IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)
An International Open Access, Peer-reviewed, Refereed Journal

Developing an integrated Anti-cyberbullying system

Dr. Kavitha Jayaram*, Navya Dhankar*, Ajitesh Kumar Soni*, Abhishek Ranjan*

*Department of Computer Science and Engineering BNM Institute of Technology, Bengaluru

***Abstract*—** This paper addresses the pervasive threat of cyberbullying by developing an Integrated Anti-Cyberbullying System. Inspired by the need for a secure online environment, it integrates advanced technologies and proactive strategies. Motivated by the commitment to empower users, the system facilitates effective detection, prevention, and response to cyberbullying. Ultimately, it aims to reshape the online realm, fostering a culture of empathy and digital citizenship for a safer, respectful digital landscape and to classify and compare the different strategies, algorithms and technologies in the existing literature and provide information of assistance for the current state and challenges in Anti-Cyberbullying Detection Model.

Keywords: Cyberbullying, Digital Safety, Online Security, Detection Strategies, Profanity.

I. Introduction

Cyberbullying has risen prominently as a pervasive and destructive issue in the digital age, affecting individuals across all age groups. The increasing reliance on online platforms for communication has created a breeding ground for harassment, intimidation, and humiliation. To counteract this alarming trend, There is a crucial need for creative solutions that safeguard individuals from the detrimental effects of cyberbullying. This project aims to address this challenge by developing an integrated anti-cyberbullying system that combines advanced technologies with proactive strategies to create a safer online environment.

This paper motivation comes from the realization that cyberbullying can have serious, frequently life-altering effects. In addition to emotional suffering, victims may have long-term psychological repercussions. as online harassment is so common nowdays, safeguarding people from damage requires a thorough and cutting-edge strategy. Our objective is to offer users with the necessary tools and resources to identify, stop, and deal with incidents of cyberbullying in an efficient manner by creating an integrated anti-cyberbullying system. The primary idea behind this project is that it is everyone's right to participate in the digital world without fear of abuse or harassment.

II. Literature Survey

A. *Development and Implementation of an AI-Driven Anti-Cyberbullying System: Integration,Validation, and Performance Evaluation*

This study used the methods of machine learning The Optimized Linear Support Vector Machine (SVM) and Multinomial Naïve Bayes to create an artificial intelligence (AI) driven anti-cyberbullying system. Cyberbullying messages sent online are identified and intercepted by the system before it can reach its targeted user. A Restful API created using the Flask Python framework is part of the implementation, which enables social media networks to incorporate the anti-cyberbullying system. A diverse range of datasets, including those from YouTube, Wikipedia Talk pages, Kaggle, and Twitter, were utilized to train the model for detecting cyberbullying. The usual methods for cleaning and preparing data were used, such as addressing missing values, removing noisy data, and presenting the information in an appropriate manner. Dealing with every row that has a null value is necessary to preserve data integrity; hence opted for a Python library tool called PyCaret. To make the data more acceptable for easy data mining and pattern recognition, it needed to undergo data transformation. The prepared data was employed to train using multinomial Naïve Bayes and Linear SVM for the detection of cyberbullying.

To make the trained model available to the world, a Restful API built with Flask was created. Social media networks can use the API to check and intercept communications containing cyberbullying by sending it a parameterized input. A method was developed to keep the web service from becoming overloaded with requests. The web API was designed to be supported by a Windows service running in the background. Therefore, this information is conscientiously uploaded to a CSV file stored on the IBM Object Cloud Storage. Because of its intelligence, the system can recognize new records automatically and update the CSV file whenever an existing record is changed. To guarantee that data is verified and pushed from the integrated data warehouse to the CSV file, the scheduler repeatedly makes set-interval requests to the web API.

For Validation and cross-validation purposes, Confusion matrices and F1 scores were used to verify if the model is accurate. To guarantee the model's performance on unknown data, cross-validation was carried out.

To communicate with the Restful API, a graphical user interface (GUI)-based chatbot automation system was created. The system showed a low mean square error and 96% accuracy.

This study compared the delivery times of messages both with and without the backend message interception procedure. The implementation was completed in real-time, with very little backend processing.[1]

B. *Enhancing Cyberbullying Detection through Convolutional Neural Networks: Framework, Implementation, and Optimization*

Convolutional Neural Network (CNN) implementation is incorporated into the suggested system for accurate and efficient analysis. Deep learning, which uses layers of neurons with self-learning capabilities, is inspired by the central nervous system of mammals. Three layers make up the deep learning model: the input layer, the hidden layer, and the output layer.

Cleaning up the data and putting it into a format that will be useful for model training is the first step in the process. The process involves eliminating superfluous words, characters, additionally converting the data to lowercase. To maintain consistency, sentences are broken up with padding to guarantee an even word count. Converting the processed data into vector form makes the model easier to accept.

The Sequential Layer is among the essential layers used by the CNN model. A stack of neural network layers called the Sequential model is presented by the Keras deep learning framework. Dense connections are used to connect nodes from each layer. In order to create feature maps and identify particular patterns, the input data is convolved with filters—the central components of CNN model layers. Filters will multiply each element individually and then add the result to make a feature map and non-linearity is introduced with activation functions like Sigmoid and ReLu. Text parsing into words and sentences is aided by NLTK.

After the model is defined, metrics, loss functions, and optimizers are used to compile it. Weights are updated by optimizers throughout training. With the use of parameters like batch size, validation data, and epochs, the fit() function trains the model. Batch size determines instances prior to weight updates, while epochs indicate exposures to the training set. The model's performance is evaluated using validation data, which iterates until convergence.

To sum up, the system's advantage resides in the deep learning framework's iterative examination of CNN layers. While the CNN model's sequential architecture, optimizers, and activation functions enable efficient learning, data pre-processing guarantees high-quality input. With the use of validation data, batch size, and epochs, the training process creates a model that can make precise predictions.[2]

C. Exploring Diverse Techniques for Cyberbullying Detection: Methodology, Feature Extraction, and Algorithmic Comparison

In context of cyberbullying, this paper uses a similar methodology with some technological differences. It involves polling, pattern recognition, and the application of machine and deep learning algorithms. Gathering pertinent data is the first stage in these investigations. Either directly scraping web pages or parsing API-based data can be used to do this. The decision is dependent on the kind and level of difficulty of the activity.

Following data collection, the data will be cleaned, which involves addressing missing values, fixing mistakes, and resolving dataset inconsistencies. Other tasks include tokenization (breaking down text into words), stop word removal (removing terms like "and," "the," and "is"), stemming, and lemmatization.

Following the data's successful cleaning, feature extraction is completed, involving For machine learning algorithms to function, text data must be converted into numerical vectors. Term Frequency-Inverse Document and Bag-of-Words (BoW) are two popular methods. The frequency (TF-IDF)

Words can also be included in a continuous vector space by using word embeddings, such as Word2Vec(Word-to-Vector) or GloVe.

The next step is feature generation, which will improve The standard of data accessible for machine learning models by creating new features from the raw data that is already available. The process of selecting a subset among the most pertinent features from the initial collection of features is known as feature selection. Eliminating unnecessary or redundant features aims to lower dimensionality, increase interpretability, and maybe improve model performance.

This study compares the accuracy provided by many algorithms for machine learning which include Binary classification,

Linear Regression, Decision trees, KNN, and Naive Bayes, Random forest, SV, etc. Binary Classification: Standard measures like as accuracy, precision, recall, and F-score are used in comparative analysis.

The accuracy of logistic regression is 0.83, whereas the accuracy of support vector machines is 0.65. Accuracy of stochastic gradient descent (SGD): 0.8

Random Forest: 0.85 Accuracy, LDA (Linear Discriminant Analysis): Accuracy: 0.87; Neural Network: 0.92.

Random Forest achieved the greatest point of general classifier execution with a SMOTE of 0.711, a big classifier precision of 91.153, and an f-measure of 0.898.[3]

D. Ensemble-Based Voting Models for Offensive vs. Non-offensive Tweet Classification: Methodology and Performance Evaluation

In order to divide the contents of this study report into two categories—“offensive” and “non-offensive,” the author suggested developing a single and double ensemble-based voting model.

It uses four learning classifiers—Multinomial I Bayes (MNB), Logistic Regression (LR), Decision Tree Model (DT), and Linear Support Vector Classifier (LSVC)—to train and test the models on the dataset. Three ensemble models—Gradient Boosting Classifier (Gboost), AdaBoost (AdB), and Bagging— enhance the effectiveness of each classifier by combining their predictions. Two feature extraction methods—BoW and Term Frequency-Inverse Document Frequency (TF-IDF) and BoW—are used to convert the textual data into numerical vectors.

Increasing accuracy of the voting process, two new classifiers were implemented: the Single Level Ensemble Model (SLE) and the Double Level Ensemble Model (DLE).ensemble models by the use of weighted or majority voting procedures.

They took 9093 tweets at random from a benchmark dataset that was initially produced. The two columns in each data file are “class” and “tweet,” and the data are saved as CSV files. The tweet displays the compilation of all tweets, regardless of whether they are offensive or not. The labels “0” and “1” denote non-offensive tweets and offensive tweets, respectively.

The model has been fed unprocessed tweets as Input. To get the data in a clean format, pre-processing techniques like noise reduction, tokenization, stemming, and detokenization have been applied. A train set (80%) and a test set (20%) have been created using the cleaned label dataset.

Two methods for extracting features: Bag of Words TF-IDF has been employed. Using each tweet as input, the Bag of Words (BoW) technique counts how many times each word appears in the tweet. This yields a numerical representation of the text called vector features of fixed length. Consequently, raw words are encoded as key-value pairs holding the frequency of every word in the text into a count vector format. A statistical metric called TF-IDF is used to give text data used in mining numerical weights.

By counting how often a term appears in text data, it determines its relative importance across the whole dataset.

This work evaluates the performance, of MNB, LR, DT, LSVC, Gboost, AdB, and Bagging on our dataset. To get the most accurate expected production from each group, these have been separated into groups.

There are now two voting classifiers and two groups. MNB, LR, DT, and LSVC are in V C1; Gboost, AdB and Bagging are in V C2.Hard voting is used in this paradigm to establish the ensemble frameworks. The ensemble model will evaluate a tweet as "offensive" if the majority of classifiers in a classifier group expect it to be such, and vice versa, based on the hard voting mechanism.[4]

E. Ensemble-Based Voting Models for Offensive vs. Non-offensive t Classification: Methodology and Performance Evaluation

The suggested architecture uses an input dataset from the social network as the starting point for the process of identifying cyberbullying actions. Data pre-processing takes input and is used to remove stop words, excess characters, and hyperlinks enhancing the quality of the research data and the analytical processes that follow. The input data undergoes pre-processing before being sent to the feature extraction procedure. Feature extraction is the process of extracting textual elements such as nouns, adjectives, and pronouns as well as word frequency data. The retrieved features are given to the learning algorithm.

The pre-processing and feature extraction unit's processed social network discussion dataset is fed into the suggested algorithm. It uses a genetic algorithm to carry out the evolutionary process, and the chromosome is evaluated using a fuzzy rule set. Every member of the population has their chromosome's fitness value determined, and the result is a list of the cyberbully phrases found in the input dataset.

The Fun-Zen algorithm, It combines generative algorithms with fuzzy rules (logic) on the dataset, is used in this article. Using a GA with fuzzy set genes, the learning module integrates the adaptive component of the system. Adaptive search and optimization algorithms, or Gas, function by drawing inspiration from the ideas of natural selection. The function that needs to be optimized in the suggested system is a fictitious Social Network representation of phrases related to cyberbullying.

A fuzzy rule-based system functions similarly to a situational aid. They lack precision or are unclear. Its foundation lies in applying principles that capture the hazy or unclear parts of a circumstance.

Genetic algorithms can be thought of as a technique that emulates natural selection. It use a fitness function to select the best options from a population of viable ones. Then, to produce new generations and progressively enhance the overall solution, it merges or modifies these solutions.

The capability of a chromosome to classify the activity Is called the fitness of the chromosome. The chromosome having a greater fitness value gives the classified output.

It assigns a fitness value to the terms by using the present information in the population of chromosomes. The chromosome having a greater fitness value will be obtained as classified output.[5]

III. Comparative Analysis of Methodologies

After going through the research papers and getting a deep understanding of various methodologies proposed for Cyberbullying detection, we have comprehended limitations of different approaches.

METHODOLOGIES	LIMITATION
Multinomial Naïve Bayes, Linear Support Vector Machine [1]	Quantity and quality of training data
TF-IDF technique [3]	Limited Dataset Focus: Twitter
CNN technique [4]	CNN complexity: resource-intensive computations
FuzGen Learning Algorithm [5]	limited scalability to handle large datasets

Table I: Comparative study of methodologies

IV. Conclusion

In this work, we surveyed a wide range of Anti-Cyberbullying detection techniques and models by different authors. A brief introduction about the related work is given, and major aspects of those systems are reviewed.

By combining cutting-edge technologies with proactive strategies, this project aims to equip individuals with the tools necessary to successfully combat cyberbullying. A system like that has a big impact on society, as it not only protects individuals from harm but also promotes a culture of empathy, respect, and responsible digital citizenship. The integration of this anti-cyberbullying system into online platforms holds the potential to reshape the digital landscape, creating a place where people can openly express themselves without worrying about being harassed.

V. Proposed Methodology

After surveying a wide range of paper and understanding the methodologies used and their limitations we able to conclude and make our architecture achieve the highest accuracy given the limited amount of dataset.

Our Proposed solution is to use a web extension for frontend and in backend we first extract the comments from the social media sites and then these comments are data pre-processing and then its is classified as offensive and non-offensive also saves this data is the database in the procedure and take the necessary steps.

The system collects data from various online sources, such as social media platforms, blogs, forums, etc. The data consists of text messages or images that may or may not contain cyberbullying. After the Data collection process, the system collects a variety of comments using the concept of web scraping. This is a technique, employed to automatically extract information from websites. It involves fetching web pages, parsing the HTML or XML content, and extracting valuable data for analysis, the system can access a diverse range of user-generated content that captures the essence of online interactions occurring in various digital environments.

After gathering the comments, the system starts the process of data preparation, which is a means of improvement. This is an important stage where the raw data must be transformed into a format that allows for meaningful analysis. This includes the process of tokenization, stop word removal, text normalization, etc. A cleaned dataset remains after stop words, punctuation, and noise are eliminated. Following suit, tokenization, stemming, and lemmatization dissect the text into its constituent parts, reduce words to their root forms, and guarantee consistency in the representation of words.

The pre-processed data finds refuge in the system's internal storage, a virtual archive that protects the information extracted from the internet. This reservoir allows the system to revisit and investigate trends throughout time, providing the framework for further phases of research.

After obtaining an improved dataset, sentiment analysis and offensive comment detection are performed on the data. It uses cutting-edge methods, such as deep learning or machine learning algorithms, to extract the emotional undertone from each comment. This phase is comparable to reading between the lines in a person's expression, categorizing remarks into favourable, unfavourable, or neutral groups according to the feelings that underlie them. The next step will be Identifying whether the comment is offensive or not.

After identifying potential language minefields, the system goes ahead and categorizes comments as either offensive or not. This classification depends on a sophisticated interpretation of civility and online etiquette. If the comment is not offensive the process will terminate.
If the comment(s) are offensive it will not be visible to the user or on the particular website or page.

The classified remarks are also permanently stored in an extensive database as the system navigates the terrain of emotion and offense analysis.

Hide from User: Putting Safety Measures in Place:

Providing a buffer between users and offensive remarks is the last line of defence. The comments that are being classified as offensive will be hidden from the user as a result user will not be able to see the comments. Thus, protecting the user from cyberbullying.

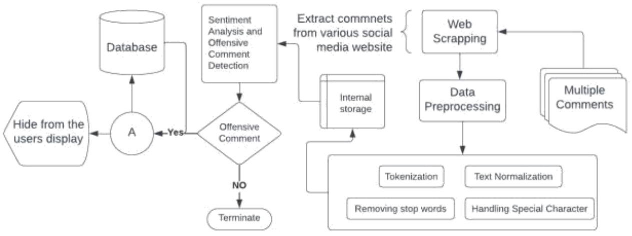
Database
Sentiment Analysis and Offensive Comment Detection
Extract commnets from various social media website
Web Scrapping
Internal storage
Data Preprocessing
Multiple Comments
Hide from the users display
A
Yes
Offensive Comment
NO
Terminate
Tokenization
Text Normalization
Removing stop words
Handling Special Character

Fig. 1. Architecture of anti-cyberbullying system

REFERENCES

[1] Tosin Ige and Sikiru Adewale Interception of Cyberbully Contents in a Messaging System by Machine Learning Algorithm (IJACSA)

[2] Volume 13, Issue 5, 2022 of the International Journal of Advanced Computer Science and Applications, Vidya Chitre

[3] Fourth International Conference on Computing Methodologies and Communication (ICCMC-2020), Saloni Mahesh Kargutkar and Prof. Vidya Chitre

[4] Daniyar Sultan, Azizah Suliman and Aigerim Toktarova Cyberbullying Detection and Prevention: Data Mining in Social Media 11th International Conference on Cloud Computing, Data Science & Engineering (Confluence-2021)

[5] Cyberbullying Detection: An Ensemble Based Machine Learning Approach Priyo Ranjan Kundu Prosun, Kazi Saeed Alam, and Shovan Bhowmik Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV-2021)

[6] Online Social Network Bullying Detection Using Intelligence Techniques, B. Sri Nandhini and J. I. Sheeba, International Conference on Advanced Computing Technologies and Applications (ICACTA-2015)