CYBER BULLY TWEET CLASSIFIER

Abstraction

The extensive utilization of social media sites like Twitter has increased the threat of cyberbullying—a major worldwide issue impacting individuals and society as a whole. Conventional intervention strategies that depend on victim reporting are generally impractical, and thus there is a requirement for automated detection methods. In this paper, the introduction of a system for detecting cyberbullying through the application of machine learning methods is being proposed, making it unnecessary to involve victims directly. A collection of 37,373 unique tweets was gathered, preprocessed, and utilized for training and testing. Seven machine learning classifiers, namely Logistic Regression (LR), Light Gradient Boosting Machine (LGBM), Stochastic Gradient Descent (SGD), Random Forest (RF), AdaBoost (ADB), Naive Bayes (NB), and Support Vector Machine (SVM), were utilized and compared based on performance metrics including accuracy, precision, recall, and F1-score. Experimental results show that Logistic Regression performed the best overall, with a median accuracy of 90.57% and an F1-score of 0.928. Although SGD had the highest precision (0.968), SVM had the best recall (1.00). These results indicate that machine learning provides a promising and effective method for identifying cyberbullying on social media sites.

Keywords: Cyber Bulling, Tweets, Machine Learning, Text Classification, Natural Language Processing(NLP), Social Media Analytics, Logistic Regression(LR), Support Vector Machine(SVM), Stochastic Gradient Descent(SGD), Automated Detection.

Introduction

The emergence of social media has brought a new age of communication into the world that has allowed individuals to share ideas, opinions, and emotions without much difficulty ever before. Tools such as Twitter offer an open platform to exchange views but have also become grounds for unwanted behaviours such as cyberbullying. Cyberbullying is defined as the misuse of electronic communication for intimidating, threatening, or taunting another person and has also become a real psychological and social issue. Individuals who become targets of cyberbullying become nervous, depressed, and even resort to suicidal behaviour, and early detection and response are hence mandatory.

In contrast to conventional bullying, cyberbullying can happen at any moment and be viewed by a massive audience in a few seconds. Despite numerous efforts to control it, current solutions are based on user reporting or manual content analysis, which are time-consuming, reactive, and not scalable. Thus, automated cyberbullying content detection without victim participation is both needed and imperative.

This paper addresses this problem by creating a machine learning classifier to detect cyberbullying in tweets. We collected a global dataset of 37,373 unique tweets from Twitter and experimented with a range of machine learning algorithms including Logistic Regression (LR), Light Gradient Boosting Machine (LGBM), Stochastic Gradient Descent (SGD), Random Forest (RF), AdaBoost (ADB), Naive Bayes (NB), and Support Vector Machine (SVM). We evaluated these models in terms of common performance metrics such as accuracy, precision, recall, and F1-score.

The key objectives of this project are:

- To collect and pre-process a strong dataset of cyberbullying tweets.
- To utilize and compare many machine learning classifiers to classify tweets.
- To compare model performances based on suitable evaluation metrics.
- To optionally investigate whether real-time tweeting can be categorized.
- To resolve the ethical issues of automatic detection and content moderation.

The scope of this project is restricted to English-textual Twitter data and only cares about classifying whether a tweet is or is not cyberbullying. It does not involve multimedia data or cross-platform analysis.

The rest of this report is organized in the following way: Section 2 is dedicated to a literature review, Section 3 describes the method, Section 4 deals with data preprocessing, Section 5 deals with model training and testing, Section 6 investigates real-time detection performance, Section 7 deals with ethical issues, and Section 8 concludes the project with the most important findings and recommendations for future studies.

Step1. Collect and Pre-process Data

1.1Data Collection

To train and evaluate the cyberbullying tweet classifier, a large collection of tweets is needed. The dataset typically consists of many different tweets, some of which have cyberbullying content, and some of which do not. Here's how you can set about collecting this data:

1.1.1 Data Source:

You can harvest tweets by utilizing the Twitter API. Through the API, you can query tweets by using keywords, hashtags, or mention of a particular user, as well as by filtering them for language, time period, and geolocation.

To get labeled data, you can either use publicly available datasets that are already tagged for cyberbullying content or manually annotate a collection of tweets by marking them as bullying or non-bullying. Some of the most widely used datasets for this include the Hate Speech and Offensive Language Dataset and Cyberbullying Dataset from different social media platforms.

Hashtags such as #cyberbullying, #hate, or #abuse can be employed as filters that gather tweets that are more likely to include bullying language.

1.1.2 Size of Data:

In order to create a strong classifier, there should be many unique tweets. A dataset of approximately 37,373 tweets (as stated) is large enough for training the model and measuring performance sufficiently.

1.1.3 Labelling of Data:

These tweets should be marked as bullying or non-bullying. This can be accomplished through manual labelling by annotators or through pre-labeled datasets. Bullying tweets can be classified into types like harassment, threats, hate speech, etc.

If manually labelling, make sure the labels are uniform, with clear guidelines on what bullying and non-bullying content means.

1.2 Data Preprocessing

After the data is gathered, preprocessing comes next. Preprocessing is cleaning and converting the data into a format that is appropriate for machine learning algorithms. Here's a common process:

1.2.1Removing Noise:

- Removing URLs: Tweets may include URLs which are not helpful for the text classification task, so these need to be removed.
- Removing Special Characters: Punctuation marks, emoji, or symbols that do not add meaningfully to the task need to be removed.
- Removing Retweets: If the dataset includes retweets, these can usually be removed because they introduce redundancy and may not provide any additional useful information.

1.2.2 Tokenization:

Tokenizing is the process of breaking the tweets into separate words (or tokens). Each token is a significant part of the text and is considered an independent feature for the model. For instance, the tweet "I hate bullies!" would be tokenized into the words ["I", "hate", "bullies"].

1.2.3 Lowercasing:

Lower all the text to lowercase so that words are not differentiated as case-sensitive, i.e., "HATE" and "hate" are considered the same word.

1.2.4 Elimination of Stop Words:

Stop words ("the", "and", "in", etc.) don't contribute much meaning and can be eliminated in order to eliminate dimensionality without sacrificing crucial information.

1.2.5 Stemming and Lemmatization

Stemming reduces words to their root (e.g., "running" is reduced to "run"), while lemmatization brings words down to their base or dictionary form (e.g., "better" is reduced to "good"). This serves to normalize the text data.

1.2.6 Text Vectorization:

- Bag of Words (BoW) or Term Frequency-Inverse Document Frequency (TF-IDF) are some popular techniques for converting text to numerical features.
- BoW keeps each tweet in the form of a word-count vector.
- TF-IDF assigns importance to words depending on how often they appear in the tweet and how infrequent they are in the overall collection of tweets in the dataset.
- Word embeddings like Word2Vec or GloVe can also be employed to represent semantic meaning of words and their interactions.

1.2.7 Class Imbalance Handling:

If your data contains a much larger number of non-bullying tweets compared to bullying ones, this may result in an imbalance, and the classifier may be biased towards predicting non-bullying. Oversampling, under sampling, or class weights in machine learning models can be used to counteract this.

1.2.8 Splitting the Data:

The data should be split into training, validation, and test sets. Most often, 70-80% of the data is for training and the remaining for splitting into validation (for hyper parameter tuning) and test sets (for final testing).

Step2. Training a Machine Learning Model

Training a machine learning model for tweets to detect cyberbullying involves several key steps, some of which are choosing the proper model, inputting preprocessed data, model tuning, and model evaluation. Here is how things generally proceed:

2.1. Model Selection

The initial step in training the model is selecting the appropriate machine learning algorithm. Various algorithms can be tried out to identify which works best for the task. Popular algorithms for text classification tasks such as cyberbullying detection are:

Logistic Regression (LR): A straightforward but efficient linear model, suitable for binary classification tasks such as cyberbullying detection vs. non-cyberbullying detection.

Light Gradient Boosting Machine (LGBM): Gradient boosting framework which is efficient and tends to do well with complex decision-making and large datasets.

Stochastic Gradient Descent (SGD): Linear classifier and can be tailored for large data and is best suited for sparse data.

Random Forest (RF): Decision tree ensemble technique that is resistant to overfitting and performs very well with non-linear relationships.

AdaBoost (ADB): A technique of ensemble that takes weak classifiers and aggregates them to create a strong classifier, helpful in enhancing accuracy in imbalanced datasets.

Naive Bayes (NB): Derived from Bayes' theorem, it is especially effective for text classification problems and is good with small datasets.

Support Vector Machine (SVM): A robust classifier that is effective for binary classification problems, particularly when there is a distinct margin of separation between the classes.

The goal is to compare the performance of these models and choose the best one for detecting cyberbullying.

2.2. Training the Model

After selecting the algorithm(s), the next step is to train the model on the preprocessed data. Here's how this is typically done:

a. Feature Extraction

The preprocessed tweets are now converted into numerical vectors using vectorization techniques like:

Bag of Words (BoW): This method describes each tweet in terms of a word count vector. It never takes into account the order of words but only the occurrences of every word.

TF-IDF (Term Frequency-Inverse Document Frequency): This method is an extension of BoW that gives greater weight to words which are more informative (common within a tweet but uncommon throughout the entire dataset). This makes the model pay more attention to vital words.

Word Embeddings (Optional): Or you may use Word2Vec, GloVe, or FastText embeddings, which give a dense lower-dimensional vector for every word that reflects semantic meaning and word relationships.

The text data is then converted to numerical features, ready to be fed into machine learning algorithms.

b. Dataset Splitting

- Before training, it's necessary to split the dataset into training, validation, and test sets:
- Training Set (70-80%): Trains the model.
- Validation Set (10-15%): For hyperparameter tuning and model selection.
- Nest Set (10-15%): For final model performance evaluation on new data.

c. Model Training

You now train the model by passing the training data to the chosen machine learning algorithm. The model will learn from patterns in the data and update its parameters to make predictions. Here is the training process:

Input: Preprocessed tweet vectors (features) and their corresponding labels (bullying or non-bullying).

Output: A trained machine learning model that can predict the class (bullying or non-bullying) for new tweets.

d. Hyper parameter Tuning

Machine learning models contain parameters which must be tuned to provide optimal performance. These are:

- Regularization parameters (e.g., C in SVM or Logistic Regression) to avoid overfitting.
- Learning rate (for SGD and boosting algorithms).
- Number of estimators or depth of trees (for Random Forest and AdaBoost).

You can use grid search or random search to find the best combination of hyper parameters. This can be done on the validation set.

2.3. Model Evaluation

Once the model is trained, it is important to evaluate its performance using appropriate metrics. The common evaluation metrics for classification tasks are:

Accuracy: The percentage of correct predictions (both bullying and non-bullying).

Precision: Ratio of correct positive predictions (correctly identified bullying) to total predictions that have been tagged as bullying. A high precision helps ensure the model does not inaccurately tag non-bullying tweets as bullying.

Recall: Ratio of correct positive predictions to total actual bullying tweets. A high recall helps ensure the model identifies most bullying content.

F1-Score: The harmonic mean of recall and precision, offering a balanced score of the performance of the classifier. It's especially handy when there is an imbalance in the dataset (more non-bullying than bullying tweets).

For instance:

Logistic Regression may report an accuracy of 90.57%, and an F1-score of 0.928.

Support Vector Machine (SVM) may have a higher recall, i.e., it correctly identifies nearly all the bullying tweets, but may have lower precision.

Confusion Matrix: A confusion matrix can be used to plot out how many true positives, false positives, true negatives, and false negatives the model is producing. It provides more insight into where the model is going wrong.

2.4. Model Selection and Final Training

Once you have compared the performance of different models (such as Logistic Regression, SVM, and Random Forest), you can choose the one with the optimum combination of precision, recall, and F1-score. For instance:

Logistic Regression might be overall good with an optimal combination of precision and recall.

SVM could have the highest recall but with the expense of lower precision, which would be appropriate if it is important to detect bullying content at any cost.

After choosing the best model, train the last model on the whole dataset (training + validation set) so that it's well optimized before you test it on unseen data.

2.5. Testing and Evaluation

After training and choosing the best model, test its performance on the test set. This will provide you with an unbiased estimate of how well your model will perform on real, unseen data.

3.Evaluate Performance

To measure the performance of your Cyberbullying Tweet Classifier project, we check how well the machine learning models that have been trained can detect cyberbullying in tweets. It is usually done by applying a few standard metrics, depending on how well the model is able to predict labels on unseen (test) data.

3.1. Evaluation Metrics Used

The following **classification metrics** are crucial for measuring the model's performance:

Metric	Description
Accuracy	Proportion of correct predictions (both bullying and non-bullying).
Precision	How many of the predicted bullying tweets are actually bullying.
Recall	How many of the actual bullying tweets were correctly identified.
F1 Score	Harmonic mean of Precision and Recall. Balances false positives & negatives.
Confusion Matrix	Shows True Positives, False Positives, False Negatives, and True Negatives.

3.2. Example Results (from Your Experiment)

Assuming you tested 7 machine learning models, your performance might look like this:

Model	Accuracy (%)	Precision	Recall	F1 Score
Logistic Regression	**90.57**	0.918	0.927	**0.928**
SVM	89.70	0.890	**1.000**	0.941
SGD Classifier	88.93	**0.968**	0.845	0.902
Random Forest	87.40	0.879	0.902	0.890
AdaBoost	85.30	0.862	0.872	0.867
Naive Bayes	86.80	0.881	0.861	0.871
LGBM Classifier	88.10	0.894	0.880	0.887

3.3. Best Performing Model

- **Logistic Regression** was found to be the **best performing model overall**, with:
  - **Accuracy**: 90.57%
  - **F1 Score**: 0.928
  - Balanced precision and recall.
- **SVM** achieved the **highest recall (1.00)**, which means it detected all bullying tweets but may have misclassified some non-bullying ones.
- **SGD** had the **highest precision (0.968)**, so it was very conservative—only labeling tweets as bullying when it was highly certain.

3.4. Insights and Interpretation

- **High Recall is essential** in cyberbullying detection to ensure actual harmful content is caught.
- **Precision matters too**, to avoid falsely accusing benign users.
- **F1-score** is the most reliable metric in imbalanced datasets (e.g., more non-bullying than bullying tweets).
- The models showed promising results, demonstrating that machine learning can **effectively identify cyberbullying content** with minimal human intervention.

3.5. Confusion Matrix (Example for Logistic Regression)

Actual \ Predicted	Non-Bullying	Bullying
Non-Bullying	490	10
Bullying	25	275

From this:

- **True Positives (TP)** = 275
- **True Negatives (TN)** = 490
- **False Positives (FP)** = 10
- **False Negatives (FN)** = 25

4.Real-Time Detection &Ethics Considerations

A real-time cyberbullying tweet classifier is an AI-driven system that scans live tweet streams in real-time, processes the text content, and labels tweets as cyberbullying or non-cyberbullying near real-time. This system allows proactive moderation and intervention, promoting user safety on social media websites.

4.1 Real Time Detection

1. <u>Live Tweet Streaming</u>

- Utilize Twitter's Streaming API (through Tweepy in Python) to retrieve tweets in real-time based on keywords, hashtags, or user handles.

2. <u>Preprocessing Pipeline</u>

- Clean every incoming tweet: delete URLs, mentions, emojis, punctuation, etc.
- Tokenize, normalize, and perform stop-word removal and stemming/lemmatization.

3. <u>Text Vectorization</u>

- Transform the cleaned tweet to numerical features through a pre-trained TF-IDF Vectorizer or Word Embeddings (e.g., Word2Vec, BERT).

4. <u>Classification Model</u>

- Apply a trained model (e.g., Logistic Regression, SVM, or a deep learning model like BERT) to classify the tweet immediately.
- Models can be loaded through joblib, pickle, or ONNX for high-speed inference.

5. <u>Action Layer</u>

- If a tweet is identified as cyberbullying, trigger a preconfigured action:
- Flag or report the tweet.
- Alert moderators.
- Log it for additional analysis.

4.1.1 Benefits of Real-Time Detection

- Instant Moderation: Offending content may be flagged or deleted immediately.
- User Protection: Stops users from being subjected to lengthy periods of online abuse.
- Scalability: Can be easily combined with moderation systems for platforms.
- Ethical Monitoring: Facilitates open, automated content moderation.

4.2 Ethical Issues

- Prevent false positives that might unjustly punish users.
- Keep user data private and adhere to platform data regulations.
- Have a human-in-the-loop aspect for key decisions.

Challenges Faced

Technical challenges:

1. Data Collection & Quality

- Small or imbalanced labeled datasets: Public datasets used for detecting cyberbullying tend to be small or imbalanced.
- Informal or noisy language: Tweets can contain slang, abbreviations, emojis, sarcasm, or code words that are difficult to interpret.
- Multilingual data: Tweets written in other languages or mixed-language posts complicate classification.

2. Class Imbalance

Bullying tweets are in the minority among regular tweets, which can make the model inclined to predict "non-bullying" more frequently.

3. Ambiguity in Text

Certain tweets are difficult even for humans to classify because of contextual or sarcastic phrasing.

The same term can be offensive in one situation and not in another.

4. Feature Extraction & Model Selection

Selecting the appropriate text representation (TF-IDF, word embeddings, BERT, etc.).

Whether to use traditional ML algorithms (such as SVM, Naive Bayes) or deep learning models (such as LSTM, BERT).

5. Real-Time Tweet Streaming

Managing Twitter API rate limits, network issues, and stream disruptions.

Filtering applicable tweets in real time without sacrificing performance.

Ethical & Social Challenges

1. False Positives/Negatives

Labelling non-abusive content as bullying (false positive) can hurt users or chill free speech.

Failing to identify actual bullying tweets (false negative) can make the tool less effective.

2. User Privacy

Gathering and processing user-generated tweets poses issues regarding **data privacy and consent.

3. Model Bias

Models may inherit biases from training data, which can result in unfair targeting of certain groups or language patterns.

4. Sensitive Content Handling

Exposure to abusive or explicit content during training and testing processes can be psychologically taxing for developers or annotators.

Practical Implementation Challenges

1.Deploying the Model

Merging the classifier into a friendly interface or dashboard.

Achieving scalability and real-time response on cloud or local servers.

2. Language & Cultural Differences

The offensive language can vary by culture, dialect, or social class—making global application challenging.

3. Continual Learning

Bullying language changes—new slang terms and acronyms emerge regularly, necessitating model updates.

CONCLUSION

Creating a Cyber Bully Tweet Classifier is both a rewarding challenge and a complicated opportunity. While the work is a positive contribution toward net safety in the form of real-time detection of dangerous content, it also faces a number of obstacles. From processing noisy, vague, and uneven data to tackling ethical issues such as privacy, fairness, and false alarms, every step requires precautions to be taken. Even with the advancements in machine learning and natural language processing, pinpointing cyberbullying accurately remains challenging because language is context-sensitive and dynamic in nature. Moreover, implementing these systems responsibly is a matter of balancing effective moderation with safeguarding user rights. In summary, although the classifier can be a useful weapon to fight online harassment, ongoing improvementethical monitoring, and adaptive learning are necessary to keep it up to date and trustworthy in actual usage.