March 2024, Vol:2, Issue:1

International Journal of New Findings in Engineering, Science and Technology

journal homepage: *https://ijonfest.gedik.edu.tr/*

Using Artificial Intelligence Algorithms to Detect Hate Speech in Social Media Posts

Aytaç Uğur Yerden[a], Kadir Turgut[b]

[a]Department of Industrial Engineering, Istanbul Gedik University, Istanbul, TURKIYE e-mail: aytac.yerden@gedik.edu.tr
[b]Department of Computing, Istanbul Gedik University, Istanbul, TURKIYE e-mail: kadiring@hotmail.com

Abstract

Detecting hate speech on social media is of great importance to prevent negative impacts on people and communities and to remove such content. However, detecting hate speech is a complex and challenging process due to linguistic and cultural diversity. Therefore, it is important to develop powerful and effective machine learning algorithms. Since detecting such content using traditional methods can be time-consuming and costly, it is stated that artificial intelligence-based machine learning algorithms have great potential in this regard. The aim of this study is to evaluate the performance of artificial intelligence-based machine learning algorithms used in detecting posts containing hate speech on social media. The study focuses on the problem of detecting and managing hate speech on social media platforms. In this study, we will compare the performances of different algorithms and determine the most suitable methods. Additionally, the effects of the dataset and feature extraction methods on algorithm performance will be analyzed. Algorithms are often based on natural language processing techniques and try to detect hate speech by learning features in texts. The performance of these algorithms can vary depending on factors such as language, culture, the attributes they use, and the training dataset, so a comprehensive analysis is required. In the research, the performance of the algorithms used in detecting hate speech was compared with the dataset and feature extraction methods. In this process, the algorithms' linguistic and cross-cultural effectiveness, feature selection and representation, false positive and false negative rates, and overall accuracy will be analyzed.

Keywords: Social Media; Hate Speech; Artificial Intelligence.

1. INTRODUCTION

Social media offers people great freedom to share their thoughts, ideas and feelings. These platforms provide social benefits by facilitating communication and information exchange between people [1]. However, with the widespread use of social media, negative effects such as hate speech have also emerged. Hate speech is expressions of intolerance and hostility towards a specific group, community or individuals [2]. Such expressions can lead to discrimination, violence and social tension in various societies and countries [3].

Detection and management of posts containing hate speech has become an important problem for social media platforms [4]. With traditional methods (e.g., manual review by moderators), detecting and managing such posts can

https://doi.org/10.61150/ijonfest.2024020102

be time-consuming and costly [5]. Therefore, AI-based machine learning algorithms have great potential to automatically detect and manage hate speech on social media [6].

In this study, the performance evaluation of artificial intelligence-based machine learning algorithms used in detecting posts containing hate speech on social media is discussed. The performance of various algorithms will be compared with the dataset and feature extraction methods used for hate speech detection [7].

As a result, this study, which focuses on the performance evaluation of artificial intelligence-based machine learning algorithms in detecting posts containing hate speech on social media, can be seen as an important step to prevent the spread of hate speech and make social media environments safer.

2. MATERIAL AND METHOD

This study on the performance evaluation of artificial intelligence-based machine learning algorithms in detecting posts containing hate speech on social media aims to use various machine learning and deep learning methods to automatically detect hate speech. The flow diagram of the method used is shown in Figure 1. This section provides a detailed description of the methods and techniques used.

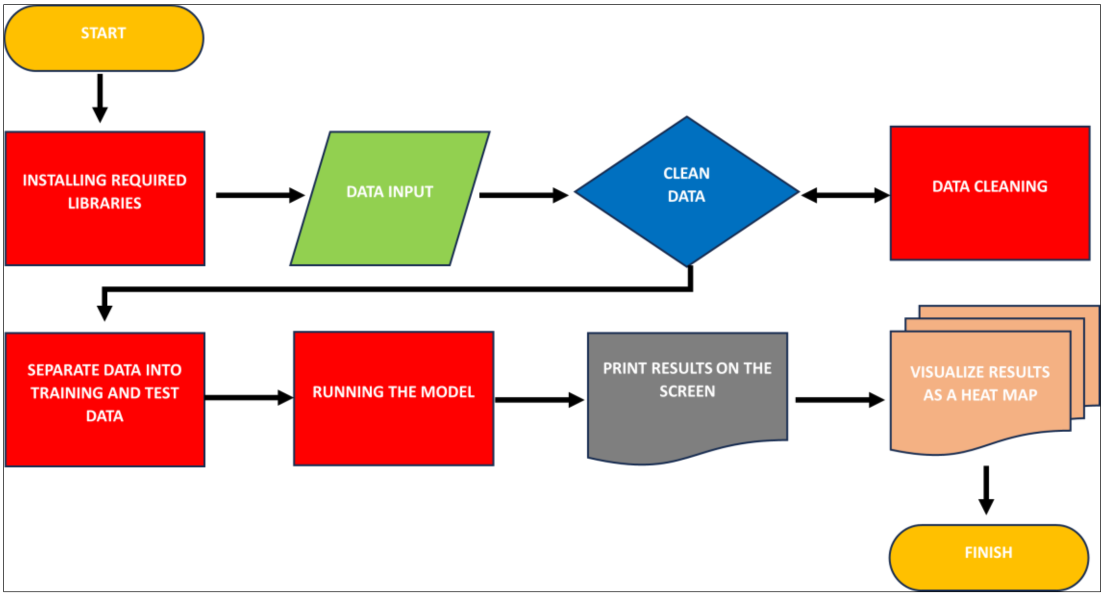

START
INSTALLING REQUIRED LIBRARIES
DATA INPUT
CLEAN DATA
DATA CLEANING
SEPARATE DATA INTO TRAINING AND TEST DATA
RUNNING THE MODEL
PRINT RESULTS ON THE SCREEN
VISUALIZE RESULTS AS A HEAT MAP
FINISH

Figure 1. Flow Diagram.

2.1 Data collection and preprocessing

The dataset that forms the basis of this study consists of social media posts that contain and do not contain hate speech. The dataset is the Twitter hate speech dataset obtained from the kaggle.com platform. When determining the dataset, accounts and pages that specifically focused on topics related to hate speech were examined. In the pre-processing step, texts were cleaned to reduce noise and prepared for feature extraction.

2.2 Feature extraction

To detect hate speech using machine learning and deep learning methods, features need to be extracted from texts.

2.3 Classification models

Various machine learning and deep learning algorithms have been used to classify posts that contain and do not contain hate speech. These algorithms are Support Vector Machines, Multi-Layer Perceptron, Decision Trees, Multinomial Naive Bayes, K-Nearest Neighbor, Gradient Boosting, Logistic Regression and Random Forest models.

2.4 Model training and performance evaluation

The dataset is divided into training, validation, and testing subsets for model training and performance evaluation [8]. Classification models were trained on the training dataset and hyperparameter tuning was performed on the validation dataset [9]. To evaluate model performance, various metrics were used on the test dataset. These metrics include AUC-ROC Score, F1-score, Recall, Precision and Accuracy the value under the area curve [10].

2.5 Algorithms

In this section, Support Vector Machines, Multi-Layer Perceptron, Decision Trees, Multinomial Naive Bayes, K-Nearest Neighbor, Gradient Boosting, Logistic Regression and Random Forest algorithms are briefly described.

3. FINDINGS AND DISCUSSION

In this section, first of all, 8 different algorithms were run one by one and it was aimed to reach a successful result.

A high Accuracy, Precision, Recall, F1 Score and AUC-ROC Score indicates that the model is performing well. Low Precision indicates that the number of false positive predictions is high, while low Recall indicates that the number of false negative predictions is high. Ideally, the aim is to achieve a high F1 Score with high Precision and Recall.

Accuracy: Shows the correct rate of all predictions.
Precision: Shows how many of the samples predicted as positive are actually positive.
Recall: Shows how many of the true positive samples were predicted as positive.
F1 Score: Takes the harmonic average of the Precision and Recall value and provides a balanced performance measure.
AUC-ROC Score: Indicates the probability of the classifier correctly sorting a random positive and negative sample.

According to all these results, the performance of the model in detecting tweets containing hate speech can be evaluated. High accuracy indicates the proportion of samples classified correctly. However, Precision, Recall and F1 Score values should also be checked for balance. In particular, attention should be paid to the Precision value to reduce the false alarm rate (false positive) in detecting hate speech. Recall indicates how many true positives were classified correctly, while F1 Score provides a balance between Precision and Recall. AUC-ROC Score indicates the model's ability to distinguish classes; A value close to 1 indicates perfect discrimination.

3.1 Decision Trees (DT) Performance Evaluation

The metric values shown in Table 1 are the results of the performance test conducted with the Decision Trees (DT) algorithm on the twitter-hate-speech dataset.

Table 1. Decision Trees (DT) Metric Values.

Metric	**Value**
AUC-ROC Score	0.7534795173887220
F1 Score	0.5628604382929643
Recall	0.5350877192982456
Precision	0.5936739659367397
Accuracy	0.9407164085718754

3.2 Gradient Boosting Performance Evaluation

The metric values shown in Table 2 are the results of the performance test conducted with the gradient boosting algorithm on the twitter-hate-speech dataset.

Table 2. Gradient Boosting Metric Values.

Metric	**Value**
AUC-ROC Score	0.63908761291809610
F1 Score	0.42737896494156924
Recall	0.28070175438596490
Precision	0.89510489510489510
Accuracy	0.94634756765211950

3.3 K-Nearest Neighbour (KNN) Performance Evaluation

The metric values shown in Table 3 are the results of the performance test conducted with the K-Nearest Neighbor (KNN) algorithm on the twitter-hate-speech dataset.

Table 3. K-Nearest Neighbor (KNN) Metric Values.

Metric	**Value**
AUC-ROC Score	0.59716663120661680
F1 Score	0.32363636363636360
Recall	0.19517543859649122
Precision	0.94680851063829790
Accuracy	0.94181135617081180

3.4 Logistic Regression (LR) Performance Evaluation

The metric values shown in Table 4 are the results of the performance test conducted with the Logistic Regression (LR) algorithm on the twitter-hate-speech dataset.

Table 4. Logistic Regression (LR) Metric Values.

Metric	**Value**
AUC-ROC Score	0.6665841112381763

F1 Score	0.4911717495987159
Recall	0.3355263157894737
Precision	0.9161676646706587
Accuracy	0.9504145158767402

3.5 Multi-Layered Perceptron (MLP) Performance Evaluation

The metric values shown in Table 5 are the results of the performance test conducted with the Multi-Layered Perceptron (MLP) algorithm on the twitter-hate-speech dataset.

Table 5. Multi-Layered Perceptron (MLP) Metric Values.

Metric	**Value**
AUC-ROC Score	0.7899180429598504
F1 Score	0.6103183315038420
Recall	0.6096491228070176
Precision	0.6109890109890110
Accuracy	0.9444705146253715

3.6 Multinomial Naive Bayes (MNB) Performance Evaluation

The metric values shown in Table 6 are the results of the performance test conducted with the Multinomial Naive Bayes (MNB) algorithm on the twitter-hate-speech dataset.

Table 6. Multinomial Naive Bayes (MNB) Metric Values.

Metric	**Value**
AUC-ROC Score	0.66793325532122380
F1 Score	0.49597423510466987
Recall	0.33771929824561403
Precision	0.93333333333333330
Accuracy	0.95104020021898950

3.7 Random Forest (RF) Performance Evaluation

The metric values shown in Table 7 are the results of the performance test conducted with the Random Forest (RF) algorithm on the twitter-hate-speech dataset.

Table 7. Random Forest (RF) Metric Values.

Metric	**Value**
AUC-ROC Score	0.7381120182973857
F1 Score	0.6171107994389902
Recall	0.4824561403508772
Precision	0.8560311284046692
Accuracy	0.9572970436414828

3.8 Support Vector Machines (SVM) Performance Evaluation

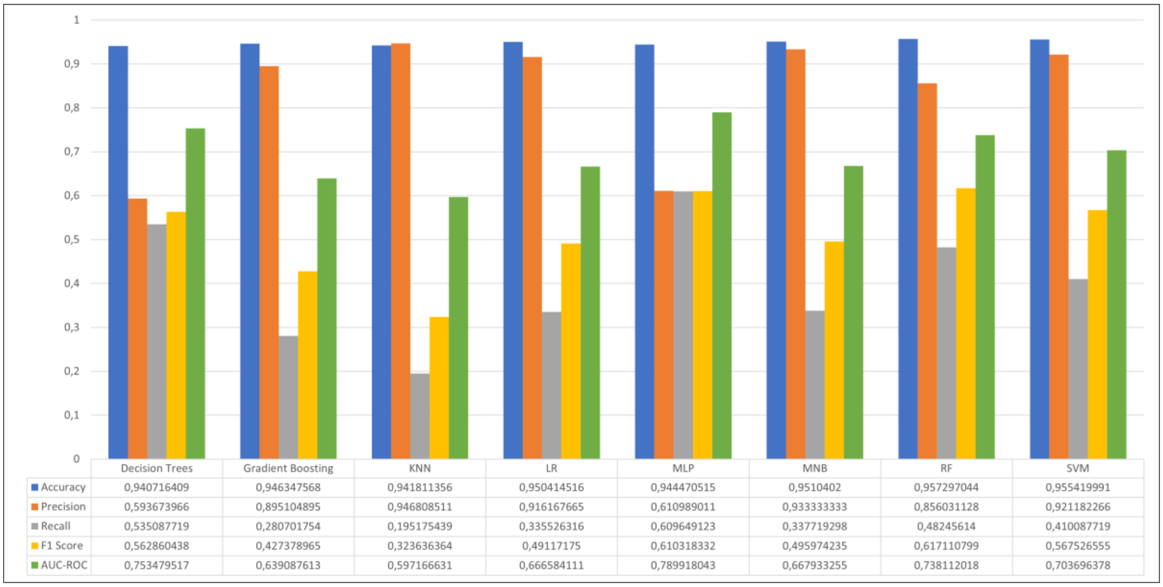
The metric values shown in Table 8 are the results of the performance test conducted with the Support Vector Machines (SVM) algorithm on the twitter-hate-speech dataset.

Table 8. Support Vector Machines (SVM) Metric Values.

Metric	Value
AUC-ROC Score	0.7036963777559109
F1 Score	0.5675265553869500
Recall	0.4100877192982456
Precision	0.9211822660098522
Accuracy	0.9554199906147348

3.9 Support Vector Machines (SVM) Performance Evaluation

Metric values of all algorithms used in performance evaluation tests are shown in Figure 2.



	Decision Trees	Gradient Boosting	KNN	LR	MLP	MNB	RF	SVM
■ Accuracy	0,940716409	0,946347568	0,941811356	0,950414516	0,944470515	0,9510402	0,957297044	0,955419991
■ Precision	0,593673966	0,895104895	0,946808511	0,916167665	0,610989011	0,933333333	0,856031128	0,921182266
■ Recall	0,535087719	0,280701754	0,195175439	0,335526316	0,609649123	0,337719298	0,48245614	0,410087719
■ F1 Score	0,562860438	0,427378965	0,323636364	0,49117175	0,610318332	0,495974235	0,617110799	0,567526555
■ AUC-ROC	0,753479517	0,639087613	0,597166631	0,666584111	0,789918043	0,667933255	0,738112018	0,703696378

Figure 2. Metric Values of All Algorithms.

Accuracy values of all algorithms used in performance evaluation tests are shown in Figure 3.

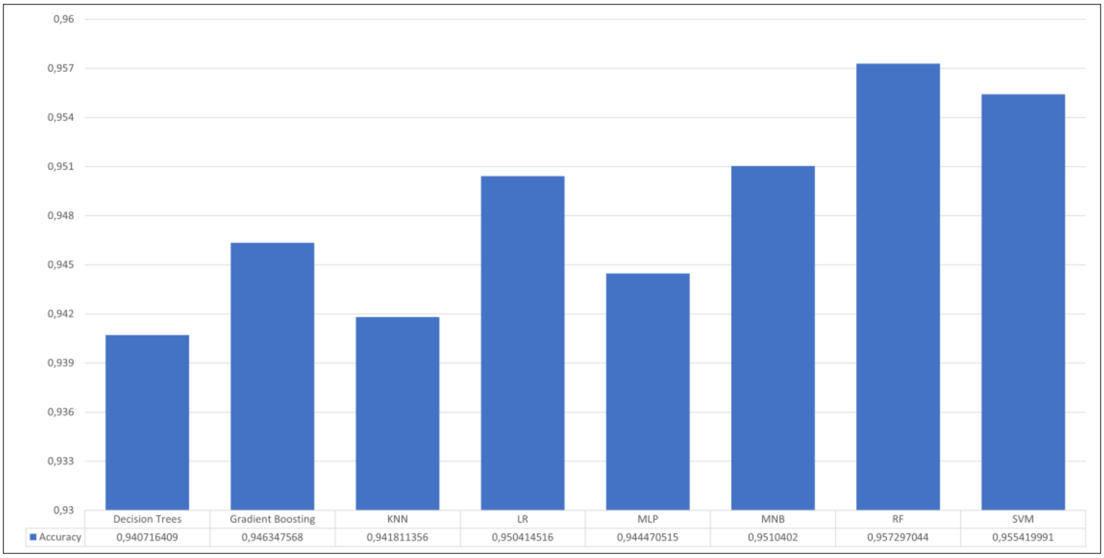


Figure 3. Accuracy Values of All Algorithms.

While Random Forest (RF) has the highest accuracy value, the lowest accuracy value belongs to the Decision Trees (DT) algorithm.

4. CONCLUSION

This study focused on evaluating the performance of various machine learning algorithms in classifying tweets containing hate speech on Twitter. These algorithms include Support Vector Machines, Multi-Layer Perceptron, Decision Trees, Multinomial Naive Bayes, K-Nearest Neighbor, Gradient Boosting, Logistic Regression and Random Forest. For each of these algorithms, their performance was evaluated according to AUC-ROC Score, F1 Score, Recall, Precision and Accuracy metrics. For each algorithm, pre-processing, model definition, training, evaluation and interpretation of the results stages were carried out.

The findings show that, overall, these algorithms have fairly high Accuracy rates. This shows that most of the models' predictions are in line with the actual values and indicates that the models are generally successful. On the other hand, it has been observed that Recall and F1 scores are generally lower, indicating that the models have some difficulties in detecting true positive cases.

Precision metrics show that the models retain a very low rate of false positives (false alarms), while low Recall values indicate that a significant portion of true positive cases are missed. This indicates that models need to be improved, especially in terms of detecting positive cases. F1 scores show that the models deliver balanced but not perfect performance between Precision and Recall.

Although AUC-ROC Score values show that the classification performances of the models are above average, they indicate that no model exhibits perfect performance. This suggests that models may be more prone to certain types of errors and therefore should be used with caution in certain scenarios.

To briefly summarize the performance of each algorithm:

Decision Trees (DT): It made successful predictions with high Accuracy, but remained weak in certain classifications with low Precision and Recall values.

Gradient Boosting: It attracted attention with its high Accuracy and Precision, but its low Recall value caused some true positive cases to be missed.

K-Nearest Neighbor (KNN): Showed high Accuracy and Precision, but missed most of the true positive cases with very low Recall value.

Logistic Regression (LR): Made reliable predictions with high Accuracy and good Precision, but missed some positive cases with low Recall rate.

Multilayer Detectors (MLP): Showed high Accuracy, but Precision, Recall and F1 Score values remained lower.

Multinomial Naive Bayes (MNB): It made successful predictions with very high Accuracy and Precision, but could not detect some positive situations with its low Recall value.

Random Forest (RF): Showed good overall performance with highest Accuracy and good Precision, but missed some positive cases with low Recall rate.

Support Vector Machines (SVM): Showed the second highest Accuracy and Precision, but missed some positive cases with a low Recall rate.

In conclusion, this study shows that various machine learning algorithms can be used effectively to detect content containing hate speech on Twitter. However, it is also clear that each algorithm has its own advantages and limitations and must therefore be carefully selected and implemented. In order for algorithms to be used more effectively and responsibly, they need to be constantly improved and implemented in accordance with ethical standards. Further improving these algorithms and testing them on different data sets may produce more effective results in detecting and blocking content containing hate speech. Therefore, it was concluded that algorithms should be combined or improved to develop a more balanced and effective model. These findings may provide guidance for the development of more powerful and balanced models that can detect hate speech on Twitter.

Although the algorithms examined in this study make an important step forward in the field of hate speech detection, continuous development and careful thought are required in this field. Detection of hate speech is not only a technological issue, but also a social, cultural and ethical responsibility.

Future Studies: This study lays an important foundation for future studies in this field and reveals the potential of machine learning applications in managing social media content. Future studies could focus on datasets in different languages, the changing nature of social media, and the evolution of hate speech.

Acknowledgements

This research is a part of the master's thesis prepared by the researcher at Istanbul Gedik University.

Funding

There are no financial interests in this study.

Declaration of Competing Interest

There is no conflict of interest in this study.

References

[1] Kaplan, A.M. and Haenlein, M. (2010). Users of the world, unite! The challenges and opportunities of social media. Business Horizons, 53(1), 59-68.
[2] Allport, G. W. (1954). The nature of prejudice. Cambridge, MA: Addison-Wesley.
[3] Perry, B. (2001). In the name of hate: Understanding hate crimes. New York: Routledge.
[4] Chetty, N., and Alathur, S. (2018). Hate speech review in the context of online social networks. Aggression and Violent Behavior, 40, 108-118.
[5] Davidson, T., Warmsley, D., Macy, M., and Weber, I. (2017). Automated hate speech detection and the problem of offensive language. In Proceedings of the 11th International Conference on Web and social media, ICWSM 2017, pp. 512-515.

[6] Schmidt, A., and Wiegand, M. (2017). A survey on hate speech detection using natural language processing. In Proceedings of the Fifth International Workshop on Natural Language Processing for social media (pp. 1-10).
[7] Fortuna, P., and Nunes, S. (2018). A survey on automatic detection of hate speech in text. ACM Computing Surveys (CSUR), 51(4), 1-30.
[8] Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. In Proceedings of the 14th International Joint Conference on Artificial Intelligence (pp. 1137-1143). Morgan Kaufmann.
[9] Bergstra, J., and Bengio, Y. (2012). Random search for hyper-parameter optimization. Journal of machine learning research, 13(Feb), 281-305.
[10] Sokolova, M., and Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. Information Processing and Management, 45(4), 427-437.