Cyberbullying Detection Through Sentimental Analysis and NLP Techniques

R. S. Latha[1], R. Rajadevi[1], K. Logeswaran[1], G. R. Sreekanth[2], S. Arun Kumar[1] and D. Praveen[1]
[1]Department of AI, Kongu Engineering College, Perundurai, Erode and 638060, India
[2]Department of IT, Nandha Engineering College, Perundurai, Erode and 638060, India

Keywords: Natural Language Processing (NLP), Tamil Language, Emotion Classification, Logistic Regression, Random Forest Classifier, K-Nearest Neighbor (KNN), Gradient Boost Classifier, Decision Tree Classifier, Naïve Bayes Classifier.

Abstract: Cyberbullying has become a major issue with the rise of social media, impacting mental health, especially among adolescents. Traditional content moderation methods struggle to manage the vast volume of posts generated daily, creating a need for efficient automated detection systems. This project explores advanced NLP and sentiment analysis techniques to detect harmful content in real-time. Using machine learning models like Random Forest , Logistic Regression , Decision Tree , Gradient Boost , Voting classifier , Naive Bayes ,the aim is to identify cyberbullying effectively. Preliminary findings show Random forest outperforms other models in accuracy and reliability. Future work will focus on improving precision, expanding datasets, and enabling real- time detection to foster safer online environment.

1 INTRODUCTION

Cyberbullying is a widespread and escalating problem that poses significant psychological and emotional threats, particularly to young people using social media. Traditional methods for detecting such behavior rely on human moderation, which is not only labor-intensive but also inconsistent due to the sheer volume of data generated daily on these platforms. The importance of quick and accurate detection has grown, as online harassment can have long-lasting effects. The challenge lies in the complex language used in cyberbullying, which often includes sarcasm, slang, and veiled threats, requiring more advanced technological solutions to improve 99detection and enhance online safety. Natural Language Processing (NLP) has become a valuable tool in text analysis, offering the ability to grasp nuances in human language and provide automated insights. Sentiment analysis, a crucial NLP technique, allows systems to evaluate the emotional tone of text, making it particularly effective in detecting harmful or abusive content in online interactions. These technologies, when applied to cyberbullying detection, have the potential to surpass traditional methods by identifying patterns of abuse across large data sets in real time. This project investigates the advanced NLP techniques and sentiment analysis for the detection of cyberbullying on social media. The research centers on analyzing user- generated content, assessing the performance of sentiment-based models in identifying harmful language, and detecting potential bullying behavior. By utilizing a robust dataset of social media posts, the study aims to show how automated systems can assist in the early detection of cyberbullying, reducing its harmful effects and fostering a safer online environment.

2 LITERATURE SURVEY

By evaluating multiple machine learning models, including Naive Bayes and Bi- LSTM. For class-wise performance, Naive Bayes demonstrated varying results across different attributes. For instance, it achieved an F1-score of 0.91 for Religion and Ethnicity, while performing less effectively on the Not bullying class, with F1-score of 0.60. In contrast, the Bi- LSTM model consistently outperformed Naive Bayes, particularly for Age and Ethnicity, achieving an F1-score of 0.97 and 0.98, respectively.

Latha, R. S., Rajadevi, R., Logeswaran, K., Sreekanth, G. R., Arun Kumar, S. and Praveen, D.
Cyberbullying Detection Through Sentimental Analysis and NLP Techniques.
DOI: 10.5220/0013583100004664

In *Proceedings of the 3rd International Conference on Futuristic Technology (INCOFT 2025) - Volume 1*, pages 638-643
ISBN: 978-989-758-763-4

The study highlights the superiority of Bi-LSTM, especially in addressing imbalanced classes, as reflected in its improved precision, recall, and overall F1-scores. These results support the transition from traditional models like Naive Bayes to deep learning models such as Bi-LSTM for more robust performance in cyberbullying detection tasks(Adeyinka Orelaja , 2024).

Explored a novel method for cyberbullying detection by combining Support Vector Machine (SVM) techniques with NLP methodologies. The hybrid approach aims to enhance the detection of cyberbullying by leveraging the strengths of both SVM, known for its robustness in classification tasks, and NLP, which excels in understanding textual data. The study presents findings at ADICS 2024, indicating a significant advancement in the application of machine learning techniques for social media monitoring and online safety(J.Sathya , 2024).

The Naive Bayes and Bi-LSTM models performance for attribute- specific cyberbullying detection. For the Naive Bayes model, the F1-scores varied across different attributes, with Religion achieving an F1- score of 0.91 and Not bullying performing less effectively with an F1-score of 0.60. The Bi- LSTM model, however, showed superior performance. For instance, it recorded an F1-score of 0.97 for Age and 0.98 for Ethnicity. This comparative analysis highlights the strength of Bi-LSTM in handling imbalanced classes, improving both precision and recall across key attributes in the dataset.The study, published in the Journal of Electronic & Information Systems on February 28, 2024, underscores the shift from traditional classifiers like Naïve Bayes to advanced deep learning models like Bi-LSTM for improved accuracy in detecting cyberbullying behaviors across multiple dimensions(Adeyinka Orelaja , 2024)

By analyzing the performance of several classifiers on a custom-built dataset designed to capture various aspects of cyberbullying. The Multinomial Naive Bayes (Multinomial-NB) classifier achieved an F1- score of 0.82, demonstrating balanced precision (0.83) and recall (0.80). Support Vector Machine (SVM) performed similarly with an F1-score of 0.83. Logistic Regression also showed competitive results, scoring 0.79 in F1, while Stochastic Gradient Descent (SGD) had a higher precision 0.91 but lower recall at 0.58, yielding an F1-score of 0.71. The Shallow Neural Network (Shallow NN) recorded an F1- score of 0.77, highlighting moderate performance.The study underscores the variability in performance across classifiers when addressing complex factors such as aggressive language and intent to harm, indicating the potential for hybrid or ensemble methods to improve detection accuracy (Naveed Ejaz,2024).

Reviewed various methods for hate speech detection across multiple social media platforms. The study highlighted several machine learning models and techniques applied to datasets from platforms such as YouTube, Twitter, and MySpace. For instance, Chen et al. used an unsupervised lexical and syntactic match rule-based approach on YouTube data, achieving a high precision of 0.98 and recall of 0.94. Xiang et al. applied semi-supervised logistic regression using topic modeling on Twitter, reporting an F1-score of 0.84. A hybrid CNN model trained on character and Word2vec embeddings by Park and Fung achieved an F1-score of 0.73.Additionally, Wiegand et al. used SVM with lexical and linguistic features along with word embeddings, reaching an F1-score of 0.81 on datasets from Twitter, Wikipedia, and UseNet.The study emphasizes the effectiveness of both supervised and unsupervised approaches, as well as the potential of word embeddings and deep learning methods for improving the accuracy of hate speech detection in multilingual and multi-platform environments(Areej Al-Hassan,2019).

3 PROPOSED SYSTEM ARCHITECTURE

The proposed system aims to create a robust model that can accurately identify the sentiment of text data using advanced Natural Language Processing (NLP) techniques. The primary objective is to develop a classifier model trained on textual data, which can then be used as a pre-trained model to predict sentiment in new and unseen text inputs. To achieve this, the first step involves preparing and collecting a diverse text dataset and assigning each entry the corresponding sentiment label (positive, negative, or neutral). The dataset may consist of multiple languages, including Tamil, to reflect diverse linguistic use cases. After data collection, preprocessing steps like tokenization, stemming or lemmatization, and stopword removal are applied to optimize the text for model training. Additionally, handling missing values and addressing any class imbalances are crucial parts of the process. For missing text values, strategies like filling in with the most common word or using the median can be implemented.

Table 1: Feature description of dataset

Text	Different types of sentences
Label	Positive, negative , Unknown State and MixedFeelings

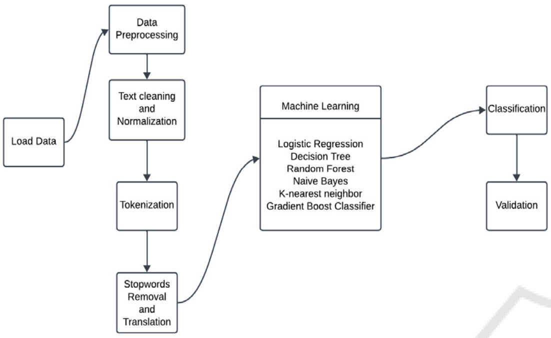


Figure 1: Proposed workflow of machine Learning Model

The next phase involves constructing and preprocessing the text dataset, followed by utilizing traditional machine learning models such as Decision Trees, Random Forest, K-Nearest Neighbors, Logistic Regression, Naive Bayes, and Voting Classifiers for sentiment analysis. Although these models are less complex than deep learning methods, they can still provide valuable insights, particularly with smaller datasets or when interpretability is a priority. Decision Trees and Random Forests work by dividing the data into subsets based on feature importance, while K-Nearest Neighbors classifies data by comparing it to similar labeled examples. Logistic Regression and Naive Bayes are popular for binary or multi-class classification, with Logistic Regression offering a probabilistic framework and Naive Bayes excelling when the assumption of feature independence is valid. A Voting Classifier combines the outputs of various models to enhance overall accuracy. Once these models are trained on the preprocessed text data, the best-performing model is chosen for predicting sentiment in new text inputs. This method strikes a balance between simplicity and effectiveness, making it well-suited for structured text data.

4 MODULES DESCRIPTION

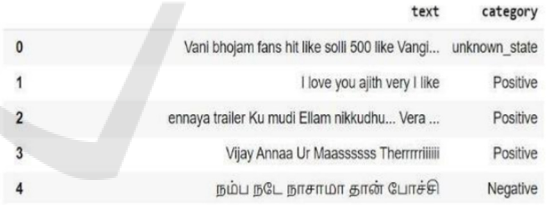
4.1 Data Collection

The dataset used for this project was collected from YouTube comments, which served as a rich source of diverse textual data, including instances of cyberbullying. YouTube, being a popular social media platform, offers a variety of user interactions, making it an ideal source for this study. The comments collected were primarily in Tamil or Tanglish (Tamil written in English script), providing real-world examples of online communication in a bilingual context.

4.2 Data Preprocessing

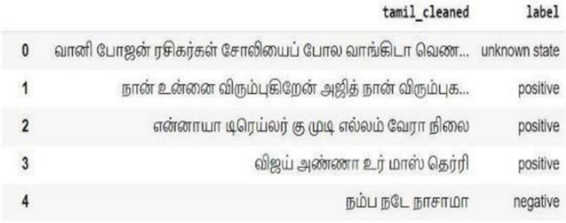
Preprocessing of the dataset involved multiple steps aimed at cleaning and preparing the data for further analysis. The following key procedures were followed

The raw text data included a mix of numbers, special characters, and emojis, which needed to be removed to ensure consistency. Additionally, repetitive characters and unnecessary white spaces were cleaned, and the text was normalized by converting it to lowercase. This pre-processing step ensured that only relevant and clean data was used for model training, eliminating any noise from the dataset.



	text	category
0	Vani bhojam fans hit like solli 500 like Vangi...	unknown_state
1	I love you ajith very I like	Positive
2	ennaya trailer Ku mudi Ellam nikkudhu... Vera ...	Positive
3	Vijay Annaa Ur Maassssss Therrrriiiiii	Positive
4	நம்ப நடே நாசாமா தான் போச்சி	Negative

Figure 2: Dataset before preprocessing



	tamil_cleaned	label
0	வானி போஜன் ரசிகர்கள் சோலியைப் போல வாங்கிடா வெண...	unknown state
1	நான் உன்னை விரும்புகிறேன் அஜித் நான் விரும்புக...	positive
2	என்னாயா டிரெய்லர் கு முடி எல்லம் வேரா நிலை	positive
3	விஜய் அண்ணா உர் மாஸ் தெர்ரி	positive
4	நம்ப நடே நாசாமா	negative

Figure 3: Dataset after preprocessing

Since the dataset contained text written in Tanglish (Tamil words using the English script), it was essential to translate this into the proper Tamil script. A translation tool was used to convert the

Tanglish text into Tamil. This translation step was done in batches to manage the size of the dataset and maintain efficiency. After translation, the dataset was stored for further use. 11 Once the Tanglish text was translated into Tamil, the various parts of the dataset were merged to form a single, unified dataset. This step ensured that the translated text was consolidated, providing a consistent format for model training. To ensure that the text data used for the model was entirely in the Tamil script, any non-Tamil characters were removed. This step guaranteed that the dataset contained only relevant characters, further improving the quality of the data and making it suitable for the machine learning models used in this project.

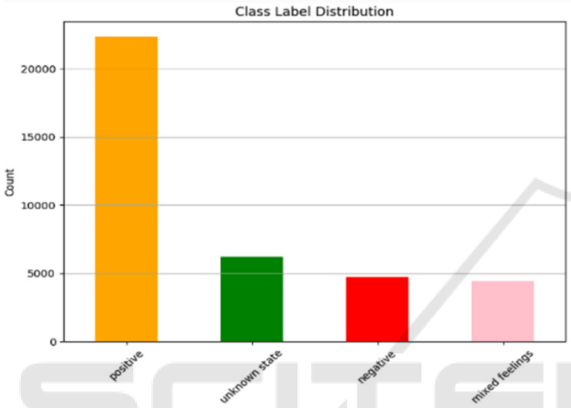


Figure 4: Frequency of classes in the dataset

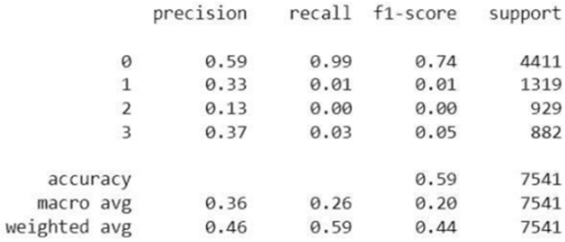
4.3 Machine Learning models

Cyberbullying detection through machine learning employs various classification techniques to identify harmful or abusive online content. Techniques like 12 Decision Trees, Random Forests, Gaussian Naive Bayes, Logistic Regression, K-Nearest Neighbors, and Gradient Boosting are key in this process. Each algorithm uses unique methods to learn patterns from text data, enabling the classification of content as either cyberbullying or non-cyberbullying. By training on labeled datasets, these models can recognize abusive language by detecting specific linguistic features and context. Their performance is evaluated using metrics such as accuracy, precision, recall, and F1 score, helping to ensure they effectively automate moderation and improve online safety.

4.3.1 Logistic Regression

Logistic regression is a supervised machine learning technique used for classification tasks, aiming to estimate the likelihood that a data instance falls into a specific class. It is a statistical method that examines the relationship between two variables. This article delves into the basics of logistic regression, its various types, and practical applications.

In this project, Logistic Regression was applied as a multinomial model to handle the classification of four distinct classes in the cyberbullying detection task. Instead of just distinguishing between cyberbullying and non-cyberbullying (a binary problem), this model was used to predict one of four possible outcomes, which correspond to different types or levels of bullying behavior which can be mathematically expressed as represented in fig 4.5.

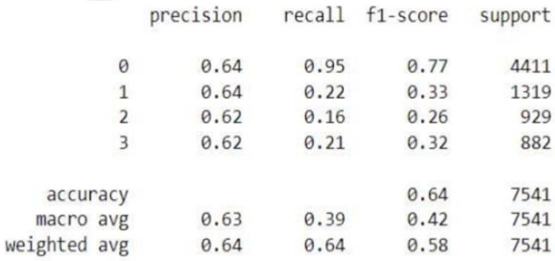


```
              precision    recall  f1-score   support

           0       0.59      0.99      0.74      4411
           1       0.33      0.01      0.01      1319
           2       0.13      0.00      0.00       929
           3       0.37      0.03      0.05       882

    accuracy                           0.59      7541
   macro avg       0.36      0.26      0.20      7541
weighted avg       0.46      0.59      0.44      7541
```

Figure 5: Performance analysis of Logistic regression

4.3.2 Random Forest Classifier

The Random Forest Classifier is highly efficient for cyberbullying detection as it builds an ensemble of decision trees, each trained on a subset of the data. This ensemble approach reduces the risk of overfitting, a common problem in machine learning, especially with complex datasets like those involving natural language. Additionally, Random Forest can effectively capture non-linear relationships and interactions between features, allowing it to learn nuanced patterns in the data that signify cyberbullying behavior. Its robustness and accuracy make it a preferred choice for this task.

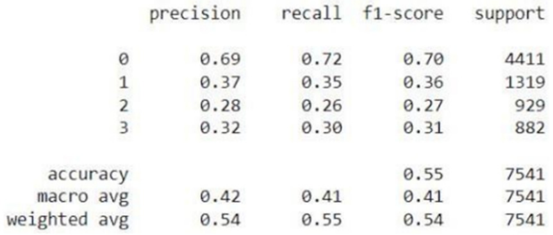


```
              precision    recall  f1-score   support

           0       0.64      0.95      0.77      4411
           1       0.64      0.22      0.33      1319
           2       0.62      0.16      0.26       929
           3       0.62      0.21      0.32       882

    accuracy                           0.64      7541
   macro avg       0.63      0.39      0.42      7541
weighted avg       0.64      0.64      0.58      7541
```

Figure 6: Performance analysis of Random Forest Classifier

4.3.3 Decision Tree Classifier

The Decision Tree Classifier is a versatile supervised learning algorithm used for both classification and regression tasks. It is widely valued for its ability to provide interpretable decision-making, as the model's decisions can be easily visualized and understood.

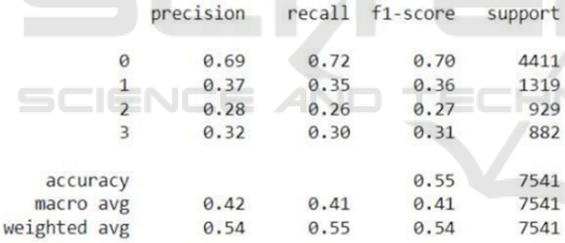


```
              precision    recall  f1-score   support

           0       0.69      0.72      0.70      4411
           1       0.37      0.35      0.36      1319
           2       0.28      0.26      0.27       929
           3       0.32      0.30      0.31       882

    accuracy                           0.55      7541
   macro avg       0.42      0.41      0.41      7541
weighted avg       0.54      0.55      0.54      7541
```

Figure 7: Performance analysis of Decision Tree Classifier

4.3.4 Naive Bayes Classifier

The Naive Bayes Classifier is a probabilistic machine learning algorithm commonly used for classification tasks. It is built on Bayes' Theorem, which calculates the probability of a hypothesis given certain evidence. The algorithm operates under the naïve assumption that all features are independent of one another, which simplifies computation and allows it to handle large datasets efficiently. Despite this assumption being unrealistic,Naive Bayes often delivers strong performance, particularly in tasks such as text classification and spam filtering. The classifier determines the probability of each class and selects the one with the highest posterior probability.

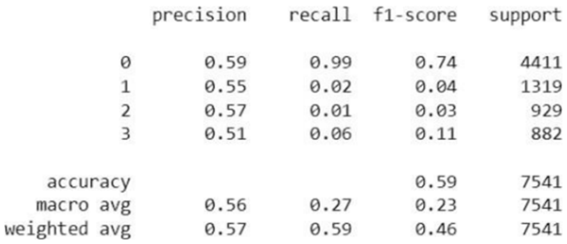


```
              precision    recall  f1-score   support

           0       0.69      0.72      0.70      4411
           1       0.37      0.35      0.36      1319
           2       0.28      0.26      0.27       929
           3       0.32      0.30      0.31       882

    accuracy                           0.55      7541
   macro avg       0.42      0.41      0.41      7541
weighted avg       0.54      0.55      0.54      7541
```

Figure 8: Performance analysis of Naïve Bayes Classifier

4.3.5 Gradient Boost Classifier

The Gradient Boosting Classifier is a highly effective ensemble learningalgorithm used for both classification and regression tasks. It constructs the model in stages with each new model aiming to correct the mistakes of the previous ones. Gradient boosting combines several weak learners, usually decision trees, to form a strong predictive model. Unlike other ensemble methods like bagging, gradient boosting emphasizes sequential learning, where each tree is trained to minimize the residual errors, or gradients, from the prior model.

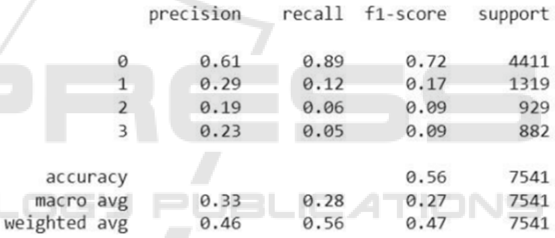


```
              precision    recall  f1-score   support

           0       0.59      0.99      0.74      4411
           1       0.55      0.02      0.04      1319
           2       0.57      0.01      0.03       929
           3       0.51      0.06      0.11       882

    accuracy                           0.59      7541
   macro avg       0.56      0.27      0.23      7541
weighted avg       0.57      0.59      0.46      7541
```

Figure 9: Performance analysis of Gradient Boost Classifier

4.3.6 K-Nearest Neighbor

The K-Nearest Neighbor (K-NN) classifier is a straightforward, non-parametric algorithm used for both classification and regression tasks. It operates by storing the full training dataset and classifying new data points based on the majority class of their k-nearest neighbors within the feature space. The distance between points is usually determined using metrics like Euclidean distance, though other measures such as Manhattan or Minkowski distances can also be applied.



```
              precision    recall  f1-score   support

           0       0.61      0.89      0.72      4411
           1       0.29      0.12      0.17      1319
           2       0.19      0.06      0.09       929
           3       0.23      0.05      0.09       882

    accuracy                           0.56      7541
   macro avg       0.33      0.28      0.27      7541
weighted avg       0.46      0.56      0.47      7541
```

Figure 10: Performance analysis of K-Nearest Neighbor Classifier

4.4 Performance Evaluation

Random Forest achieved the highest accuracy among the models with an accuracy of 64%, followed closely by the Voting Classifier at 63%. These ensemble-based models benefit from the aggregation of predictions, improving overall accuracy by reducing overfitting and capturing complex patterns in the data. Logistic Regression came in next with 58% accuracy, performing better than simpler models like Decision Tree and Naive Bayes, but still lagging behind ensemble approaches due to its linear nature. K-Nearest Neighbor and Decision Tree both performed similarly with accuracies in the mid-50s, reflecting their challenges in handling high-dimensional text data. Multinomial Naive Bayes showed the weakest performance, highlighting its limitation in capturing interdependent relationships in the dataset. Overall, ensemble methods like Random

Forest and Voting Classifier are more suitable for handling the complexities of cyberbullying detection which is displayed in fig 4.16, as they combine the strengths of various models to deliver more robust and accurate predictions .

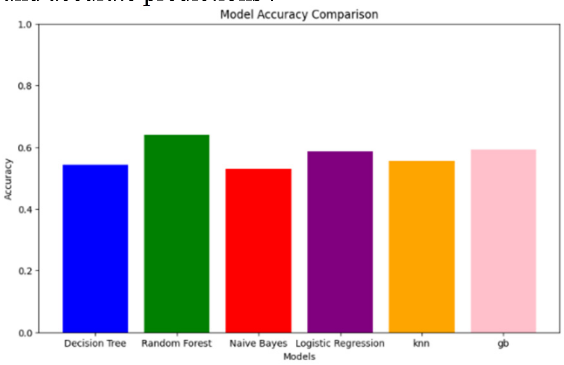


Figure 11: Accuracies of Machine Learning models

5 CONCLUSIONS

In summary, this project presents an investigation into detecting cyberbullying in text data using machine learning models. The data collection involved gathering and processing text samples related to online interactions. Logistic Regression came with 58% accuracy, performing better than simpler models like Decision Tree and Naive Bayes, but still lagging behind ensemble approaches due to its linear nature.Random Forest achieved the highest accuracy among the models with an accuracy of 64%. These ensemble-based models benefit from the aggregation of predictions, improving overall accuracy by reducing overfitting and capturing complex patterns in the data.Gradient-Boost Classifier provides 59% accuracy,K-Nearest Neighbor and Decision Tree both performed similarly with accuracies in the mid-50s, reflecting their challenges in handling high-dimensional text data. Multinomial Naive Bayes showed the weakest performance, highlighting its limitation in capturing interdependent relationships in the dataset.Overall, ensemble method like Random Forest is more suitable for handling the complexities of cyberbullying detection which is displayed as they combine the strengths of various models to deliver more robust and accurate predictions.

Table 2: Analysis of different models

Model	Accuracy(%)
Logistic Regression	58
Decision Tree Classifier	54
Random Forest Classifier	64
Multinomial NB	52
K-Nearest Neighbor	55
Gradient Boost Classifier	59

REFERENCES

Al-Hassan., Al-Dossari, H. (2019, February). Detection of hate speech in socialnetworks: a survey on multilingual corpus. In 6th international conference on computer science and information technology (Vol. 10, pp. 10-5121).

Alkasassbeh M., Almomani A., Aldweesh A., Al-Qerem A., Alauthman M., Nahar K., Mago B. (2024, February). Cyberbullying Detection Using Deep Learning: A Comparative Study. In 2024 2nd International Conference on Cyber Resilience (ICCR) (pp. 1-6).

Altynzer Baiganova.,Saniya Toxanova.,Meruert Yerekesheva(2024,January).Hybrid Convolutional Recurrent Neural Network for Cyberbullying Detection on Textual Data,15(5).

Ammar Almomani., Khalid Nahar., Mohammad Alauthman (2023,March).Image cyberbullying detection and recognition using transfer deep machine learning.(14-26).

Ejaz N., Razi F., Choudhury S. (2024). Towards comprehensive cyberbullying detection: A dataset incorporating aggressive texts, repetition, peerness, and intent to harm. Computers in Human Behavior, 153, 108123.

Islam M., Uddin M. A., Islam L., Akter A., Sharmin S., & Acharjee U. K. (n.d.). Cyberbullying detection on social networks using machine learning approaches. IEEE. Determine the most effective approach for identifying harmful online interactions. (pp. 1-6).

Kini M., Keni A., Deepa.,Deepika K. V. (2020). Cyber-bullying detection using machine learning algorithms. In 2020 International Conference on Advances in Data Engineering and Intelligent Computing Systems (ADICS) (pp. 1-6).

Naveen Kumar M.R., Vishwachetan D(2024).An Efficient Approach to Deal with Cyber Bullying using Machine Learning: A Systematic Review.(pp. 1-8).

Orelaja A., Ejiofor C., Sarpong S., Imakuh S., Bassey C., Opara I., Akinola,O. (2024). Attribute-specific Cyberbullying Detection Using Artificial Intelligence.Journal of Electronic & Information Systems, 6(1), 10- 21.

Reynolds, Kelly, April Kontostathis, and Lynne Edwards. "Using machine learning to detect cyberbullying." 2011 10th International Conference on Machine learning and applications and workshops. Vol. 2. IEEE, 2011.