bayes and XGBoost to produce an accuracy of 91.91% and an AUC of 96.76%. The accuracy results obtained show that this Stacking model has better prediction capabilities compared to models in previous studies.

4. Conclusion

The aim of this research, described in the Introduction, has been to successfully detect abusive language on Twitter with a Stacking approach that combines TF-IDF, XGBoost, and Naive Bayes. The results in this study show that the proposed model excels with an accuracy of 91.91% and an AUC of 96.76%, surpassing previous methods. This shows that the combination of both models improves accuracy and reduces classification errors. In further development, the model can be tested with a larger and more diverse dataset or with other social media platforms. In addition, trying different types of more sophisticated models can improve the accuracy of detection and prediction results in abusive language detection.

References

[1] Y. Wang, J. Guo, C. Yuan, and B. Li, "Sentiment Analysis of Twitter Data," *Appl. Sci.*, vol.. 12, no. 22, pp. 1–14, 2022, doi: 10.3390/app122211775.

[2] Y. Li and Y. Xie, "Is a Picture Worth a Thousand Words? An Empirical Study of Image Content and Social Media Engagement," *J. Mark. Res.*, vol. 57, no. 1, pp. 1–19, 2020, doi: 10.1177/0022243719881113.

[3] A. Maleki and K. Holmberg, "Tweeting and retweeting scientific articles: implications for altmetrics," *Scientometrics*, no. 0123456789, 2024, doi: 10.1007/s11192-024-05127-8.

[4] Y. Djenouri, A. Belhadi, G. Srivastava, and J. C. W. Lin, "Toward a Cognitive-Inspired Hashtag Recommendation for Twitter Data Analysis," *IEEE Trans. Comput. Soc. Syst.*, vol. 9, no. 6, pp. 1748–1757, 2022, doi: 10.1109/TCSS.2022.3169838.

[5] E. W. Pamungkas, V. Basile, and V. Patti, *Investigating the role of swear words in abusive language detection tasks*, vol. 57, no. 1. Springer Netherlands, 2023. doi: 10.1007/s10579-022-09582-8.

[6] R. Shukla and M. Vidhwani, "and Engineering Trends Electricity Theft Detection Using Machine Learning," vol. 4, no. 9, pp. 2019–2021, 2020.

[7] R. Gupta, J. Kumar, H. Agrawal, and Kunal, "A Statistical Approach for Sarcasm Detection Using Twitter Data," *Proc. Int. Conf. Intell. Comput. Control Syst. ICICCS 2020*, no. Iciccs, pp. 633–638, 2020, doi: 10.1109/ICICCS48265.2020.9120917.

[8] R. Arifudin, D. I. Wijaya, B. Warsito, and A. Wibowo, "Voting Classifier Technique and Count Vectorizer with N-gram to Identify Hate Speech and Abusive Tweets in Indonesian," vol. 10, no. 4, pp. 469–478, 2023, doi: 10.15294/sji.v10i4.46633.

[9] F. Rodriguez-Sanchez, J. Carrillo-De-Albornoz, and L. Plaza, "Automatic Classification of Sexism in Social Networks: An Empirical Study on Twitter Data," *IEEE Access*, vol. 8, pp. 219563–219576, 2020, doi: 10.1109/ACCESS.2020.3042604.

[10] M. Amjad, N. Ashraf, G. Sidorov, A. Zhila, L. Chanona-Hernandez, and A. Gelbukh, "Automatic Abusive Language Detection in Urdu Tweets," *Acta Polytech. Hungarica*, vol. 19, no. 10, pp. 143–163, 2022, doi: 10.12700/APH.19.10.2022.10.9.

[11] P. Utami, M. R. Ningsih, D. Ananda, and A. Pertiwi, “Sentimen based-emotion classification using bidirectional long,” pp. 281–289, 2024.

[12] K. Tzoumpas, A. Estrada, P. Miraglio, and P. Zambelli, “A Data Filling Methodology for Time Series Based on CNN and (Bi)LSTM Neural Networks,” *IEEE Access*, vol. 12, no. January, pp. 31443–31460, 2024, doi: 10.1109/ACCESS.2024.3369891.

[13] K. M. El Hindi, R. R. Aljulaidan, and H. AlSalman, “Lazy fine-tuning algorithms for naïve Bayesian text classification,” *Appl. Soft Comput. J.*, vol. 96, p. 106652, 2020, doi: 10.1016/j.asoc.2020.106652.

[14] A. R. Safitri and M. A. Muslim, “Improved Accuracy of Naive Bayes Classifier for Determination of Customer Churn Uses SMOTE and Genetic Algorithms,” pp. 70–75, 2020.

[15] S. Li and X. Zhang, “Research on orthopedic auxiliary classification and prediction model based on XGBoost algorithm,” *Neural Comput. Appl.*, vol. 32, no. 7, pp. 1971–1979, 2020, doi: 10.1007/s00521-019-04378-4.

[16] M. O. Ibrohim and I. Budi, “Multi-label Hate Speech and Abusive Language Detection in Indonesian Twitter,” pp. 46–57, 2019, doi: 10.18653/v1/w19-3506.

[17] I. F. Putra and A. Purwarianti, “Improving Indonesian Text Classification Using Multilingual Language Model,” *2020 7th Int. Conf. Adv. Informatics Concepts, Theory Appl. ICAICTA 2020*, 2020, doi: 10.1109/ICAICTA49861.2020.9429038.

[18] Rofik, R. Aulia, K. Musaadah, S. Shafira, F. Ardyani, and A. A. Hakim, “The Optimization of Credit Scoring Model Using Stacking Ensemble Learning and Oversampling Techniques,” *J. Inf. Syst. Explor. Res.*, vol. 2, no. 1, pp. 11–20, 2024.

[19] M. Kayest and S. K. Jain, “Optimization driven cluster based indexing and matching for the document retrieval,” *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 34, no. 3, pp. 851–861, 2022, doi: 10.1016/j.jksuci.2019.02.012.

[20] L. C. Chen, “An extended TF-IDF method for improving keyword extraction in traditional corpus-based research: An example of a climate change corpus,” *Data Knowl. Eng.*, vol. 153, no. September 2023, p. 102322, 2024, doi: 10.1016/j.datak.2024.102322.

[21] P. Mohseni and A. Ghorbani, “Exploring the synergy of artificial intelligence in microbiology: Advancements, challenges, and future prospects,” *Comput. Struct. Biotechnol. Reports*, vol. 1, no. June, p. 100005, 2024, doi: 10.1016/j.csbr.2024.100005.

[22] R. Alanazi and S. Alanazi, “A hybrid NLP and domain validation technique for disposable email detection,” *Alexandria Eng. J.*, vol. 102, no. May, pp. 200–210, 2024, doi: 10.1016/j.aej.2024.05.068.

[23] T. Chen and C. Guestrin, “XGBoost: A scalable tree boosting system,” *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, vol. 13-17-Augu, pp. 785–794, 2016, doi: 10.1145/2939672.2939785.

[24] M. Niazkar *et al.*, “Applications of XGBoost in water resources engineering: A systematic literature review (Dec 2018–May 2023),” *Environ. Model. Softw.*, vol. 174, no. January, p. 105971, 2024, doi: 10.1016/j.envsoft.2024.105971.

[25] R. Islam and M. A. Layek, “StackEnsembleMind: Enhancing well-being through accurate identification of human mental states using stack-based ensemble machine learning,” *Informatics Med. Unlocked*, vol. 43, no. August, p. 101405, 2023, doi: 10.1016/j.imu.2023.101405.

[26] D. Ling, T. Jiang, J. Sun, Y. Wang, Y. Wang, and L. Wang, “An Ensemble Learning System Based on Stacking Strategy for Survival Risk Prediction of Patients with Esophageal Cancer,” *Irbm*, vol. 45, no. 6, p. 100860, 2024, doi: 10.1016/j.irbm.2024.100860.

[27] M. A. Muslim *et al.*, “New model combination meta-learner to improve accuracy prediction P2P lending with stacking ensemble learning,” *Intell. Syst. with Appl.*, vol. 18, no. February, p. 200204, 2023, doi: 10.1016/j.iswa.2023.200204.

[28] A. Parvez, S. D. Ali, H. Tayara, and K. T. Chong, “Stacking based ensemble learning framework for identification of nitrotyrosine sites,” *Comput. Biol. Med.*, vol. 183, no. May, p. 109200, 2024, doi: 10.1016/j.compbiomed.2024.109200.

[29] J. Miao and W. Zhu, “Precision–recall curve (PRC) classification trees,” *Evol. Intell.*, vol. 15, no. 3, pp. 1545–1569, 2022, doi: 10.1007/s12065-021-00565-2.

[30] R. Y. Gultom, F. I. Zulkarnaen, Y. Nurhasanah, and A. Sholahuddin, “Indonesian Abusive Tweet Classification based on Convolutional Neural Network and Long Short Term Memory Method,” *2021 Int. Conf. Artif. Intell. Big Data Anal. ICAIBDA 2021*, pp. 121–126, 2021, doi:

10.1109/ICAIBDA53487.2021.9689728.

[31] R. Hendrawan, Adiwijaya, and S. Al Faraby, “Multilabel Classification of Hate Speech and Abusive Words on Indonesian Twitter Social Media,” *2020 Int. Conf. Data Sci. Its Appl. ICoDSA 2020*, 2020, doi: 10.1109/ICoDSA50139.2020.9212962.

[32] M. R. Mahardika, I. P. J. Wijaya, A. R. Prayoga, H. Lucky, and I. A. Iswanto, “Exploring the Performance of BERT Models for Multi-Label Hate Speech Detection on Indonesian Twitter,” *2023 4th Int. Conf. Artif. Intell. Data Sci. Discov. Technol. Adv. Artif. Intell. Data Sci. AiDAS 2023 - Proc.*, pp. 256–261, 2023, doi: 10.1109/AiDAS60501.2023.10284596.