Erzincan Üniversitesi
Fen Bilimleri Enstitüsü Dergisi
2024, 17(2), 445-459
ISSN: 1307-9085, e-ISSN: 2149-4584
Araştırma Makalesi

Erzincan University
Journal of Science and Technology
2024, 17(2), 445-459
DOI: 10.18185/erzifbed.1474112
Research Article

Performance Analysis of NLP-Based Machine Learning Algorithms in Cyberbullying Detection

Funda AKAR[1*]

[1]Department of Computer Engineering, Faculty of Engineering and Architecture, Erzincan Binali Yıldırım University, Erzincan, Türkiye

Received: 26/04/2024, **Revised:** 25/06/2024, **Accepted:** 25/06/2024, **Published:** 31/08/2024

Abstract

In today's pervasive online landscape, the escalating threat of cyberbullying demands advanced detection and mitigation tools. This study utilizes Natural Language Processing (NLP) techniques to confront this imperative challenge, particularly in the dynamic realm of social media, focusing on tweets. A comprehensive NLP-based classification methods is deployed to uncover instances of cyberbullying. Nine prominent machine learning algorithms are meticulously evaluated: Logistic Regression, Decision Tree, Random Forest, Naive Bayes, K-Nearest Neighbor, Support Vector Machine, XGBoost, AdaBoost, and Gradient Boosting. Through the analysis, encompassing accuracy, precision, recall, and F1 score metrics, the study offers insights into the strengths and limitations of each approach. The findings carry profound implications for online user safeguarding and cyberbullying prevalence reduction. Notably, Random Forest and XGBoost classifiers emerge as pioneers with accuracy rates of 93.34% and 93.32%, respectively. This comparative research underscores the pivotal role of expert algorithmic choices in addressing the urgency of cyberbullying and has the potential to be a valuable resource for academics and practitioners engaged in combatting this pressing societal issue.

Keywords: Cyberbullying, Machine Learning, Multi-Class Classification, Natural Language Processing (NLP).

Siber Zorbalık Tespitinde NLP Tabanlı Makine Öğrenimi Algoritmalarının Performans Analizi

Öz

Günümüzün yaygın çevrimiçi ortamında, artan siber zorbalık tehdidi, gelişmiş tespit ve azaltma araçlarını gerektirmektedir. Bu çalışma, özellikle sosyal medyanın dinamik dünyasında, tweet'lere odaklanarak bu zorunlu zorlukla yüzleşmek için Doğal Dil İşleme (NLP) tekniklerinden yararlanmaktadır. Siber zorbalık örneklerini ortaya çıkarmak için kapsamlı NLP tabanlı sınıflandırma yöntemleri kullanılmıştır. Öne çıkan dokuz makine öğrenimi algoritması titizlikle değerlendirilmiştir: Lojistik Regresyon, Karar Ağacı, Rastgele Orman, Naive Bayes, K-En Yakın Komşu, Destek Vektör Makinesi, XGBoost, AdaBoost ve Gradient Boosting. Doğruluk, kesinlik, geri çağırma ve F1 puanı metriklerini kapsayan analiz aracılığıyla çalışma, her yaklaşımın güçlü yönlerine ve sınırlamalarına dair içgörüler sunmaktadır. Bulgular, çevrimiçi kullanıcıların korunması ve siber zorbalığın yaygınlığının azaltılması konusunda derin etkiler taşımaktadır. Özellikle Random Forest ve XGBoost sınıflandırıcıları sırasıyla %93,34 ve %93,32 doğruluk oranlarıyla öncü olarak ortaya çıkmıştır. Bu karşılaştırmalı araştırma, siber zorbalığın aciliyetine değinerek güçlü algoritmik seçimlerin önemli rolünün altını çizmekte ve bu acil toplumsal sorunla mücadele eden akademisyenler ve uygulayıcılar için değerli bir kaynak olma potansiyeline sahiptir.

Anahtar Kelimeler: Siber Zorbalık, Makine Öğrenimi, Çok Sınıflı Sınıflandırma, Doğal Dil İşleme (DDİ).

*Corresponding Author: fakar@erzincan.edu.tr
Funda AKAR, https://orcid.org/0000-0001-9376-8710

1. Introduction

Bullying is any physical or verbal behavior that one person does to another person. This behavior is seen in almost every age group and is a serious social problem. On the other hand, Cyberbullying is the behavior done with various communication tools used on the internet. This type of behavior can be done in all kinds of social media, as well as frequently seen in social media [1]. Cyberbullying, which has increased a lot in recent years, has become difficult to prevent. Because the number of people using social media has increased considerably. Due to the increasing number of users and inadequate detection systems, this behavior, which annoys and humiliates other people, has unfortunately become unavoidable. There are two roles in cyberbullying. One of them is the person who is cyberbullying, and the other is the person who is suffering from this situation. Considering all types of cyberbullying, there may be various reasons behind the behavior of the person doing this job [2]. According to Xu et.al., there are many roles in cyberbullying: cyber bully, victim, accuser, advocate, reinforcer, bystander, reporter and the helper [3]. In order to solve this problem, automatic systems can be developed and relevant solutions can be created. Since written language is generally used in social media, the solution to the problem can be provided by natural language processing. It is very important to develop a system for detecting cyberbullying and for this system to prevent such bad events. Because the expanding number of social media users includes all kinds of people. In such systems, audits are usually carried out by a manager or a management team assigned to these tasks. However, it is really difficult to audit manually as it is a very large auditing environment [4].

One area that has seen extensive development over the years is natural language processing (NLP). Within the field of Natural Language Processing (NLP), diverse syntactic structures serve as pivotal tools for drawing conclusions or making assumptions in both multi-class and single-class research contexts, encompassing domains such as emotional states and textual representations of actions. Consequently, a myriad of scenarios emerges, each amenable to multiple methodological approaches. Accordingly, the employment of NLP based machine learning algorithms for classification purposes is deemed appropriate for scholarly inquiry. The comprehension of human language entails a nuanced progression through distinct stages akin to communicative interactions. These stages are systematically categorized to facilitate comprehensive analysis [5]. Numerous prominent corporations have successfully developed and implemented Natural Language Processing (NLP) systems, a trend that has witnessed widespread adoption within industrialized frameworks in recent years. Consequently, NLP has attained considerable reputation and effective, enabling the resolution of diverse challenges through automated systems, thereby obviating the necessity for traditional one-on-one customer service interactions and yielding notable time and cost savings. This paradigm shift has in turn allowed increased investment in research and the refinement of NLP system capabilities [6].

There has also been a lot of work on cyberbullying and cyber violence [7]–[12]. While conducting the literature research, the methods that can be considered as closer as possible were

emphasized with the study conducted in order to make comparisons. The study mainly includes datasets and the methods used, as well as performance evaluation.

In a study conducted for the detection of cyberbullying, 5453 tweets were tried to be made and a classification process was carried out by examining various variations such as baseline, personalities, emotion and sentiment. Values such as accuracy, f1-score, AUC were used as evaluation metrics. As a result of the study, about 90% success was achieved and the classification process was carried out. With the addition of an extended feature and the pre-processing processes applied, the result has been improved [13]. Another study conducted in 2019, dataset was taken from the site called Kaggle and the data obtained includes texts written on cyberbullying made on social media. The data was labelled and classified as cyberbullying or not cyberbullying, SVM and neural network were used for classification. It was determined that the neural network gave better results. Considering the scores obtained in the neural network in the 2-gram, 3-gram and 4-gram results, success rates above 90% were obtained [14]. Raza et al studied a dataset on the activities of PTA members on their website for cyberbullying in Japan. Using many machine learning methods for classification, the highest results were found with logistic regression to 82.7%. They then increased their success rate to 84.4% using supervised machine learning. In the study, it is stated that the main factor that leads to success is the voting classifier [15]. As another example of cyberbullying, pre-processing has been done for machine learning-based detection. Here, stop-word, repeating letters, punctuation, tokenization, Vector Space Model are used. In this way, the preliminary preparation necessary for the dataset to be evaluated, that is, for its classification, has been completed. The classification in the dataset has three classes that it is divided into positive, neutral and negative. Machine learning types used for classification are also SVM and Naive Bayes. As a result of the study, it is seen that the highest success rates are 89.54% in SVM and 73.03% in Naive Bayes [16]. Using multilingual data, [17] suggested a tailored deep network model to recognize and promote optimism in comments. Their approach obtained macro F1 scores of 75% for English, 62% for Tamil, and 67% for Malayalam using the CNN model by combining embedding from T5-Sentence. In another study the experiments made use of four deep learning models: RNN, LSTM, GRU, and BLSTM. In comparison to RNN, LSTM, and GRU, the BLSTM model was achieved the highest accuracy (82.18%) and the highest F1-measure score (88%) [18].

2. Material and Methods

Nine different machine learning algorithms were used in the study and the main purpose is to use too many algorithms and to determine which algorithm works better in such problems. It is inevitable to see what kind of results are obtained in a multi-class structure used in the dataset, and it will shed light when using it for other datasets. While evaluating the various classes in the dataset, both the display of weighted results and their performance on a class basis are shown. The dataset was taken from Twitter, which contains a large amount of data and is labeled. This data set has a 5-class structure and is multi-class. Since the classes are almost equal to each other, they are in a better position to give the results of the study. In the study, it

was used in pure form without any changes with the data. Then, using NLP methods, the texts in the dataset were made suitable for working with pre-processing.

After the pre-processing processes, a wide variety of machine learning methods were used to classify the dataset using machine learning methods. Here, a multi-class dataset has been classified using many machine learning methods and the methods have been compared. No parameter changes were made in the machine learning methods used here and default values were used. Better results are likely to be found by performing cross-validation. What is seen as important here is to show which machine learning method performs better.

The data set used in the study consists of tweets taken from Twitter. It contains 47 thousand tweets in total. The reason for choosing this data set is that multiple classes are evenly distributed in the data set. The dataset was accessed via the Kaggle site [19], [20]. Tweets related to cyberbullying were collected and labelled in the dataset. There is a balanced classification distribution, approximately 8 thousand of each class. You can see the percentage representation of the distribution in Figure 1.

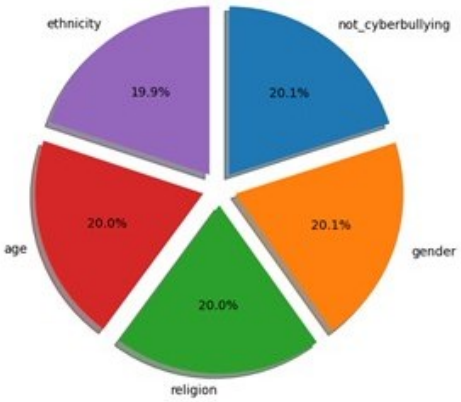

Figure 1. Distribution of classes in the dataset.

The dataset includes a total of 5 classes: ethnicity, religion, gender, age and not_cyberbullying. Since it is not appropriate to share the word cloud content in the data, it is not included in the article. Basic pre-processing techniques used in natural language processing are used. Then, this data was divided into train and test sets and tested with many different machine learning algorithms. The flowchart of the study is given in Figure 2.

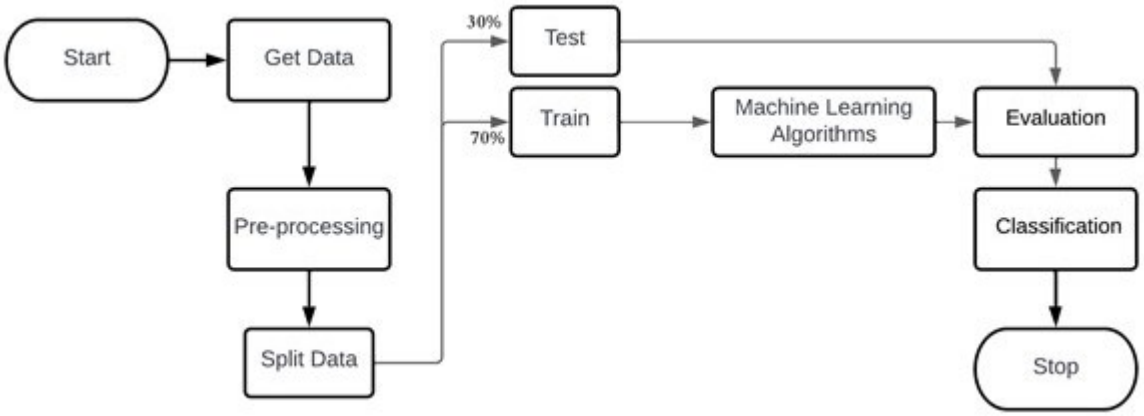

Figure 2. Flowchart of the study.

2.1. Pre-processing

This section employs classical natural language processing methods, as certain preliminary operations are requisite for accurate sentence classification within the dataset, thereby influencing the overall success of the classification endeavor. Specifically, the Natural Language Toolkit (NLTK) was employed for preprocessing tasks in the study [21]. NLTK, a comprehensive library, facilitates various operations including but not limited to eliminating redundant letters within sentences, performing lemmatization, and conducting tokenization procedures [22]. Due to its comprehensive features, NLTK is extensively utilized in various applications. The pre-processing methods employed in the study encompass:

- Lower text: Convert all text to lowercase. This is done to ensure uniformity and avoid the model treating uppercase and lowercase versions of the same word differently.
- Tokenize text [20] and remove punctation: Tokenization involves breaking down the text into individual words or tokens. Punctuation removal involves eliminating punctuation marks from the text. This step helps in creating a clean and standardized set of words.
- Remove words that contain numbers: This step is often performed to focus on the textual content and remove any alphanumeric characters.
- Remove stop words: Stop words are common words (e.g., "and", "the", "is") that are often removed from text data because they don't carry significant meaning for certain types of analysis. Removing them can reduce noise in the data.
- Remove empty tokens: This step involves removing any empty tokens that might result from tokenization or other preprocessing steps. These empty tokens do not contribute to the analysis and need to be discarded.
- Lemmatized text: Lemmatization involves reducing words to their base or root form. For example, "running" would be lemmatized to "run." This step helps in reducing words to their core meaning and can improve the consistency of the dataset.
- Remove words with only one letter. Such words are often considered less informative and might not contribute much to the analysis.

These methods are commonly employed to clean and prepare text data for analysis. The pre-processing methods involved in preparing textual data prior to submitting it to machine learning techniques are intended to produce a purer and more uniform dataset, thereby enhancing its suitability for computational processing. These measures include removing unwanted characters, normalizing the text, and converting all words into lowercase. Depending on the unique features of the data and the objectives of the investigation, these methods may vary. To transform the text information into numerical vectors amenable to machine learning models, TFIDF (Term Frequency-Inverse Document Frequency) transformation is then applied before exporting the prepared data to the algorithm [14].

2.2. Machine Learning Algorithms

Machine learning algorithms have been broadened to be widely evaluated. While evaluating machine learning in the study, both class-based achievements and weighted average results were shared. Machine learning algorithms are concluded with default values. Therefore, it is likely that better performances will emerge if the studies to be carried out are worked on. The

main purpose here is to reveal which algorithms should be used to approach the results in this problem and to solve the problem. Machine learning algorithms used in the study:

- Logistic Regression (LR): Logistic Regression is a statistical and machine learning algorithm used for binary classification tasks, where the goal is to predict one of two possible outcomes (usually denoted as 0 and 1, or "negative" and "positive"). It's called "logistic" because it uses the logistic function (also known as the sigmoid function) to model the probability of the binary outcome. Logistic Regression is widely used in various fields, including healthcare (disease diagnosis), marketing (customer churn prediction), and natural language processing (sentiment analysis) [23]–[25].

- Decision Tree Classifier (DTC): Decision Trees are a popular machine learning algorithm used for both classification and regression tasks. They are a powerful and interpretable model that makes decisions by recursively splitting the data into subsets based on the most significant features. Each split is based on a decision rule, creating a tree-like structure, hence the name "Decision Tree" [26], [27].

- Random Forest Classifier (RFC): Random Forest is an ensemble machine learning algorithm that is widely used for both classification and regression tasks. It is built upon the foundation of Decision Trees and offers several advantages, including improved accuracy and reduced overfitting [28]–[31].

- Naive Bayes (NB): Naive Bayes is a simple yet effective probabilistic machine learning technique that is used for classification and, to a lesser extent, regression applications. It is founded on Bayes' theorem and makes the "naive" assumption of feature independence, which simplifies modeling and probability calculation [32]–[34].

- K- Nearest Neighbor (K-NN): A straightforward and user-friendly supervised machine learning technique for classification and regression applications is the k-Nearest Neighbors (K-NN) algorithm. Being a non-parametric method, it bases its predictions on how similar the data points in the training dataset are to one another. Based on the dominant class or average value of a data point's k-nearest neighbors, K-NN is used to categorize it or create a regression forecast [35], [36].

- Extreme Gradient Boosting-XGBoost (XGB): The class of gradient boosting algorithms includes sophisticated and extremely effective machine learning techniques like XGBoost. It is frequently employed for both classification and regression problems, and winning machine learning challenges on websites like Kaggle has often relied on it. XGBoost is renowned for its quickness, precision, and capacity for handling large, complicated datasets [37], [38]

- Support Vector Machine (SVM): An effective and flexible machine learning approach called SVM is utilized for both classification and regression problems. When needing to determine a distinct border (hyperplane) between two classes or making predictions using sparse data, SVMs are especially useful [39].

- AdaBoost Classifier (ABC): AdaBoost, which stands for Adaptive Boosting, is an ensemble learning method primarily used for classification tasks. AdaBoost is known for its ability to improve the accuracy of weak learners (classifiers with limited predictive power) by combining them into a strong ensemble model. The algorithm adapts and assigns more weight to data points that are misclassified by previous weak learners [40]–[42].

- Gradient Boosting Classifier (GBC): Gradient Boosting is an effective ensemble learning technique that may be utilized for classification and regression applications. Gradient Boosting Classifier, a subset of this approach, focuses on classification challenges. It creates a powerful ensemble model by pooling the predictions of numerous weak learners (usually decision trees). The term "gradient" refers to the optimization of a cost function by gradient descent [43], [44].

Since many articles on how the machine learning methods used work are given with their formulas and usage logic, in this study only the scanned publications are shown by citing the researched articles. For detailed information, you can refer to the related articles. Since all these methods are used in the study and a multi-class dataset is used, both the class-based performance results and the weighted total performances of all of them are given.

2.3. Evaluation Metrics

Some performance metrics are needed to see how well machine learning techniques are working in the study. In this way, it is possible to make a comparison about which method works better than which method. The method to be used may not always be the highest accuracy value. Different algorithms that are problem and result-oriented can also be selected. The metrics used in the study are as follows:

$$Accuracy = \frac{TP + TN}{TP+FP+TN+FN} \quad (1)$$

$$F1 - score = \frac{2 * Precision * Recall}{Precision + Recall} \quad (2)$$

$$Precision = \frac{TP}{TP+FP} \quad (3)$$

$$Recall = \frac{TP}{TP+FN} \quad (4)$$

Accuracy (1) measures the proportion of correct predictions out of all predictions made by the model. It is a straightforward and intuitive metric, but it may not be the best choice when dealing with imbalanced datasets (datasets where one class is much more frequent than the other). In imbalanced datasets, a high accuracy can be misleading because the model may simply predict the majority class most of the time. The harmonic mean of precision and recall values is expressed by the F1-score (2). When measuring a model's performance, it offers a fair assessment that considers both false positives and false negatives. When trying to balance recall and precision, the F1 Score is especially helpful. When the cost of false positives and false negatives differs, it is a useful metric to employ. Precision (3), also known as Positive Predictive Value, is the fraction of true positive predictions (positive instances properly predicted by the model) out of all positive instances projected by the model. Precision indicates how well your model performs when it predicts a positive class and aids in the reduction of false positives. In simple terms, it measures the accuracy of the model's positive predictions. Recall (4), also known as Sensitivity or True Positive Rate, measures the proportion of true positive predictions out of all actual positive instances in the dataset. Recall tells how well the model captures positive instances, and it helps to minimize false negatives. In other words, it is a measure of how effectively the model can find all the positive instances [45]. To calculate all these parameters, a Confusion Matrix must be developed. A classification model's performance can be evaluated using a crucial tool called a confusion matrix, which is a tabular representation of the model's predictions against the actual outcomes for a given dataset. The accuracy of the

model and its ability to distinguish between different classes are determined by analyzing this matrix. It's a critical tool for evaluating the effectiveness of a classification model and knowing how well it classifies data items. The confusion matrix gives information about the accuracy, precision, recall, and other performance measures of the model. Receiver Operating Characteristic (ROC) curve is a graphical representation used to assess the performance of binary classification models, particularly when determining the trade-off between the true positive rate and the false positive rate at different classification thresholds. ROC curves are commonly used in machine learning and statistics to comprehend a model's discriminative capability and to compare the performance of various classifiers.

3. Results and Discussion

Since many different machine learning methods were used in the study, instead of giving the results of all of them one by one, they were given in combination. The order of giving is random and independent of the height of success. The class labels shown in the study are as follows: not_cyberbullying: 0, gender: 1, religion: 2, age: 3, ethnicity: 4.

Table 1 displays the results of the evaluation metrics for each class and Figure 3 depicts the confusion matrices and ROC curves of nine machine learning methods employed in the study.

Table 1. Evaluation Metrics Results of Methods.

Classifier	Classes	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	Support
Logistic Regression	not-cyberbullying	88,93	78,75	88,93	83,53	2375
	gender	85,15	95,10	85,15	89,85	2371
	religion	94,51	96,28	94,51	95,39	2439
	age	96,08	96,28	96,08	96,18	2399
	ethnicity	97,72	98,09	97,72	97,90	2366
Decision Tree	not-cyberbullying	83,12	80,80	83,12	81,94	2375
	gender	87,05	88,81	87,05	87,92	2371
	religion	94,26	94,53	94,26	94,40	2439
	age	96,62	96,78	96,62	96,70	2399
	ethnicity	97,25	97,67	97,25	97,46	2366
Random Forest	not-cyberbullying	89,77	81,16	89,77	85,25	2375
	gender	84,23	94,55	84,23	89,09	2371
	religion	96,27	95,37	96,27	95,82	2439
	age	97,62	98,07	97,62	97,85	2399
	ethnicity	98,69	98,90	98,69	98,79	2366
Naïve Bayes	not-cyberbullying	88,93	77,42	88,93	82,77	2375
	gender	79,76	93,38	79,76	86,03	2371
	religion	94,38	93,01	94,38	93,69	2439
	age	96,71	96,95	96,71	96,83	2399
	ethnicity	94,72	96,22	94,72	95,46	2366
K-NN	not-cyberbullying	95,16	21,65	95,16	35,28	2375
	gender	16,87	48,37	16,87	25,02	2371
	religion	2,79	87,18	2,79	5,40	2439
	age	7,04	97,13	7,04	13,14	2399
	ethnicity	18,22	99,31	18,22	30,79	2366

XGBoost	not-cyberbullying	90,53	80,19	90,53	85,05	2375
	gender	85,20	94,75	85,20	89,72	2371
	religion	94,96	96,14	94,96	95,54	2439
	age	97,21	98,31	97,21	97,76	2399
	ethnicity	98,65	99,07	98,65	**98,86**	2366
SVM	not-cyberbullying	89,52	78,02	89,52	83,37	2375
	gender	83,55	95,79	83,55	89,25	2371
	religion	94,42	96,24	94,42	95,32	2439
	age	96,79	96,03	96,79	96,41	2399
	ethnicity	97,72	98,55	97,72	98,13	2366
AdaBoost	not-cyberbullying	88,34	72,87	88,34	79,86	2375
	gender	76,13	96,11	76,13	84,96	2371
	religion	92,91	94,89	92,91	93,89	2439
	age	96,96	95,29	96,96	96,12	2399
	ethnicity	97,55	97,63	97,55	97,59	2366
Gradient Boosting	not-cyberbullying	92,21	76,36	92,21	83,54	2375
	gender	82,58	95,89	82,58	88,74	2371
	religion	92,78	96,18	92,78	94,45	2439
	age	96,50	98,97	96,50	97,72	2399
	ethnicity	98,14	98,89	98,14	98,52	2366

There are deviations in the findings of the not_cyberbullying class in the Logistic Regression and Decision Tree classifier results. Other classes have obviously demonstrated significant levels of accomplishment. The confusion matrix and ROC curve of Logistic Regression and Decision Tree is given in Figure 3 (i) and (ii) respectively. Random Forest classifier outperforms all other classifiers in terms of accuracy with 93,34% (Figure 3 (iii)). Naive Bayes shows that while it is expected to give better results in statistical results, it can also give good results in this problem. It makes sense to think that this may have value in terms of statistical approach and can be used as a detail to consider when performing feature extraction. Naive Bayes' results are given in Table 1 and confusion matrix and ROC curve in Figure 3 (iv). Naive Bayes is often used in the solution of statistical data due to its speed and simple use. However, since there is not much statistical data in the study, it is a little behind. One of the reasons for using it in the study is the thought that it will be useful in comparison with other studies since it is a frequently preferred algorithm. Surprisingly, the KNN algorithm produced really poor results (Figure 3 (v)). The results were extremely poor, of a type that might produce almost entirely misleading results. This demonstrates that an algorithm must be employed to solve this problem. XGBoost, which is commonly employed because it produces good results in machine learning competitions, produced the highest and best outcomes of any machine learning algorithm utilized in the study. It was the algorithm with the second highest accuracy, earning a score of 93.32% (Figure 3 (vi)).

The Support Vector Machine algorithm was 92.42% accurate (Figure 3 (vii)). However, in terms of processing time, the model must be able to run for extended periods of time. This time is substantially shorter in other algorithms. As a result, instead of using this model, it would be more accurate to use other machine **learning** techniques. AdaBoost is a versatile and successful

method that is often utilized in real-world applications like face detection and text categorization. The findings of the not-cyberbullying class have lower results than other classes (Figure 3 (viii)). Gradient Boosting is a powerful ensemble method that often outperforms AdaBoost. The confusion matrix and ROC curve of Gradient Boosting is given in Figure 3 (ix).

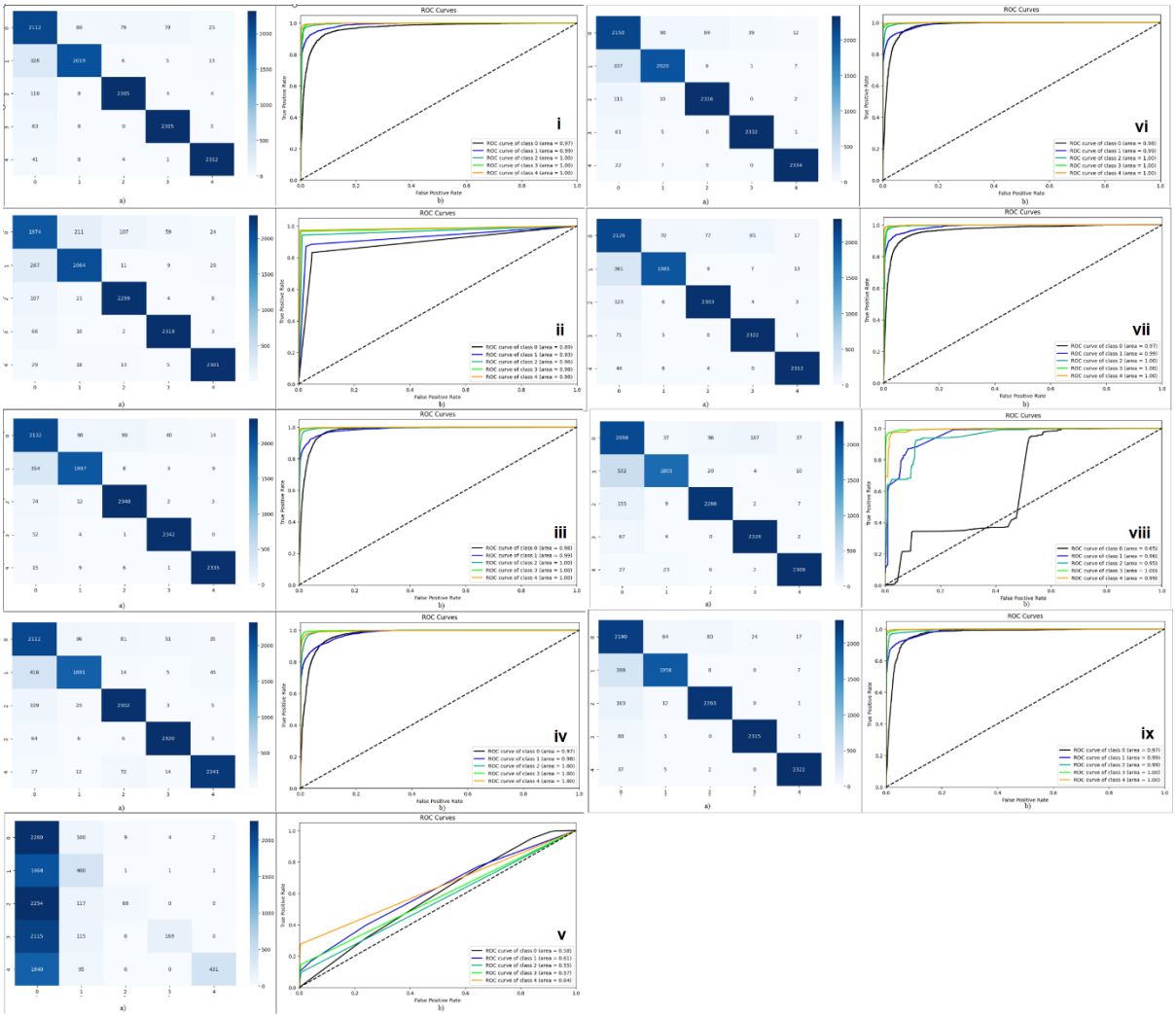

Figure 3. Confusion matrix (a) and ROC curve (b) of algorithms: (i) Logistic Regression, (ii) Decision Tree, (iii) Random Forest, (iv) Naïve Bayes, (v) K-NN, (vi) XGBoost, (vii) SVM, (viii) Adabooost, (ix) Gradient Boosting.

Table 1 previously provided the success metrics of the classifiers for five classes. Table 2 also displays the weighted metric findings for each classifier employed in the study. The authentic accuracy levels of all of the five classes are represented by these weighted metrics in Table 2. When all of the results are compared, the Random Forest and XGBoost algorithms yield the best accuracy. Due of its high success rate, XGBoost has been utilized a lot in recent years. Researchers frequently utilize it because of its capacity to classify large amounts of data quickly and efficiently. Weighted accuracy value was found 93,32% with XGBoost and 93,34% with Random Forest. The highest precision value was found in XGBoost, which is 93,71%. The highest recall value was on Random Forest at 93,34% and followed by XGBoost at 93.32%. The F1-Score metric shows the harmonic mean of precision and recall values. A high F1-Score indicates that both precision and recall are balanced. The XGBoost model has the highest F1-

Score with 93,40%. The method with the lowest results is KNN with 27.85% weighted accuracy.

In terms of F1-Score, XGBoost slightly outperforms Random Forest (93.40% vs. 93.38%). Because of its adaptability, speed, and performance, XGBoost has become a go-to solution for many machine learning practitioners. It is especially well-suited for structured data and is frequently utilized in real-world applications across multiple areas. Random Forest is a versatile and powerful algorithm that is widely employed in a wide range of applications.

Table 2. Weighted metric results of all classifiers.

Classifier	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Logistic Regression	92,49	92,92	92,49	92,59
Decision Tree	91,68	91,74	91,68	91,71
Random Forest	**93,34**	93,62	**93,34**	93,38
Naïve Bayes	90,93	91,41	90,93	90,98
K-NN	27,85	70,85	27,85	21,81
XGBoost	93,32	**93,71**	93,32	**93,40**
SVM	92,42	92,94	92,42	92,52
AdaBoost	90,40	91,38	90,40	90,51
Gradient Boosting	92,45	93,28	92,45	92,61

Many machine learning algorithms have been included in the study and have been examined in many ways. The studies were aimed to find a result such as which one should be preferred for this problem. It sheds light on how subsequent similar problems should be solved with which algorithms. Default values were used intentionally in the study so that it was possible to observe which algorithm was better in a fundamental sense.

4. Conclusion

A large-scale dataset related to cyberbullying was discussed in the study. The dataset consists of five different classes. The distribution of the classes in the dataset is roughly equal. This makes it possible to more accurately assess how the study affected the overall success. There is no missing or unlabeled data in the dataset. As a result, neither additions nor subtraction were required. Feature extraction was done prior to the dataset being inserted into machine learning algorithms. For natural language processing tasks like tokenization and lower case, this procedure is widely utilized. It is a process that simplifies the tweets and ensures that the processes yield better results. It was then fed into machine learning algorithms after the required preparations were finished. Numerous machine learning algorithms were contrasted and discussed among themselves. It was determined that the Random Forest Classifier and XGBoost algorithms had the best success rates.

Random Forest is an ensemble learning model that trains many decision trees independently on subsets of the data. The final predictions are made by averaging or taking the majority of the predictions from individual trees. XGBoost similarly is an ensemble learning model, but it builds trees sequentially and focuses on correcting what previous trees have learned. It operates using the gradient boosting method and iteratively works to optimize the loss function. XGBoost tends to be faster than Random Forest. The sequential learning approach of gradient boosting can make the error reduction process more efficient. However, both models perform well on large datasets and in high-dimensional feature spaces. XGBoost provides more hyperparameter tuning options, allowing for better customization of the model. This provides flexibility but may require more attention. Random Forest generally requires fewer parameter adjustments and is simpler to use. Random Forest often exhibits a tendency towards overfitting, especially when including a large number of trees. XGBoost includes regularization terms and adjustments for tree size, providing better control over overfitting.

It was focused on vectorizing words, utilizing TFIDF to concentrate on the syntactic features of the words. However, future studies could enhance this approach by incorporating various word embedding methods such as word2vec, fastText, and GloVe, which consider the semantic properties of words. Additionally, integrating deep learning networks could provide more robust and comparable results. Similarly, the classification performance of large language models could be evaluated.

The goal of the study is to identify the optimal algorithmic technique for this particular problem, which is why numerous machine learning algorithms are used. This makes it possible to decide which algorithm to concentrate on when carrying out improvement experiments. Default parameters were utilized while applying machine learning methods. Based on weighted metrics, the Random Forest algorithm had the greatest Accuracy of 93,34% and the XGBoost algorithm had the highest F1 score of 93,4%. In general, all algorithms were discussed and the best one was revealed in the study.

In conclusion, both models demonstrate strong performance, but the choice between them depends on the specific application and dataset. While Random Forest may be simpler to use, XGBoost offers more parameter tuning options and often achieves higher performance.

Ethics in Publishing

There are no ethical issues regarding the publication of this study.

Author Contributions

Designing the research, collecting data, evaluating the results, writing articles, etc. transactions were made by Funda Akar.

References

[1] A. Saravanaraj, J. I. Sheeba, and S. P. Devaneyan, "Automatic Detection of Cyberbullying From Twitter," *IRACST-International Journal of Computer Science and Information Technology & Security (IJCSITS)*, vol. 6, no. 6, pp. 2249–9555, 2019, [Online]. Available: https://www.researchgate.net/publication/333320174.

[2] W. N. H. W. Ali, M. Mohd, and F. Fauzi, "Cyberbullying detection: an overview," in *2018 Cyber Resilience Conference (CRC)*, 2018, pp. 1–3.

[3] J.-M. Xu, K.-S. Jun, X. Zhu, and A. Bellmore, "Learning from bullying traces in social media," in *Proceedings of the 2012 conference of the North American chapter of the association for computational linguistics: Human language technologies*, 2012, pp. 656–666.

[4] M. Dadvar, F. M. G. de Jong, R. Ordelman, and D. Trieschnigg, "Improved cyberbullying detection using gender information," in *Proceedings of the Twelfth Dutch-Belgian Information Retrieval Workshop (DIR 2012)*, 2012, pp. 23–25.

[5] D. Jurafsky, *Speech \& language processing*. Pearson Education India, 2000.

[6] T. P. Nagarhalli, V. Vaze, and N. K. Rana, "Impact of machine learning in natural language processing: A review," in *2021 third international conference on intelligent communication technologies and virtual mobile networks (ICICV)*, 2021, pp. 1529–1534.

[7] J. Cheng, C. Danescu-Niculescu-Mizil, and J. Leskovec, "Antisocial behavior in online discussion communities," in *Proceedings of the international aaai conference on web and social media*, 2015, vol. 9, no. 1, pp. 61–70.

[8] Z. Ghasem, I. Frommholz, and C. Maple, "Machine learning solutions for controlling cyberbullying and cyberstalking," *J Inf Secur Res*, vol. 6, no. 2, pp. 55–64, 2015.

[9] S. Murnion, W. J. Buchanan, A. Smales, and G. Russell, "Machine learning and semantic analysis of in-game chat for cyberbullying," *Computers \& Security*, vol. 76, pp. 197–213, 2018.

[10] K. Reynolds, A. Kontostathis, and L. Edwards, "Using machine learning to detect cyberbullying," in *2011 10th International Conference on Machine learning and applications and workshops*, 2011, vol. 2, pp. 241–244.

[11] D. Van Bruwaene, Q. Huang, and D. Inkpen, "A multi-platform dataset for detecting cyberbullying in social media," *Language Resources and Evaluation*, vol. 54, pp. 851–874, 2020.

[12] J. Wang, R. J. Iannotti, and T. R. Nansel, "School bullying among adolescents in the United States: Physical, verbal, relational, and cyber," *Journal of Adolescent health*, vol. 45, no. 4, pp. 368–375, 2009.

[13] V. Balakrishnan, S. Khan, and H. R. Arabnia, "Improving cyberbullying detection using Twitter users' psychological features and machine learning," *Computers \& Security*, vol. 90, p. 101710, 2020.

[14] J. Hani, N. Mohamed, M. Ahmed, Z. Emad, E. Amer, and M. Ammar, "Social media cyberbullying detection using machine learning," *International Journal of Advanced Computer Science and Applications*, vol. 10, no. 5, 2019.

[15] M. O. Raza, M. Memon, S. Bhatti, and R. Bux, "Detecting cyberbullying in social

commentary using supervised machine learning," in *Advances in Information and Communication: Proceedings of the 2020 Future of Information and Communication Conference (FICC), Volume 2*, 2020, pp. 621–630.

[16] M. Sintaha and M. Mostakim, "An empirical study and analysis of the machine learning algorithms used in detecting cyberbullying in social media," in *2018 21st International Conference of Computer and Information Technology (ICCIT)*, 2018, pp. 1–6.

[17] B. R. Chakravarthi, "Hope speech detection in YouTube comments," *Social Network Analysis and Mining*, vol. 12, no. 1, p. 75, 2022.

[18] C. Iwendi, G. Srivastava, S. Khan, and P. K. R. Maddikunta, "Cyberbullying detection solutions based on deep learning architectures," *Multimedia Systems*, vol. 29, no. 3, pp. 1839–1852, 2023, doi: 10.1007/s00530-020-00701-5.

[19] Kaggle, "Cyberbullying Classification." https://www.kaggle.com/datasets/andrewmvd/cyberbullying-classification (accessed Apr. 17, 2023).

[20] J. Wang, K. Fu, and C.-T. Lu, "Sosnet: A graph convolutional network approach to fine-grained cyberbullying detection," in *2020 IEEE International Conference on Big Data (Big Data)*, 2020, pp. 1699–1708.

[21] S. Bird, "NLTK: the natural language toolkit," in *Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions*, 2006, pp. 69–72.

[22] G. Grefenstette, "Tokenization," in *Syntactic wordclass tagging*, Springer, 1999, pp. 117–133.

[23] S. Sperandei, "Understanding logistic regression analysis," *Biochemia medica*, vol. 24, no. 1, pp. 12–18, 2014.

[24] J. Chen *et al.*, "A comparison of linear regression, regularization, and machine learning algorithms to develop Europe-wide spatial models of fine particles and nitrogen dioxide," *Environment international*, vol. 130, p. 104934, 2019.

[25] D. Maulud and A. M. Abdulazeez, "A review on linear regression comprehensive in machine learning," *Journal of Applied Science and Technology Trends*, vol. 1, no. 4, pp. 140–147, 2020.

[26] S. R. Safavian and D. Landgrebe, "A survey of decision tree classifier methodology," *IEEE transactions on systems, man, and cybernetics*, vol. 21, no. 3, pp. 660–674, 1991.

[27] Y. K. Qawqzeh, M. M. Otoom, and F. Al-Fayez, "A Proposed Decision Tree Classifier for Atherosclerosis Prediction and Classification," *International Journal of Computer Science and Network Security (IJCSNS)*, vol. 19, no. 12, pp. 197–202, 2019.

[28] L. Breiman, J. Friedman, C. Stone, and R. Olshen, "Classification and regression trees (crc, boca raton, fl)," 1984.

[29] L. Breiman, "Random forests; uc berkeley tr567," *University of California: Berkeley, CA, USA*, 1999.

[30] L. Breiman, "Random Forests for Scientific Discovery," *Presentation*, pp. 1–167, 2013, [Online]. Available: http://www.math.usu.edu/adele/RandomForests/ENAR.pdf.

[31] J. R. Quinlan, *C4. 5: programs for machine learning*. Elsevier, 2014.

[32] I. Rish and others, "An empirical study of the naive Bayes classifier," in *IJCAI 2001 workshop on empirical methods in artificial intelligence*, 2001, vol. 3, no. 22, pp. 41–46.

[33] E. Frank and R. R. Bouckaert, "Naive bayes for text classification with unbalanced classes," in *Knowledge Discovery in Databases: PKDD 2006: 10th European Conference on Principles and Practice of Knowledge Discovery in Databases Berlin, Germany, September 18-22, 2006 Proceedings 10*, 2006, pp. 503–510.

[34] Ö. Şahinaslan, H. Dalyan, and E. Şahinaslan, "Naive bayes s{\i}n{\i}fland{\i}r{\i}c{\i}s{\i} kullan{\i}larak youtube verileri üzerinden çok dilli duygu analizi," *Bili{\c{s}}im Teknolojileri Dergisi*, vol. 15, no. 2, pp. 221–229, 2022.

[35] Y. Wu, K. Ianakiev, and V. Govindaraju, "Improved k-nearest neighbor classification," *Pattern recognition*, vol. 35, no. 10, pp. 2311–2318, 2002.

[36] G. Guo, H. Wang, D. Bell, Y. Bi, and K. Greer, "KNN Model-Based Approach in Classification," in *On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE*, 2003, pp. 986–996.

[37] T. Chen *et al.*, "Xgboost: extreme gradient boosting," *R package version 0.4-2*, vol. 1, no. 4, pp. 1–4, 2015.

[38] T. Chen, T. He, M. Benesty, and V. Khotilovich, "Package 'xgboost,'" *R version*, vol. 90, pp. 1–66, 2019.

[39] W. S. Noble, "What is a support vector machine?," *Nature biotechnology*, vol. 24, no. 12, pp. 1565–1567, 2006.

[40] A. Vezhnevets and V. Vezhnevets, "Modest AdaBoost-teaching AdaBoost to generalize better," in *Graphicon*, 2005, vol. 12, no. 5, pp. 987–997.

[41] X. Li, L. Wang, and E. Sung, "AdaBoost with SVM-based component classifiers," *Engineering Applications of Artificial Intelligence*, vol. 21, no. 5, pp. 785–795, 2008.

[42] T.-K. An and M.-H. Kim, "A new diverse AdaBoost classifier," in *2010 International conference on artificial intelligence and computational intelligence*, 2010, vol. 1, pp. 359–363.

[43] J. Son, I. Jung, K. Park, and B. Han, "Tracking-by-segmentation with online gradient boosting decision tree," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 3056–3064.

[44] S. Peter, F. Diego, F. A. Hamprecht, and B. Nadler, "Cost efficient gradient boosting," *Advances in neural information processing systems*, vol. 30, 2017.

[45] M. Hossin and M. N. Sulaiman, "A review on evaluation metrics for data classification evaluations," *International journal of data mining \& knowledge management process*, vol. 5, no. 2, p. 1, 2015.