Published Online First: November, 2024
https://dx.doi.org/10.21123/bsj.2024.10743
P-ISSN: 2078-8665 - E-ISSN: 2411-7986

Baghdad Science Journal

Unmasking Online Hostility: Analysing and Mitigating Hate Speech in Social Media

Jawaid Ahmed Siddiqui*[1], **Siti Sophiayati Yuhaniz**[1], **Zulfiqar Ali Memon**[2]*

[1]Razak Faculty of Technology and Informatics, University Technology Malaysia, Kuala Lumpur, Malaysia.
[2]Fast School of Computing, National University of Computer and Emerging Sciences, Karachi, Pakistan.
*Corresponding Author.

Received 21/01/2024, Revised 13/07/2024, Accepted 15/07/2024, Published Online First 20/11/2024

Abstract

The social media platforms have been generating an enormous amount of data for every second. Twitter, in practice by the individuals is producing more than six hundred tweets in each second. While freely posting opinions and expressions by users, it is very difficult to confine the hate speech shared against any individual, religion or any ethnic group. Consequently, the persons targeted by such hateful content get frustrated. In this regard the different approaches have been solving this serious problem but, sometimes unable to achieve satisfactory results. Therefore, we propose different Machine Learning models to classify given data in two categories, offensive or non-offensive. The experiments were conducted on Twitter data generated by ourselves using Twitter API and Tweepy library by Python. The generated results were evaluated based upon various metrics such as accuracy, precision, recall, F1-measure and MCNEMAR test. Compared to the different machine learning algorithms, random forest ensemble classifier outperformed against other algorithms, the novelty and contribution of our research paper is: The development of Twitter dataset that consists of several tweets containing 11 object variables with four different class variables showing the different offensive levels, Machine Learning algorithms' application to detect the hate speech, Comparative analysis of different Machine Learning algorithms against different evaluating metrics including McNemar Test. The significance of proposed technique is well explained by the Twitter datasets generated through Twitter API and Tweepy library by Python.

Keywords: Hate speech detection, Machine Learning, Natural Language Processing, Social media, Text Classification.

Introduction

With the evolution of technology and the fastest communication through social media networks[1], different individuals can express their opinions, feelings or emotions without any restriction[2]. The interaction among users on social media platforms[3] for instance Twitter, Facebook, YouTube, Snapchat, Instagram etc that generate enormous data that is significant to mine or extract the valuable insights, such as hate speech sentiment analysis and sarcasm detection[4,5]. People use social media platforms for different purposes including communication that also led to several issues containing propagation of hate messages[6, 7]. Hate speech in social media aspect is defined as the online shared tweet or post showing hatred against towards any instance, such as religion, ethnicity, race, gender, colour, sexual orientation or any individual's tendency towards any political group[8,9]. Although, social media platforms also suggest the option to its users to unfollow or unfriend those persons spreading such disgusting remarks[10]. It has risen on such levels due to the ease with which people can access social media platforms[11, 12] such as Facebook, Twitter or Instagram and just post their views those results in destructive consequences in society[13]. Circulation of such offensive and unacceptable expressions shared on social media is massive threat to victims throughout the world. Due to these prevalent opinions, the victims suffer from depression[14], frustration, violence and sometimes, commit suicide[15].

Meanwhile, such catastrophic disease in the form of hate speech needs an immediate and technical solution as overcoming this problem manually is only time-consuming and worthless. Existing research studies for the hate speech in social media platforms can be classified into two categories, such as analyzing semantics, context and the hate speech symbols, while another category is to detect, recognize and predict the hate speech[16]. In literature, sematic analysis can be done through ordinal semantic approaches while the prediction can be done through information processing paradigms[17, 18]. Twitter, one of the greatly used social media platforms, is the most popular and leading platform, where users express their opinions in the form of tweets[19, 20]. Tweet is actually a short message of 140 to 280 characters mostly in informal language and in unstructured form[19]. In this era of big data, when social media is generating the enormous amount of data for each second, it is time-consuming and too difficult to classify such huge amount of data[21]. To achieve more effective and accurate results, the most recent methodologies to automate such text classification tasks[22] are based on NLP (Natural Language Processing)[23,24], supervised Machine Learning[25,26] and deep learning-based approaches[27] such as CNN (Convolutional Neural Network)-based[28], LSTM(Lon-Short Term Memory)[29], Bi-Directional Gated Recurrent Unit[30], Transformer language models[31] etc. The performance of these algorithms heavily depends upon the quality as well as quantity of data. The quality of data denotes prelabelled data, labelled by the humans as well as pre-processing techniques used to prepare the data for training purpose[27].

This study has utilized several Machine Learning algorithms[32] such Naïve Bayes, Random Forest Ensemble Classifier[33], Logistic Regression[12] and K-Nearest Neighbor. The Machine Learning algorithms will be given a textual tweet as input X having an output class Y categorized into two values. These algorithms will predict the tweet from its text that either that particular tweet falls in the category of offensive or non-offensive. The main objectives of proposed research study are: to analyse the hate speech shared against any individual person, any religion, and any ethnic group, to develop Twitter datasets that contain offensive language, and to provide experimental analysis on provided datasets for the significance of proposed technique. Thus, for the automatic identification of hate speech, the novelty and contribution of our paper is: The development of Twitter dataset that consists of several tweets containing 11 object variables with four different class variables showing the different offensive levels, Machine Learning algorithms' application to detect the hate speech, Comparative analysis of different Machine Learning algorithms against different evaluating metrics including McNemar.

Related Works

In literature several studies have proposed different approaches to solve the problem of hate speech detection such as, Ayo et al.[34], has proposed a probabilistic approach for the twitter hate speech classification. A metadata extractor was used to acquire the desired tweets, and the tweets were labelled into two categories, such as hate speech and non-hate speech. While evaluating the generated results, the proposed approach has achieved F1-score of 0.9256. In a study[35] author has conducted a research study during convid-19 when several people were sharing their opinions in the form of hate speech on Twitter. The proposed methodology used deep learning based pre-trained model for multilingual text for English, Chinese and German. After improving the results using data augmentation and cross-lingual contrastive learning, the proposed methodology has achieved the F1-scores of 0.728, 0.799 and 0.612 for English, Chinese and German respectively. Perez et al.[36] has suggested that the addition of contextual information greatly improves the performance in hate speech detection. Under this study, several replies to newspaper related tweets have been collected in the Spanish that mainly provide contextual information. Moreover, transformer-based machine learning approach has been proposed that has achieved 91% accuracy, 75% precision, 65% recall, and 70% F1-score. The research study[37] has proposed hybrid approach of combining NLP with machine learning to detect hate speech from social media platforms. Authors have scrapped online tweets related to specific issue and conducted several experiments after pre-processing the collected data. The generated results show that the highest scores acquired for Accuracy, precision, recall and F1-measure are 98.71%, 98.72%, 98% and 98.3% respectively. Dwivedy and Roy[38] have suggested the multimodal architecture that contains concatenated transfer learning and LSTM (Long Short Term Memory) models for social media posts classification into hate speech and non-hate speech. First, they have focused on text and images to understand the context and then detected hate in the post. While analyzing generated results, the proposed methodology has achieved better results, such as 69% precision, 69% recall, 69% F1-score and 69.04% accuracy. Sahinuc et al.[39] has studied the hate speech detection based on gender in English and Turkish languages.

The size of datasets used was 20k tweets for each language, while different SOTA (state-of-the-art) algorithms were experimented with different setups. The analysis of all the results has shown that the highest scores achieved by proposed methodology experiments are 0.809 precision, 0.806 recall and 0.807 F1-score. Miok et al.[40] has proposed the Bayesian methodology using Monte Carlo dropout in attention layers of transformer for detecting hate speech from several languages. Authors have used three different datasets related to three different languages, English, Croatian and Slovene and results achieved by the proposed methodology are 91% and 90% for the highest accuracy and F1-score achieved respectively. Chiril et al.[41] have advised an approach to capture common properties from hate speech as well as transferring this knowledge from generic topic datasets to particular topic datasets. Various datasets containing different kinds of tweets have been used for several experiments, such as Davidson, Founta, Waseem, AMI Corpora and HatEval. The results evaluation denoted that the multi-task architectures are the best performing models. Stanković and Mladenović[42] have proposed a methodology that either a model trained on the general dataset relevant to social media platform can be effectively tested for binary classification of the hate speech in sports domain. The collected dataset is related to the Serbian language and the deep learning model proposed under this study is the BiLSTM (Bi-directional Long Short-Term Memory) with several parameters. The generated results showed the highest precision score, i.e. 96% and 97% in the sports domain while the achieved recall score is too low. Ganfure[20] has examined several variants of the deep learning algorithms, such as CNN, LSTM, BiLSTM on the collected largest dataset related to the Afaan Oromo (one of the Ethiopian languages). The results evaluation showed that the models dependent on CNN and Bi-LSTM have secured best results with the average F1-score of 87%. García-Díaz et al.[43] have examined the most effective features in hate speech identification in Spanish language as well as how these features can play their role to develop more accurate systems. The combination of linguistic features and transformers by the means of knowledge integration has achieved the best results, 90.4% Accuracy, 88.9% F1-score and 90.2% Macro F1-score, while testing on different datasets. In a study[44] authors have studied, multi-domain hate speech corpus of English language tweets that consists of hate speech against various instances, such as religion, gender nationality, ethnicity etc.

The stacked deep learning-based model consists of CNN, Bi-LSTM and BiGRU (bidirectional gated recurrent unit) was trained on the collected English language tweets dataset. Moreover, the proposed methodology achieved 88.92% Precision, 88.87% Recall, 88.86% F1-score and 88.87% of Accuracy. A framework for hate speech using Machine Learning algorithms have also been proposed[45]. The proposed approach has used Thomas Davidson dataset consists of tweets labelled as offensive but not hate speech, hate speech and neither hate neither speech nor offensive speech. While evaluating the generated results, the SVM (Support Vector Machine) classifier with word2vec+Doc2vec technique has achieved the best scores for accuracy, precision, recall and F1-score in comparison with Random Forest, Logistic Regression and K-Nearest Neighbor. In a study[46] authors have presented a methodology for cyber hate online mainly on Twitter for women. The research study has collected Turkish tweets related to the women clothing and trained five different Machine Learning algorithms, such as SVM classifier, J48, Naïve Bayes, Random Forest and Random Tree. The generated results were evaluated on four metrics such as Accuracy, Precision, Recall and F-measure. While analyzing conducted different experiments, it was observed that SVM Classifier is the only algorithm that has achieved 100% precision in all the experiments. Pronoza et al.[47] have addressed numerous problems faced while targeting different ethnic groups in Russian language. The proposed methodology has used a dataset of size 2.4M user messages regarding ethnic groups and experimented several Machine Learning algorithms and Deep Learning algorithms. The finetuned and pre-trained RuBERT along with linguistic features, outperformed with 0.813 F1-hate and 0.833 F1-macro scores.

In the previous studies, most of the research studies have mainly focused on only two class variables, such as offensive or non-offensive showing the hate speech severity instead of focusing on different levels of hate speech. Meanwhile, previous research studies have not emphasized on the targeted gender, religion or sexual orientation simultaneously in the similar dataset. Moreover, none of the research studies have used any data imbalance technique such as SMOTE to balance the dataset records that greatly impact the consequent results of supervised learning models. However, this study have more focused on the different dataset's object features that contain user's ID, name, location, gender, sexual orientation, disability, target gender, target religion, target ethnicity, tweet's text and class variables. Additionally, this paper has also used data imbalance technique to balance the dataset records that ultimately impact the results of supervised learning models.

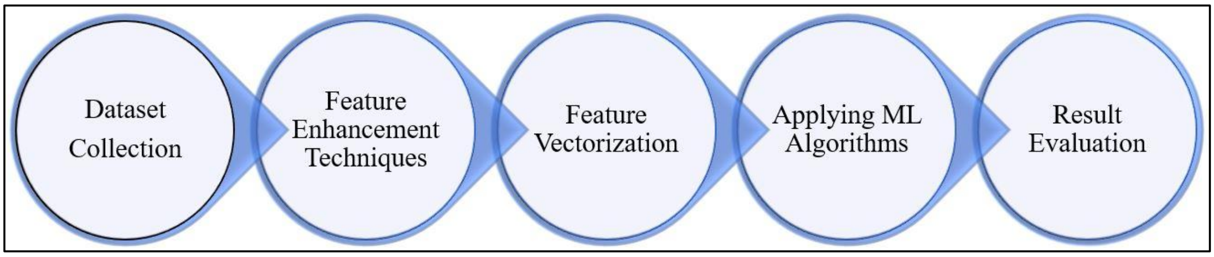


Figure 1. Overall scheme for designed methodology

Proposed Methodology

The methodology designed for proposed approach is illustrated in Fig. 1. It contains different steps such as data collection, feature enhancement techniques, feature vectorization, applying machine learning algorithms followed by results evaluation. Moreover, this section is divided into different subsections such as section 3.1 describes dataset collection and preparation that is further subdivided into feature engineering and extraction, label encoding, data preprocessing, data cleaning, data separation, count vectorization, handling imbalanced data, data splitting, and applying Tf-Idf transformer, section B explains machine learning algorithms such as random forest ensemble classifier, naïve bayes classifier, linear Support Vector Machine (SVM), and logistic regression. Additionally, the evaluation of proposed methodology is further discussed in section 4.

Dataset Collection and Preparation

The dataset used in this study was collected from Twitter and focused on hate speech in the English language. The data was extracted from Twitter using Tweepy library of Python and Twitter API (Application Programming Interface). The dataset contains 28,211 rows and 11 columns, consisting of 10 object features and 1 numerical feature. The count of dataset is about 2821111 and after taking out duplicate rows from dataset, this number reduced to around 2815311. More importantly, the datasets used in this approach do not contain null values thus, this this can be an excellent idea for the experimental analysis that involves machine learning algorithms. The dataset contains information related to hate speech on Twitter (a social media platform). This dataset's object features contain user's ID, name, location, gender, sexual orientation, disability, target gender, target religion, target ethnicity, tweet's text and class variables. Also, the feature given in number form indicates the retweet of every tweet. The target gender and target religion were determined by extracting different words according to the object feature.

The gender was identified from the tweet by extracting words, such as he, his, she, her etc. that clearly shows the target gender, else the targeted gender was both male and female. Meanwhile, the target religion was identified by extracting words, like MiddleEast, bogan, chuslim etc. that determines the target religion in certain tweet. The given dataset is well-organized that makes it easy for analyzing and extracting valuable information. Moreover, this dataset can be a useful asset for the researchers that are working in the research directions related to extracting hate speech contents in social media. The absence of null values and well-structured format makes this dataset an invaluable resource for the NLP projects. Additionally, this dataset can be used to train the machine learning models for the detection of trending hate speech on Twitter. In this research paper, the response variable is “Class” feature consisting of four different values such as: “Extremely offensive”, “Mildly offensive”, “Highly offensive”, and “Moderately offensive”. Here the division of given values in this dataset is: “Extremely offensive” with 12,222 values of occurrences, “Mildly offensive” with 6,174, “Highly offensive” with 5,667, and “Moderately offensive” with 4,090 values of occurrences. More importantly, there is imbalance distribution is data as “Extremely offensive” is the most and “Moderately offensive” is the least frequent class. In this study the “Class” feature helps to categorize and detect the intensity of hate speech in twitter datasets. This study can investigate and interpret the language used in every class through the classification of tweets into four different categories. Moreover, in the “Class” feature imbalanced distribution of values causes a challenge for machine learning models, as these values can be biased to the class that most frequently occurs.

Thus, it is very difficult to handle the imbalanced data through the training and evaluation of machine learning model. The classification of “Class” feature offers the frequency of hate speech in dataset. The most occurrence “Extremely offensive” throughout the dataset indicates the most severity of hate speech in Twitter dataset. Additionally, the classification of values can help to recognize different attributes of users (age, nation, race, religion etc.) who are involved in the hate speech. The values in “Class” feature have been converted into “Highly offensive” and “Mildly offensive” to make classification simpler and decrease the classes' number. After converting into mentioned two categories, “Highly offensive” is with 17,889 and “Mildly offensive” is with 10,264 numbers of occurrences. The decision to convert original values into two categories was based on severity of hate speech language used in tweets. The two categories represent a clear distinction between tweets that contain highly offensive language and those that contain mildly offensive language. By simplifying classification process, research study can develop more straightforward and efficient machine learning models. The distribution of converted “Class” feature provides insights into prevalence of highly offensive and mildly offensive hate speech on Twitter.

The high occurrence of tweets classified as “Highly offensive” suggests that Twitter users engage in use of extremely harmful language, and this can have severe consequences on individuals and society. The distribution of values can also be utilised to uncover patterns and trends in language that is used in tweets that are considered to constitute hate speech. In addition to analyzing “Class” feature, study also visualized and analyzed distribution of values of other features in dataset. One of the features that this study examined was the “User Gender” feature. The distribution of values of this feature shows that out of total number of users whose tweets were collected, 15,864 identified as male and 12,289 identified as female. Visualizing the distribution of “User Gender” feature provides insights into gender distribution of Twitter users who engage in hate speech. This information can be used to understand how different groups use language in online spaces and to identify patterns in the behaviour of users who engage in hate speech. Another feature that research analyzed was “Target Gender” feature, which indicates the gender of person or group targeted by hate speech in tweet. The classification this feature values determines that in the given dataset 16,970 tweets targeted male and female, 9,247 targeted only female while 1,936 targeted male users. The classification of "Target Gender" gives the understanding of gender classification in hate the speech. This information can be used to understand how different genders are affected by hate speech and to identify patterns in the behaviour of users who engage in hate speech. The "Sexual Orientation" feature in dataset indicates sexual orientation of person or group targeted by hate speech in tweet. The distribution of values of this feature shows that out of total number of tweets in the dataset, 26,970 tweets targeted individuals or groups with all sexual orientations. However, there were 795 tweets that specifically targeted homosexual individuals, and 388 tweets targeted heterosexual individuals. Visualizing the distribution of "Sexual Orientation" feature provides insights into prevalence of hate speech directed towards individuals with different sexual orientations on Twitter.

Understanding the distribution of hate speech towards individuals with different sexual orientations is crucial in promoting a more inclusive and tolerant society. By analyzing distribution of values of "Sexual Orientation" feature, this study can identify patterns in behaviour of users who engage in hate speech towards specific groups and work towards addressing the underlying factors that contribute to such behaviour. The "Target Religion" feature in dataset indicates religion of person or group targeted by hate speech in tweet. The distribution of values of this feature shows that out of total number of tweets in dataset, 9,567 tweets targeted individuals or groups with all religions. However, there were 7,637 tweets that specifically targeted individuals or groups who identified as Christian, making it the most targeted religion in dataset. Islam was the second most targeted religion with 4,406 tweets, followed by Buddhism with 2,439 tweets, Catholicism with 1,897 tweets, Judaism with 1,374 tweets, and Muslims with 833 tweets. Visualizing the distribution of "Target Religion" feature provides insights into prevalence of hate speech directed towards individuals with different religious backgrounds on Twitter. The distribution of "Target Ethnicity" values provides valuable insights into diverse ethnic makeup of population under consideration. Among 7625 individuals, the most common ethnicity is American, with 5092 individuals, followed closely by Muslim at 3700. Chinese and Aboriginal ethnicities also represent a significant portion of population, with 2158 and 2087 individuals respectively. This data further reveals the representation of other ethnicities such as Jews, Irish, Hispanic, British, Arab, Italian, Amish, Asia, Korean, and African.

The distribution of "Disability" values within population under consideration reveals important insights into prevalence of disabilities among individuals. This data also indicates that from 28,153 users, 26,878 users have no any disability and 1,275 people are physically challenged. The classification of disability problem is very important to understand especially for the policy makers and organizations that work for the development of people who are facing some physical challenges. Note that, the people with physical challenges face a lot of problems in the society. After understanding and visualizing the data, the second step is processing. It includes a number of techniques such that cleaning, transformation, and normalization. In the cleaning process, correction of errors, removal of similar and unimportant data and solving the missing entries are carried out. While the transformation and normalization involve the modification of data into the format that is used for analysis and to make sure that whole the data is on same scale. These three steps make sure a better-quality data that can be used for analysis purpose. In addition to this, the proper use of data processing techniques can reduce the effect of outliers that affects the accuracy. That is why, it is essential to focus more on the data processing step that validates the reliability of final analyzed results.

Feature Engineering and Extraction

The feature engineering is the process to select, manipulate and transform the given input data based on the domain knowledge into the features that can be utilized for both supervised and unsupervised learning models. To improve or generate good results, it is necessary to select the most important features. The feature engineering process consists of four steps, such as feature creation, feature transformation, feature extraction and feature selection. In the feature creation step, the most significant variables are identified that will be highly useful for the model to perform the predictions. In this work, the identified significant variables chosen for the proposed algorithms are the Tweet Text containing the actual text and the Class variable

Published Online First: November, 2024
https://dx.doi.org/10.21123/bsj.2024.10743
P-ISSN: 2078-8665 - E-ISSN: 2411-7986
Baghdad Science Journal

showing the offensive levels according to the Tweet text. After identifying the important variables, the variables will be manipulated to ensure that all the features are in the acceptable range in the feature transformation step. In the feature extraction step, the new features will be created from the raw data, if necessary, that will ultimately reduce the data size for more manageable dataset. Lastly, the irrelevant or redundant features will be discarded and the most important feature variables will be prioritized for the model. The illustration of feature engineering and extraction is given in the Fig. 2.

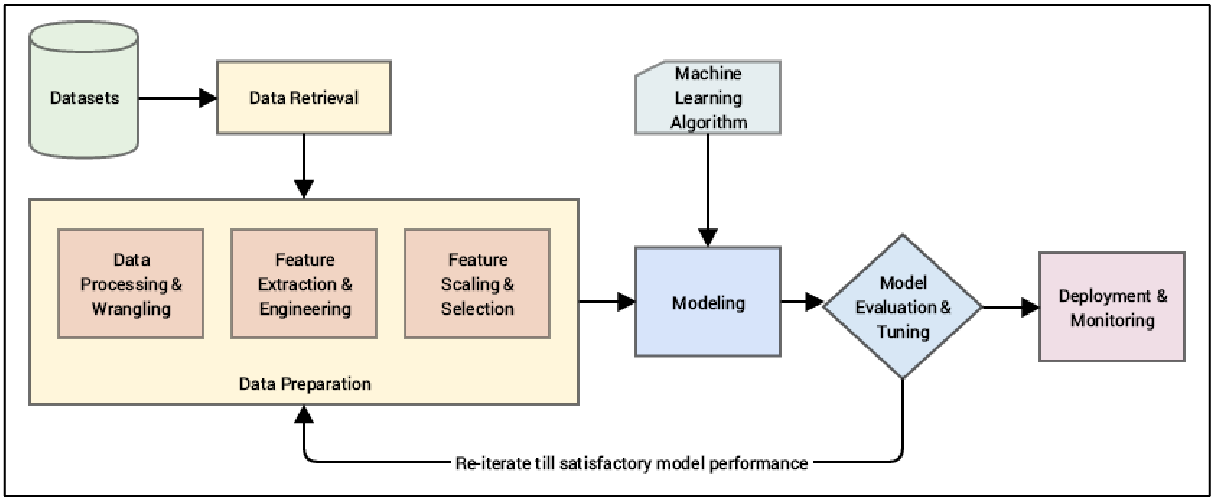
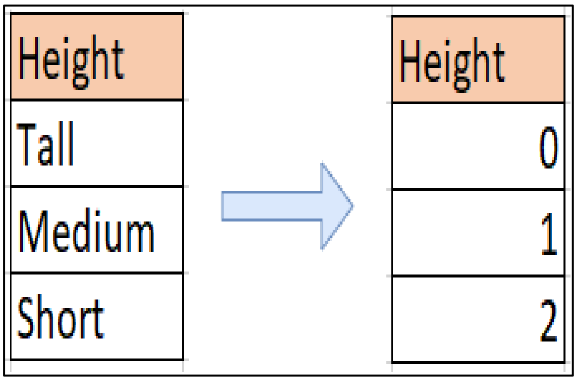

Datasets
Data Retrieval
Machine Learning Algorithm
Data Processing & Wrangling
Feature Extraction & Engineering
Feature Scaling & Selection
Data Preparation
Modeling
Model Evaluation & Tuning
Deployment & Monitoring
Re-iterate till satisfactory model performance

Figure 2. Illustration of feature engineering and extraction

Label Encoding

Machine Learning, the datasets mostly are consisted of different categorical values, such as high, low, medium etc. The label encoding is used to convert these categorical values into numerical before processing by the Machine Learning algorithms as given in Fig. 3, because these algorithms only accept the numerical values instead of the textual data.



Height
Tall
Medium
Short

Height
0
1
2

Figure 3. Label encoding

The label encoding is applicable, when a particular model does not accept any categorical values, such as classification. In Python, the sklearn library contains the class, Label Encoder that is used for the label encoding to convert the strings numerical data. The label encoding converts strings into numerical data, while assigning each string value a unique number starting from 0. In this way, the model may consider more priority for the label having highest value. In our research study, using Label Encoder library, the class variables such as mildly offensive variable is assigned 0, moderately offensive variable is assigned 1, highly offensive variable is assigned 2 and extremely offensive variable is assigned 3 values to prepare the dataset for the supervised learning models, as in Fig. 4.

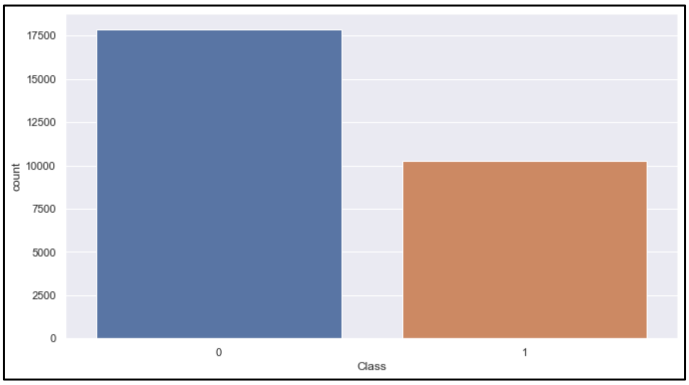


Figure 4. Number of Tweets for both class variables

Data Preprocessing

Data preprocessing is a critical stage in the development of any machine learning model, as the accuracy and effectiveness of the model heavily rely on the quality and cleanliness of the input data. Text preprocessing is a crucial first step in the overall process of NLP, involving transforming raw text input into a format more appropriate for analysis and modelling. Text preprocessing typically involves several steps, such as tokenization, stemming, stop-word removal, and normalization, which aim to transform the text into a more structured and manageable format. To capture the fundamental meaning and eliminate redundancy, the text is tokenized by separating it into individual words, or tokens, and by reducing every word to its root form. Both stop-word removal and normalization try to get rid of meaningless filler words like “the”, “and”, and “a”, whereas normalization entails changing every word to lowercase to prevent duplication. The success of a machine learning model in an NLP task is largely dependent on how well the text data has been pre-processed. If the data is not pre-processed effectively, the resulting model may suffer from poor accuracy, difficulty in generalization, and performance issues. Thus, it is crucial to carefully evaluate and optimize the text preprocessing techniques used in the development of an NLP model.

Data Cleaning

The data cleaning procedure is an important part of the data preprocessing phase, particularly in NLP applications where the raw text data may contain irrelevant or noisy information. In this study, the “Tweet Text” feature was the focus of data cleaning. To this end, several functions were defined to pre-process and clean the text data. The first function applied was the “clean” function, which removed greater than signs, apostrophes, paragraph tags, italic tags, and new lines from the data. The second function was responsible for the removal of emoji’s, including emoticons, symbols, and pictographs, transport and map symbols, and flags, which could potentially add noise to the data. The third function involved the removal of all punctuation marks from the data. Next, the text data was converted from upper-case to lower-case to avoid duplication and ensure consistency. Stop words, such as “the”, “and”, and “a”, that do not carry significant meaning, were removed from the data, as well as the words “USER” and “RT” which are often present in tweets but do not carry relevant information. Finally, lemmatization was applied to normalize the data, by reducing each word to its base or dictionary form. After following the whole process, the resultant data will be meaningful as well as in an organized way that is very important for designing the more effective Machine Learning model. While selecting the data attributes for building the model, the most significant column values need to be chosen. Our dataset also contains some irrelevant attributes, such as Twitter ID, Username and its location that have no any impact on our data. Therefore, all the redundant or inappropriate columns need to be discarded to have more accurate results. Consequently, while applying all procedures, the quality of our data will get improved that will greatly contribute in extracting the desired insights from the data.

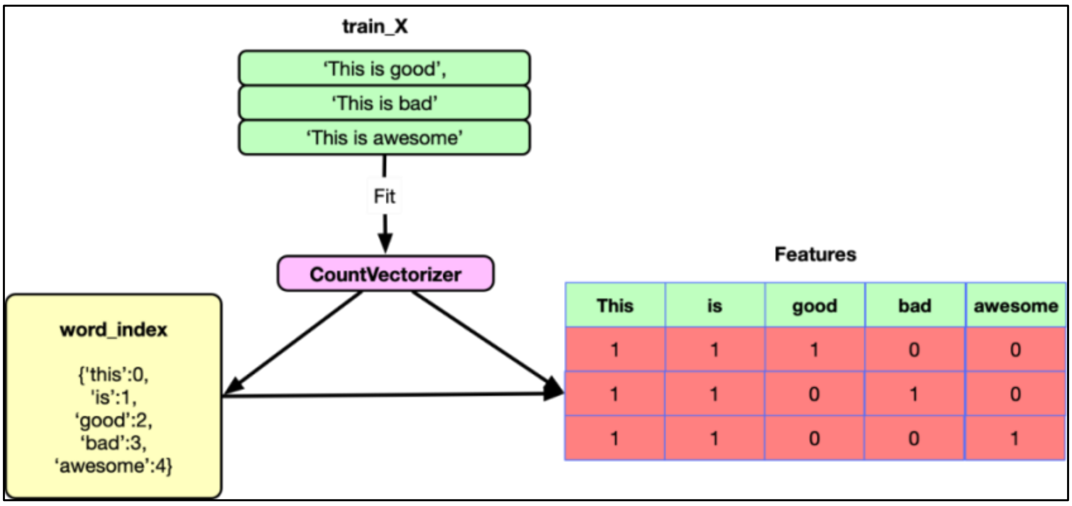
Data Separation

After removing the irrelevant data, there have two column values, such as target variable either 0 or 1 denoted by y and the actual tweet text that will be used to discover the hate speech represented by the x variable. This separation of variables is necessary, because the Machine Learning models accept target variable as the predicting class variable on the basis of the tweets’ textual data. As soon as the separation of x and y variables is completed, the next process of training, testing and evaluating the model may be initiated. The model will be feed two different variables that will be used for the predictions accordingly.

Count Vectorization

The Machine Learning algorithms cannot be trained directly on the normal textual data. The textual data needed to transform in the required structure workable for these algorithms[17]. Every collected sentence consists of different words needs to be converted into a high- dimensional vector space. Consequently, there will be a matrix containing several vectors for each sentence, while column size will be according to the vocabulary of words. The Count vectorization, applied in this study, generates a sentence vectorization while keeping the number of any word’s occurrences and zero for a word not appearing in a sentence. Fig. 5 illustrates the complete process of the count vectorization technique that generates the dictionary of the appearing words and accordingly keeping the number of occurrences for each word in a sentence.

In our research study, this research study has utilized the count vectorization for converting the text from each tweet. Overall, the number of extracted features in our study is 8000 that will be ultimately the dimension of the generated vector space. After generating the matrix using count vectorization, the Machine Learning algorithm can give this generated matrix for the training purpose.



This	is	good	bad	awesome
1	1	1	0	0
1	1	0	1	0
1	1	0	0	1

Figure 5. Illustration of Count Vectorization

Handling Imbalanced Data

The imbalance data refers to the dataset having higher number of one class label in comparison with other class labels. In our dataset, this study observed an imbalance in the proportion of tweets labelled as positive and negative. Specifically, the proportion of tweets labelled as negative was much smaller than the proportion labelled as positive. Because of this imbalance, the machine learning algorithm may have a preference for the class that constitutes the majority, which might lead to the inaccurate classification of members of the minority class. To address this issue, this study employed two techniques: SMOTE (Synthetic Minority Oversampling Technique) and Random under Sampling. Oversampling methods like SMOTE create fictitious data sets for underrepresented groups. Interpolating between close-together positive instances in the feature space is an attempt to circumvent the overfitting issue that comes from random oversampling. Instead, random under-sampling removes instances at random from the majority class from the data used for training unless a more equitable distribution is reached. The process of handling data imbalance is given in Fig. 6. To implement SMOTE and Random under sampling techniques simultaneously, the pipeline method was utilized that combines both approaches with different parameters. Resultantly, the imbalanced data problem was overcome as well as equal number of tweets for both positive and negative classes. The balanced data always plays a vital role to enhance the model's accuracy as well as generates better results.

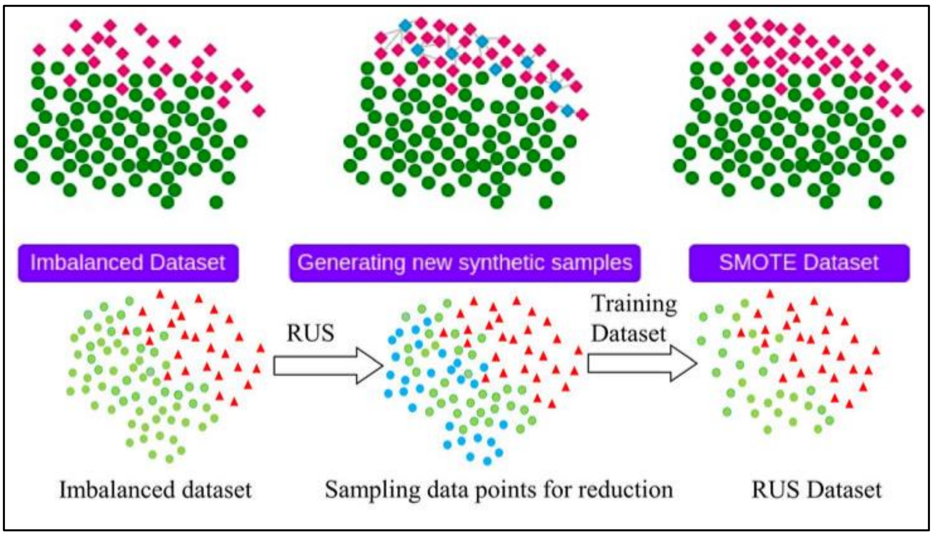


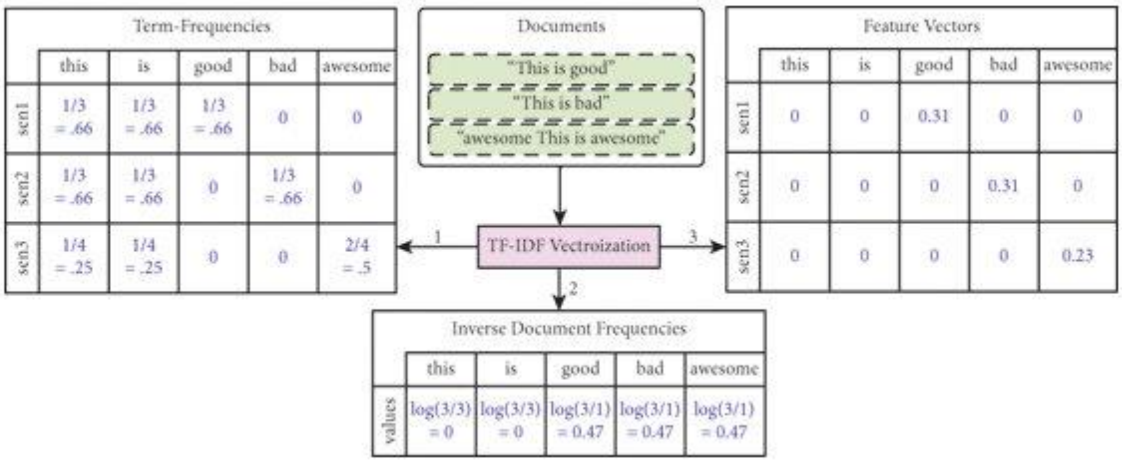
Figure 6. Handling Data Imbalance

Data Splitting

In the data splitting step, the dataset is either divided into two or three subsets that the model can be trained, tested and evaluated. If the dataset is split into two parts, both subsets will be used for the training and testing processes. In the three splits of data, the training, testing and validating process will be carried out according to three parts of data. These have split our dataset into three subsets and have utilized the 10-fold cross validation technique. The k-fold cross validation technique follows resampling procedure that is employed to evaluate the model design instead of training process. The k value in the k fold cross validation denotes the number of groups that data will be split into. The benefit of using k-fold cross validation is to overcome the overfitting problem that sometimes occur and badly degrades the final results.

Applying Tf-Idf Transformer

The Term frequency and Inverse document frequency (TF-IDF) technique mainly focuses upon the significance of a word in a given document[48]. In other words, it determines the trade-off between the highly frequent words and the less-frequent words[17]. The TF-IDF transformer has been used for information retrieval, sentiment analysis and data mining related applications[48]. The term frequency calculates the frequency of a certain term relative to the whole document, while inverse document frequency is concerned with the how common or uncommon a word is appearing among the corpus. The Eq. 1 shows the mathematical calculation of term frequency, where t denotes the number of times a certain term appears in a document, and d refers to the document. The Eq. 2 demonstrates the inverse document frequency mathematically, where N represents the total number of documents and df signifies the document frequency of a term. The Eq. 3 illustrates the whole measure of TF-IDF by multiplying the term frequency by the inverse document frequency.



"This is good"
"This is bad"
"awesome This is awesome"

Term-Frequencies

	this	is	good	bad	awesome
sen1	1/3 = .66	1/3 = .66	1/3 = .66	0	0
sen2	1/3 = .66	1/3 = .66	0	1/3 = .66	0
sen3	1/4 = .25	1/4 = .25	0	0	2/4 = .5

Inverse Document Frequencies

	this	is	good	bad	awesome
values	log(3/3) = 0	log(3/3) = 0	log(3/1) = 0.47	log(3/1) = 0.47	log(3/1) = 0.47

Feature Vectors

	this	is	good	bad	awesome
sen1	0	0	0.31	0	0
sen2	0	0	0	0.31	0
sen3	0	0	0	0	0.23

Figure 7. Illustration of TF * IDF Vectorization

$$TF\ (t,d) = \frac{number\ of\ times\ t\ appears\ in\ d}{total\ number\ of\ terms\ in\ d} \tag{1}$$

$$IDF(t) = log\,\frac{N}{1+df} \tag{2}$$

$$TF - IDF(t,d) = TF(t,d) \times IDF(t) \tag{3}$$

Fig. 7 explains the complete process of feature vector generation by using TF-IDF vectorization technique. The TF-IDF vectorization is very popular for the transformation of raw text into vectors mainly for the Machine Learning algorithms.

Machine Learning Algorithms:

Random Forest Ensemble Classifier

The Random Forest is an ensemble of Decision Trees that is more convenient and optimized for the Decision Trees. The Random Forest classifier generates the output results by creating different decision trees that all trees will operate as ensemble. All the trees will predict the given data and the class with highest number of votes will be considered as the prediction by model. This classifier has ability to avoid overfitting, while training the model. While growing trees, it introduces extra randomness. The Random Forest can be used for classification and regression. This study have used Random Forest algorithm for classification. Fig. 8 shows that how Random Forest Classifier works and output the final results from all the decision trees.

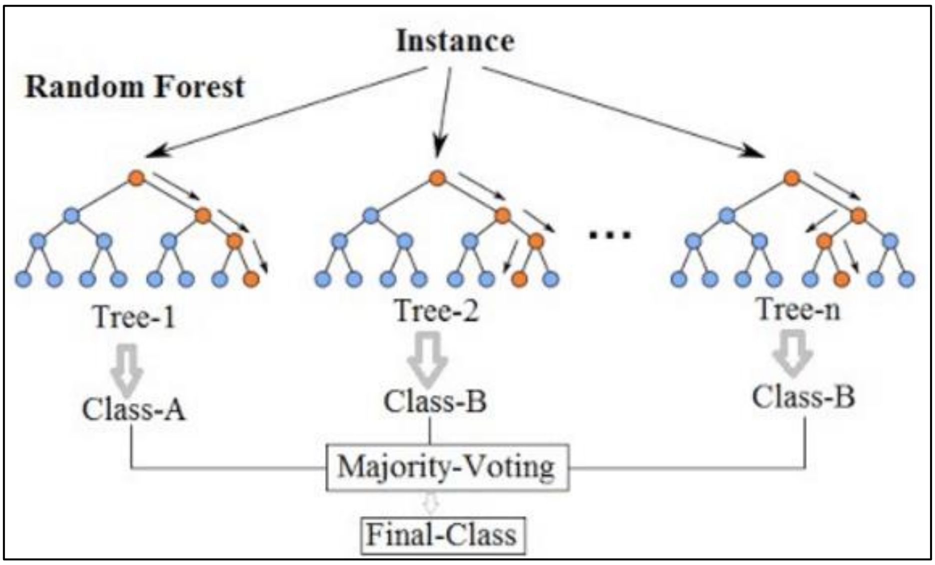


Figure 8. Random Forest Trees Illustration

Naïve Bayes Classifier

Naïve Bayes is the classification algorithm that belongs to supervised learning. The Naïve Bayes is mainly used for text classification. It classifies the sentences by exploiting conditional probability using Bayes’ equation with the assumption that each feature is independent and equal. In the Eq. 4
4 Bayes’ theorem formula is stated mathematically

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \qquad 4$$

In the Bayes’ equation, there are two events, such as A and B. using this equation, Naïve Bayes classifier finds the probability of event A given that the event B is true.

Linear Support Vector Machine (SVM) Classifier

Support Vector Machine is a classifier that represents the data as points in space categorically. These data points are separated by a hyperplane by a clear margin to denote each class as separate class. It is well suited for the classification of complex but should be small or medium sized datasets. In this research, this study has used Linear SVM classification. Fig. 9 shows the data points plotted by using SVM classifier. Maximizing the margin between data points provides reinforcement that the future data points will be classified with more confidence.

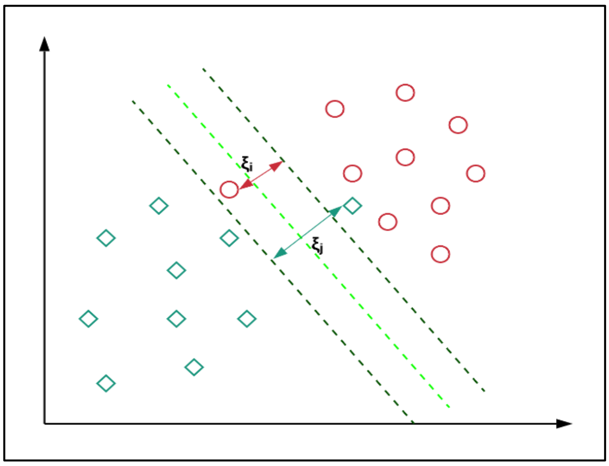


Figure 9. Linear SVM Illustration

Logistic Regression

The Linear Regression is highly used for Text Classification to solve Classification problems on large-scale. The Logistic Regression, also known as Logit Regression, is the probabilistic algorithm that measures the probability of an instance to whom, it belongs. The Logistic Regression is mainly used for document classification and NLP. The Logistic Regression is one of those algorithms that can be used for Regression as well as for Classification. In the Eq. 5, the Linear Regression algorithm is discussed with the help of a function to measure the probability for each instance.

$$f(x) = \frac{1}{1 + e^{-(x)}} \qquad 5$$

Published Online First: November, 2024
https://dx.doi.org/10.21123/bsj.2024.10743
P-ISSN: 2078-8665 - E-ISSN: 2411-7986
Baghdad Science Journal

Results and Discussion

In this section, all the Machine Learning algorithms, such as Naïve Bayes, Random Forest Ensemble classifier, Linear Support Vector Machine classifier and Linear Regression results are evaluated based on different metrics. Moreover, the comparison of our proposed approach with other SOTA (Stateof-the-Art) approaches is also explained in Table 1.

Evaluation Criteria and Metrics

To evaluate the performance of any machine learning algorithm, there are various metrics to judge the performance of any implemented model. This research study is mainly focusing classification problem, therefore, the evaluating metrics that may be used are Accuracy, Precision, Recall and F1-score that can be extracted from confusion matrix. As discussed in the Fig. 10, each of the cells in the confusion matrix is representing a factor for evaluation, such as True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN). The True Positive (TP) factor determines how many positive samples were predicted correctly by the model, True Negative (TN) denotes the number of negative samples correctly predicted, False Positive (FP) signifies the number of positive class samples predicted incorrectly an the False Negative (FN) represents the number of negative samples predicted incorrectly by the trained model.

Table 1. Comparison of Proposed Approach with other SOTA (State-of-the-Art) Approaches

Paper	Dataset	Method	Accuracy	Precision	Recall	F1-score
Ayo et.al [34]	Twitter	-	-	-	-	92.5%
Liuet.al [35]	Twitter	-	-	-	-	79.9%
Perez et.al [36]	Twitter	Transformer based Machine Learning approach	91%	75%	65%	70%
Makhadmeh et.al [37]	Twitter	NLP with Machine Learning	98.71%	98.72%	98%	98.3%
Dwivedy et.al [38]	Social media text + images	Transfer learning and LSTM	69.04%	69%	69%	69%
Sahinuc et.al [39]	Twitter (20k)	-	-	80.9%	80.6%	80.7%
Miok et.al [40]	English, Croatian and Slovene	Transformer with attention layer	91%	-	-	90%
Chiril et.al [41]	Davidson, Founta, Waseem, AMI Corpora	-	-	-	-	-
Stankovic et.al [42]	Serbian language	Bi-LSTM	-	97%	-	-
Ganfure et.al [20]	Afaan Oromo	CNN, LSTM, Bi-LSTM	-	-	-	87%
PSharmila et.al [19]	Twitter	RF + TF-IDF	98.9%	98.8%	98.6%	98.5%

The accuracy is the total number of correct predictions (sum of TP and TN) divided by the total number of predictions (sum of TP, TF, FP and FN). The accuracy metric is used to determine the best performing model among several models while recognizing the existing relationship between the variables. One popular way to evaluate a classification model's efficacy is through the use of an accuracy score. It is the fraction of the dataset's samples that have been correctly labelled. On the other hand, the ROC-AUC score evaluates the capacity of the models this study use to differentiate among samples that are both positive and negative. If the ROC-AUC score is higher, this suggests that the model is doing better. The F1-Score is an essential metric that this paper uses to gauge the overall efficacy of our models. It is calculated by taking the weighted average of the precision and recall measures. Precision is the ratio of the number of real positives to the overall amount of anticipated positives, whereas recall is the ratio of the number of real positives to the number of genuine positives. A high F1-score shows that the model is accurate in both its predictions and its recall of data, as in Eqs. 6-9.

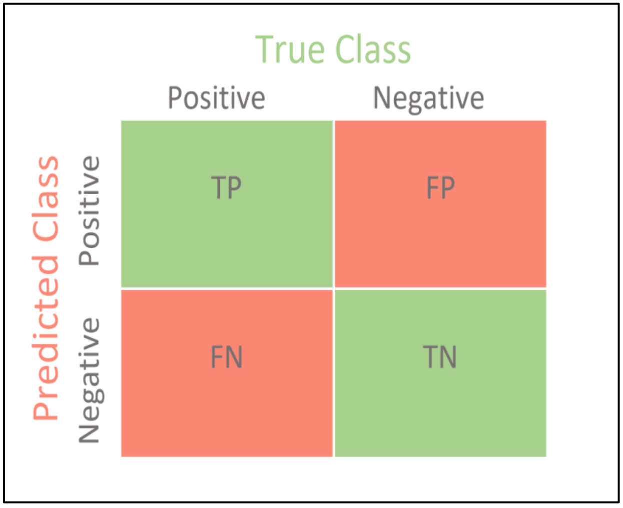
$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \qquad 6$$

$$Precision = \frac{TP}{TP + FP} \qquad 7$$

$$Recall = \frac{TP}{TP + FN} \qquad 8$$

$$F1 - Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \qquad 9$$

The F1-Score is an essential metric that this research study uses to gauge the overall efficacy of our models. It is calculated by taking the weighted average of the precision and recall measures. Precision is the ratio of the number of real positives to the overall amount of anticipated positives, whereas recall is the ratio of the number of real positives to the number of genuine positives. A high F1-score shows that the model is accurate in both its predictions and its recall of data. Overall, by extracting these metrics from each method, this study is able to acquire a more in-depth comprehension of the performance of our models and make use of the knowledge obtained to fine-tune their parameters and optimize their overall performance for a variety of classification jobs.



		True Class	
		Positive	Negative
Predicted Class	Positive	TP	FP
	Negative	FN	TN

Figure 10. Confusion Matrix Illustration

Analysis of Machine Learning Models and Results

The Random Forest model outperformed other models in terms of accuracy and F1-score. This suggests that the model is effective at accurately classifying instances in real world scenarios. The Random Forest classifier aggregates the predictions from various classifiers and consequently, predicts the class that achieved maximum votes. The ensemble classifiers outperform when the predictors are independent from each other. The generated results will also lead towards more effective as well more efficient applications for future in the area of machine learning. There are several reasons behind achieving such best scores, this result comparison is also given in Table 2.

Table 2. Results Comparison Table in Term of Accuracy and F1-Score

Models	Train Accuracy	Train F1-Score	Test Accuracy	Test F1-Score
Random Forest	100.00%	0.9999	97.85%	0.9779
Linear Support Vector Classifier	94.93%	0.9486	93.99%	0.9389
Logistic Regression	90.06%	0.8963	89.26%	0.8873
Multinomial Naïve Bayes	93.30%	0.9343	91.17%	0.9134

McNemar Test

McNemar test, also known as paired chi-squared test is used to analyze the significant change in the two correlated samples[49]. In other words, McNemar test determines the marginal homogeneity in the rows and columns from a table. It is a matching pair test for 2 * 2 tables, in this regard; both categorical variables must not be independent but correlated with each other. In the Table 3, three different statistical computations are performed for three different models, such Random Forest (RF) vs. Linear Support Vector Machine Classifier (LSVC), Random Forest (RF) vs. Linear Regression (LR) and Random Forest (RF) vs. Multinomial Naïve Bayes Classifier (MNB). While analyzing the computed McNemar test results, it is observed that classifiers, Linear Support Vector Machine classifier and Multinomial Naïve Bayes classifier have not generated satisfactory results. The Random Forest ensemble classifier has again acquired the best results compared to other Machine Learning models. Based on the results of the accuracy and F1-score tests, it appears that the Random Forest model outperformed the Multinomial Naïve Bayes model. The use of McNemar's test in comparing the performance of different models can aid in model selection and improve the accuracy of predictive models. Further research can be conducted to explore the effectiveness of McNemar's test on different datasets and compare its performance with other statistical tests. These results are important for the area of machine learning and may make it easier and more accurate to make models that can predict the future. Actually, the conclusion is the opposite of what you have stated. Because the p-values for each of the three tests came in lower than the predetermined threshold of 0.01, it is reasonable to conclude that the null hypothesis ought to be rejected.

This indicates that the results of the models are significantly distinct from one another and that they do not produce results that are on par with one another when applied to this dataset. As a result, the Linear Support Vector Classification model, Logistic Regression model, and Multinomial Naïve Bayes models were all outperformed by the Random Forest model in terms of accuracy and F1-score. However, the Random Forest model still performed the best overall. Based on these findings, it seems that the Random Forest algorithm is more trustworthy in making predictions for the target variable when using this specific dataset. The application of McNemar's test gives a dependable and statistically sound approach for analysing the performance of several machine learning models, which can be of assistance in selecting the model that will do the given task in the most effective possible manner.

Table 3. MCNEMAR test comparison

Models	Alpha Value	P-Value	Statistics
RF v LSVC	0.01	0.001	10.571
RF v LR	0.01	0.000	52.526
RF v MNB	0.01	0.004	8.191

Actually, the conclusion is the opposite of what you have stated. Because the p-values for each of the three tests came in lower than the predetermined threshold of 0.01, it is reasonable to conclude that the null hypothesis ought to be rejected. This indicates that the results of the models are significantly distinct from one another and that they do not produce results that are on par with one another when applied to this dataset. As a result, the Linear Support Vector Classification model, Logistic Regression model, and Multinomial Naive Bayes models were all outperformed by the Random Forest model in terms of accuracy and F1-score. However, the Random Forest model still performed the best overall. Based on these findings, it seems that the Random Forest algorithm is more trustworthy in making predictions for the target variable when using this specific dataset. The application of McNemar's test gives a dependable and statistically sound approach for analysing the performance of several machine learning models, which can be of assistance in selecting the model that will do the given task in the most effective manner possible.

Conclusion

The detection of hate speech shared on social media platforms is a significant task that has greatly affected the individuals' mental health as well as different religions and ethnic groups. There has been various research studies conducted while utilizing Machine Learning and Deep Learning algorithms. Through this study, this study proposed a methodology that classified the tweets into four different classes, highly offensive, mildly offensive, moderately offensive and extremely offensive. In this regard, this study conducted experiments by applying different Machine Learning algorithms on the collected dataset containing millions of tweets. Before applying the proposed models, the data was balanced using SMOTE technique that consequently made the equal distribution for all class values.

The generated results evaluation showed that the Random Forest Ensemble Classifier has outperformed as compared to other algorithms such as Naïve Bayes, SVM and Logistic Regression. The McNemar test results also emphasized that the Random Forest Ensemble Classifier has outperformed overall against all other algorithms. The novelty and contribution of our paper is the Twitter datasets development that consists of various tweets containing 11 object variables with four different class variables showing the different offensive levels, Machine Learning algorithms' application to detect the hate speech, and the comparative analysis of different Machine Learning algorithms against different evaluating metrics including McNemar Test. The limitation of our study is that the proposed methodology may not be impactful for the binary classification having only two classes, offensive and non-offensive. The achieved performance may be degraded when the dataset is not balanced having equal number of all class variables' records.

Authors' Declaration

- Conflicts of Interest: None.
- We hereby confirm that all the Figures and Tables in the manuscript are ours. Furthermore, any Figures and images, that are not ours, have been included with the necessary permission for re-publication, which is attached to the manuscript.
- No animal studies are present in the manuscript.
- No human studies are present in the manuscript.
- Ethical Clearance: The project was approved by the local ethical committee at University Technology Malaysia.

Authors' Contribution

J. A. S. conceptualization, methodology, software, coding, visualization, formal analysis, writing, original draft, and writing, review and editing. S. S. Y. supervision and guidance in conceptualization, methodology, formal analysis, coding, writing and editing. Z. A. M. review and editing, suggestions, investigation, formal analysis, and validation.

References

1. Mohammed A, Haider D, Widad K. Fake News Detection Model Basing on Machine Learning Algorithms. Baghdad Sci J. 2024; 21(2): 150-162. https://doi.org/10.21123/bsj.2024.8710
2. M. U. S. Khan, A. Abbas, A. Rehman, R. Nawaz. Hate classify: a service framework for hate speech identification on social media. IEEE Internet Comput. 2021; 25: 40-49. https://doi.org/10.1109/mic.2020.3037034.
3. Ayo F E, Folorunso O, Ibharalu F T, Osinuga I A. Machine learning techniques for hate speech classification of twitter data: State-of-the-art, future challenges and research directions. Comput. Sci. Rev..2020; 38: 100311. https://doi.org/10.1016/j.cosrev.2020.100311.
4. Khan S, Ullah F, Alquhayz H, Imran M, Mehmood A, Ahmad M, et al. HCovBi-Caps: Hate speech detection using convolutional and bi-directional gated recurrent unit with capsule network. IEEE Access. 2022; 10: 7881-94. https://doi.org/10.1109/ACCESS.2022.3143799.
5. Valentina I, Juhaida A, Nor Hazlyna H, Alaa F. A word cloud model based on hate speech in an online social media environment. Baghdad Sci J. 2021; 18(2 Suppl): 0937-page??. https://doi.org/10.21123/bsj.2021.18.2.0937.
6. Fortuna P, Soler-Company J, Wanner L. How well do hate speech, toxicity, abusive and offensive language classification models generalize across datasets? Inf Process Manag. 2021; 58(3): 102524. https://doi.org/10.1016/j.ipm.2021.102524.
7. Awal MR, Lee RK, Tanwar E, Garg T, Chakraborty T. Model-agnostic meta-learning for multilingual hate

speech detection. IEEE Trans Comput Soc Syst. 2023; 1-10. https://doi.org/10.1109/TCSS.2023.10100717.

8. Agarwal S, Chowdary CR. Combating hate speech using an adaptive ensemble learning model with a case study on COVID-19. Expert Syst Appl. 2021; 185: 115632.

9. Yin W, Zubiaga A. Towards generalisable hate speech detection: a review on obstacles and solutions. Peer J Comput Sci. 2021; 7: 598. https://doi.org/10.7717/peerj-cs.598.

10. Kim B, Wang Y, Lee J, Kim Y. Unfriending effects: Testing contrasting indirect-effects relationships between exposures to hate speech on political talk via social media unfriending. Comput Human Behav. 2022; 137: 107414. https://doi.org/10.1016/j.chb.2022.107414.

11. Kumar G, Singh JP, Singh AK. Autoencoder-based feature extraction for identifying hate speech spreaders in social media. IEEE Trans Comput Soc Syst. 2023; 10(2): 315-328. https://doi.org/10.14569/IJACSA.2023.0140542

12. Alatawi HS, Alhothali AM, Moria KM. Detecting white supremacist hate speech using domain specific word embedding with deep learning and BERT. IEEE Access. 2021; 9: 106363-106374. Available from: https://arxiv.org/abs/2010.00357.

13. Qureshi KA, Sabih M. Un-compromised credibility: Social media based multi class hate speech classification for text. IEEE Access. 2021; 9: 109465-109477. https://doi.org/10.1109/ACCESS.2021.3101977.

14. M-Harigy LM, Al-Nuaim HA, Moradpoor N, Tan Z. Building towards automated cyberbullying detection: A comparative analysis. Comput Intell Neurosci. 2020. https://doi.org/10.1155/2022/4794227.

15. Chhabra A, Vishwakarma DK. A literature survey on multimodal and multilingual automatic hate speech identification. Multimedia Syst. 2023; 29: 1203-1230. https://doi.org/10.1007/s00530-023-01051-8.

16. Wu XK, Zhao TF, Lu L, Chen WN. Predicting the hate: A GSTM model based on COVID-19 hate speech datasets. Inf Process Manag. 2022; 59: 102998.

17. Imran A, Yongming L, Witold P. Granular computing approach for the ordinal semantic weighted multiscale values for the attributes in formal concept analysis algorithm. J Intell Fuzzy Syst. 2023; 45: 1567–1586. https://doi.org/10.3233/JIFS-223764.

18. Ali I, Li Y, Pedrycz W. Granular computing approach to evaluate spatio-temporal events in intuitionistic fuzzy sets data through formal concept analysis. Axioms. 2023; 12(5): 407-423. https://doi.org/10.3390/axioms12050407

19. Sharmila P, Anbananthen KSM, Chelliah D, Parthasarathy S, Kannan S. PDHS: Pattern-based deep hate speech detection with improved tweet representation. IEEE Access. 2022; 10: 105366-105376. https://doi.org/10.1109/ACCESS.2022.3210177.

20. Ganfure GO. Comparative analysis of deep learning based Afaan Oromo hate speech detection. J Big Data. 2022; 9(76). https://doi.org/10.1186/s40537-022-00628-w.

21. Mullah NS, Zainon WMNW. Advances in machine learning algorithms for hate speech detection in social media: A review. IEEE Access. 2021; 9: 88364-88376. https://doi.org/10.1109/ACCESS.2021.3089515.

22. Khan S, Kamal A, Fazil M, Alshara MA, Sejwal VK, Alotaibi RM, Baig AR, Alqahtani S. Hcovbi-caps: Hate speech detection using convolutional and bi-directional gated recurrent unit with capsule network. IEEE Access. 2022; 10: 7881–7894. https://doi.org/10.1109/ACCESS.2022.3143799.

23. Poletto F, Basile V, Sanguinetti M, Bosco C, Patti V. Resources and benchmark corpora for hate speech detection: A systematic review. Lang Resour Eval. 2021; 55: 477–523. https://doi.org/10.1007/s10579-020-09502-8.

24. Siddiqui JA, Yuhaniz SS, Memon ZA, Amin Y. Improving hate speech detection using machine and deep learning techniques: A preliminary study. Open Int J Infor. 2021; 9(1): 45-59.

25. Wang CC, Day MY, Wu CL. Political hate speech detection and lexicon building: A study in Taiwan. IEEE Access. 2022; 10: 44337-44346. https://doi.org/10.1016/j.chb.2022.107414.

26. Abro S, Alzahrani AJ, Mehmood A, Khalid H, Rashid F, Cheikhrouhou O, Salehi S, et al. Automatic hate speech detection using machine learning: A comparative study. Int J Adv Comput Sci Appl. 2020; 11(1): 123-131. https://doi.org/10.14569/IJACSA.2020.0110861

27. Khan MY, Qayoom A, Nizami MS, Siddiqui MS, Wasi S, Raazi SMKR. Automated prediction of good dictionary examples (GDEX): A comprehensive experiment with distant supervision, machine learning, and word embedding-based deep learning techniques. Complexity. 2021. https://doi.org/10.1155/2021/2553199.

28. Oriola O, Kotzé E. Evaluating machine learning techniques for detecting offensive and hate speech in South African tweets. IEEE Access. 2020; 8: 21496-21509. https://doi.org/10.1109/ACCESS.2020.3037073.

29. Bilal M, Khan A, Jan S, Musa S. Context-aware deep learning model for detection of Roman Urdu hate speech on social media platform. IEEE Access. 2022; 10: 121133-121151. https://doi.org/10.1109/ACCESS.2022.3216375.

30. Robinson D, Zhang Z, Tepper J. Hate speech detection on Twitter: Feature engineering vs. feature selection. In: The Semantic Web: ESWC 2018 Satellite Events. Cham: Springer; 2018. p. 46-49. https://doi.org/10.1007/978-3-319-98192-5_9.

31. William P, Gade R, Chaudhari RE, Pawar AB, Jawale MA. Machine learning based automatic hate speech recognition system. In: 2022 International Conference

on Sustainable Computing and Data Communication Systems (ICSCDS). IEEE; 2022. p. 315-318. https://doi.org/10.1109/ICSCDS53736.2022.9760959

32. Agrawal T, Chakravarthy VD. Cyberbullying detection and hate speech identification using machine learning techniques. In: 2022 Second International Conference on Interdisciplinary Cyber Physical Systems (ICPS). IEEE; 2022. p. 182-187. https://doi.org/10.1109/ICPS55917.2022.00041.
33. Roy PK, Tripathy AK, Das TK, Gao XZ. A framework for hate speech detection using deep convolutional neural network. IEEE Access. 2020; 8: 204951–204962. https://doi.org/10.1109/ACCESS.2020.3037073.
34. Ayo FE, Folorunso O, Ibharalu FT, Osinuga IA, Alli AA. A probabilistic clustering model for hate speech classification in Twitter. Expert Syst Appl. 2021; 173: 114762. https://doi.org/10.1016/j.eswa.2021.114762.
35. Liu L, Xu D, Zhao P, Zeng DD, Hu PJH, Zhang Q, Luo Y, Cao Z. A cross-lingual transfer learning method for online COVID-19-related hate speech detection. Expert Syst Appl. 2023; 234: 121031. https://doi.org/10.1016/j.eswa.2023.121031.
36. Pérez JM, Luque FM, Zayat D, Kondratzky M, Moro A, Serrati PS, Zajac J, Miguel P, Debandi N, Gravano A, Cotik V. Assessing the impact of contextual information in hate speech detection. IEEE Access. 2023; 11: 30575-30590. https://doi.org/10.1109/ACCESS.2023.3258973.
37. Makhadmeh ZA, Tolba A. Automatic hate speech detection using killer natural language processing optimizing ensemble deep learning approach. Computing. 2020; 102: 501–522. https://doi.org/10.1007/s00607-019-00745-0.
38. Dwivedy V, Roy PK. Deep feature fusion for hate speech detection: A transfer learning approach. Multimed Tools Appl. 2023; 82: 36279–36301. https://doi.org/10.1007/s11042-023-14850-y.
39. Şahinuç F, Yilmaz EH, Toraman C, Koç A. The effect of gender bias on hate speech detection. Sig Img Proc Lett. 2023; 17: 1591–1597. https://doi.org/10.1007/s11760-022-02368-z
40. Miok K, Škrlj B, Zaharie D, Šikonja MR. To BAN or not to BAN: Bayesian attention networks for reliable hate speech detection. Cogn Comput. 2022; 14: 353–371. https://doi.org/10.1007/s12559-021-09826-9.
41. Chiril P, Pamungkas EW, Benamara F, Moriceau V, Patti V. Emotionally informed hate speech detection: A multitarget perspective. Cogn Comput. 2022; 14: 322–352. https://doi.org/10.1007/s12559-021-09862-5.
42. Stanković SV, Mladenović M. An approach to automatic classification of hate speech in sports domain on social media. J Big Data. 2023; 10(109): 1-15. https://doi.org/10.1186/s40537-023-00766-9.
43. Díaz JAG, Zafra SMJ, Cumbreras MAG, García RV. Evaluating feature combination strategies for hate-speech detection in Spanish using linguistic features and transformers. Complex Intell Syst. 2023; 9: 2893–2914. https://doi.org/10.1007/s40747-022-00693-x.
44. Ghosh S, Ekbal A, Bhattacharyya P, Saha T, Kumar A, Srivastava S. SEHC: A benchmark setup to identify online hate speech in English. IEEE Trans Comput Soc Syst. 2023; 10: 760-770. https://doi.org/10.1038/s41598-022-08438-z.
45. Min B, Xu H, Ma J, He X, Wang M, Guo H, Wang W, Zheng K, Jin D, Zhang C. Recent advances in natural language processing via large pre-trained language models: A survey. ACM Comput Surv. 2023; 56(2): 1–40. https://doi.org/10.1145/3605943.
46. Joni S, Maximilian H, Shammur A, Soon-Gyo J. Developing an online hate classifier for multiple social media platforms. Hum Centric Comput Inf Sci. 2020; 10(1): 1-14. https://doi.org/10.1186/s13673-019-0205-6.
47. Pronoza E, Panicheva P, Koltsova O, Rosso P. Detecting ethnicity-targeted hate speech in Russian social media texts. Inf Process Manag. 2021; 58: 102674. https://doi.org/10.1016/J.IPM.2021.102674.
48. Subba B, Gupta P. A tfidf vectorizer and singular value decomposition based host intrusion detection system framework for detecting anomalous system processes. Comput Secur. 2021; 110: 102084. https://doi.org/10.1016/j.cose.2020.102084.
49. Smith MQR, Ruxton GD. Effective use of the McNemar test. Behav Ecol Sociobiol. 2020; 133. https://doi.org/10.1007/s00265-020-02916-y.

Published Online First: November, 2024
https://dx.doi.org/10.21123/bsj.2024.10743
P-ISSN: 2078-8665 - E-ISSN: 2411-7986

Baghdad Science Journal

كشف العداء عبر الإنترنت: تحليل وتخفيف خطاب الكراهية في وسائل التواصل الاجتماعي

جويد أحمد صديقي[1]، ستي صوفياتي يوهانيز[1]، ذو الفقار علي ميمون[2]

[1]كلية رزاك للتكنولوجيا والمعلوماتية، جامعة التكنولوجيا ماليزيا، كوالالمبور، ماليزيا.
[2]المدرسة السريعة للحوسبة، الجامعة الوطنية للحاسوب والعلوم الناشئة، كراتشي، باكستان.

الخلاصة

تعمل منصات التواصل الاجتماعي على توليد كمية هائلة من البيانات في كل ثانية. تويتر، من الناحية العملية، ينتج الأفراد أكثر من ستمائة تغريدة في كل ثانية. أثناء نشر آراء المستخدمين وتعبيراتهم بحرية، من الصعب جدًا حصر خطاب الكراهية الذي يتم مشاركته ضد أي فرد أو دين أو أي مجموعة عرقية. وبالتالي، فإن الأشخاص المستهدفين بمثل هذا المحتوى الذي يحض على الكراهية يشعرون بالإحباط. وفي هذا الصدد، قامت الأساليب المختلفة بحل هذه المشكلة الخطيرة، ولكنها في بعض الأحيان لم تتمكن من تحقيق نتائج مرضية. ولذلك، نقترح نماذج مختلفة للتعلم الآلي لتصنيف البيانات المعطاة إلى فئتين، مسيئة أو غير مسيئة. تم إجراء التجارب على بيانات تويتر التي أنشأناها بأنفسنا باستخدام Twitter API ومكتبة Tweepy بواسطة Python.تم تقييم النتائج الناتجة بناءً على مقاييس مختلفة مثل الدقة والاستدعاء وقياس F1 واختبار MCNEMAR. بالمقارنة مع خوارزميات التعلم الآلي المختلفة، تفوق مصنف مجموعة الغابات العشوائية على الخوارزميات الأخرى، فإن حداثة ومساهمة ورقتنا البحثية هي: تطوير مجموعة بيانات تويتر التي تتكون من عدة تغريدات تحتوي على 11 متغير كائن مع أربعة متغيرات فئة مختلفة تظهر الهجوم المختلف المستويات، وتطبيق خوارزميات التعلم الآلي للكشف عن خطاب الكراهية، والتحليل المقارن لخوارزميات التعلم الآلي المختلفة مقابل مقاييس تقييم مختلفة بما في ذلك اختبار ماكنيمار. يتم شرح أهمية التقنية المقترحة جيدًا من خلال مجموعات بيانات Twitter التي تم إنشاؤها من خلال Twitter API ومكتبة Tweepy بواسطة Python.

الكلمات المفتاحية: كشف خطاب الكراهية، التعلم الالي؛ معالجة اللغة الطبيعية، وسائل التواصل الاجتماعي، تصنيف النص.