International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

A Comparative Analysis of Machine Learning Models Used for Hate Speech (Hs) Detection of Odia Language

Aloka Natha[1], Dr. Bichitrananda Behera[2], Debaswapna Mishra[3], Saumya Ranjan Sahu[4], Subhasis Mohapatra[5]

Submitted: 12/12/2023 **Revised**: 16/01/2024 **Accepted**: 01/02/2024

Abstract: Social media has changed the way of communication and interaction around the world. However, hostile content can be readily exposed by the internet community due to the growth and obscurity of online media. The persons who are spreading unpleasant or hateful posts to a group of people based on their gender, political stance, color, ethnicity, or any other characteristic must be identified, either individually or as a collective. Odia, one of the six classical Indo-Aryan languages of India, is spoken by 82% of Odisha's native speakers and the remaining 18% of people who live in the states of West Bengal, Jharkhand, Andhra Pradesh, and Chhattisgarh. Due to a lack of resources, relatively little research has been done in the literature on the detection of hate speech (HS) in the Odia language. This paper's primary goal is to compile posts and comments from social media pages into an HS dataset for the Odia language. There are two categories for this dataset: HS and non-HS. Feature extraction methods and machine learning classification algorithms can be used with this dataset to identify the HS patterns from a certain social media post. Here, a comparative analysis is carried out by using this generated dataset of HS to train several machine learning models. The models' performance is compared using several metrics, including F1-score, accuracy, precision, and recall.

Keywords: *component, formatting, style, styling, recall, precision, f1 score, accuracy*

1. Introduction

Social media such as Facebook, Twitter, and Instagram have transformed the communication and cultural landscapes of countries all over the world [16]. It typically concentrates on topics like gender, opinion, emotion, race, ethnicity, and religion [8]. However, the portability and anonymity of online social media define language that targets individuals or groups of individuals based on attributes. The description demonstrates how statements or hate speech (HS) that incite hatred or violence in some individuals.

The majority of social media studies describe HS as words that attack or criticize someone individually or as a whole based on characteristics including race, physical appearance, gender, ethnicity, religion, political ideas, etc. The argument focuses on how HS discourse incites animosity or violence toward specific communities. Hate speech is defined as offensive speech that can spread swiftly on social media due to prejudices or conflicts between different groups, both locally and internationally [23, 25]. Crimes motivated by hate include any actions taken against an individual due to their real or assumed affiliation with a certain group. Toxic discourses found online have the power to destroy communities and incite conflict between various groups. Hate speech [28, 29, 30] is multilayered, complex, incendiary information that targets particular individuals or groups. Numerous facets of hate speech are depicted in the below figure.

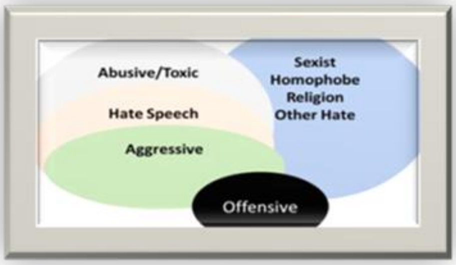


Fig. 1 Categories of Hate Speech [39]

Big data is a result of social media's continually growing content. Manually identifying hate speech from millions of text passages can be difficult for humans. Using NLP or machine learning for automated hate speech detection and moderation might be a smart approach to address this problem. The techniques for automatically identifying hate speech are predicated on numerical text representations, which are then fed into classification models. The state-of-the-art in this subject uses lexical characteristics as input features: noun patterns, polar intensity, word and character n-grams, Term Frequency-Inverse Document Frequency

[1] *C V Raman Global University, Bhubaneswar – 752054, Odisha*
Email: aloknath23@gmail.com
[2] *C V Raman Global University, Bhubaneswar – 752054, Odisha*
Email: dr.bichitrananda.behera@gmail.com
[3] *C V Raman Global University, Bhubaneswar – 752054, Odisha*
Email: debaswapnamishra@gmail.com
[4] *C V Raman Global University, Bhubaneswar – 752054, Odisha*
Email: sahu.saumya@gmail.com
[5] *Sri Sri University, Cuttack- 754006, Odisha*
Email: subhasis22@gmail.com
** Corresponding Author Email: aloknath23@gmail.com*

(TF-IDF) [17], Bag of Words (BoW), and so on...

In India, particularly in the state of Odisha, hate speech (HS) on social media[11] based on specific political beliefs, ethnicity, or religious affiliation is currently pervasive. Most of this HS encourages violence and attacks against specific people or groups. It is crucial to monitor or automatically detect hate speech on social media platforms and stop it from spreading. However, the majority of Odia officially changed its name from Oriya to Odia in November 2011) [21] and wrote, communicated, and exchanged messages in Odia. As a result, automatic hate speech identification for the Odia language is required.

Many parts of eastern India are native speakers of the Indo-Aryan language known as Odia. The majority of speakers in the state of Odisha are over 75% of its population. Odia is the mother tongue of more than 55 million people worldwide. Online access to Odia data is now easier than it has ever been. This includes news, poetry, short stories, blog posts, play and novel scripts, and more. Unicode standards for multiple Indian languages have made more Odia data available online.

Odia is still considered to have limited resources because it is devoid of several essential tools needed for many tasks related to natural language processing. These include sentiment lexicons, large parallel corpora, annotated corpora for various domains, a Morphanalyser, a machine translation system, and an appropriate Parts-Of-Speech tagger. Language research over the past few years has made an effort to close this gap. It is vital to mine this data to provide resources and tools for the language because Odia digital content creators are multiplying.

To the best of our knowledge, in the literature, very few works have been carried out for Odia hate speech detection using machine learning algorithms. We have used significant machine learning algorithms for Odia hate speech detection in this research paper, and we have analyzed their effectiveness. Hence the contribution of our research paper is presented as follows.

- Different chat or tweet messages from social media platforms like Facebook and Twitter, as well as from different news websites, have been gathered to create an Odia dataset to detect hate speech.
- Important machine learning algorithms such as decision tree (DT), and support vector machine (SVM), are presented and their performance is compared using various state-of-the-art performance metrics.

2. Literature Review

This section provides a summary of previous research methods and conclusions regarding the automatic identification of hate speech on social media.

Numerous prominent, low-resource languages, including Odia, are the subject of my current research. Nonetheless, it is undoubtedly preferable for any natural language processing model to undergo validation about its capacity to extrapolate results from data and languages that differ from those used for training and testing. The idea of language independence, which characterizes models that can be engineered to function reasonably well across languages, is analogous to this one.

Odia text classification covers the areas of a) *text classification* b) *sentiment analysis* c) *hate speech detection*.

Sagarika et al. [15] provide the scenario for the text summarizing and text mining. They proposed an effective and extractive single document summarizer for the classification of the Odia text document. They used both statistical methods and clustering methods for evaluating the F scores. They gathered documents that were related to cricket news from popular daily newspapers i.e., Samaja, Dharitri, and Sambada for datasets. They have implemented the steps of processing, sentence analysis, word analysis, and sentence extraction. They have also used two methods, first is TF-IDF of words and secondly agglomerative hierarchical clustering. They found out the F score of 66.858%.

Brojo Kishore et al. [16] focus on mining opinions or people's sentiments which comprises the following phases such as Lexical Morphological and Syntactic Semantic stages. Furthermore, they have focused on Root words, part of speech, suffixes, and synonyms of words in Odia text. They also proposed several techniques for opinion mining for the detection of subjectivity as well as polarity among the texts. The sentiment analysis was performed by using the Naïve Bayesian Classifier and Opinion Mining was done by using the Support Vector Machine. They have analyzed Odia language text to determine whether a document or sentence carries a positive or negative opinion. In the Syntactic-Semantic phase, they used a framework which was known as the Paninian framework, which was used to classify the text whether it was positive or negative.

Manoj Kumar et al. [17] defined a new model named 'Laxmi', which was used for both training and testing the datasets. Also, they have tried to implement the N-Gram-based Support Vector Machine for categorizing the opinions from the dataset and labeled the opinions as positive, negative, and neutral. The model 'Laxmi' pre-processed names, numbers, and filters the stop word, and used stemming procedure to the texts. The model 'Laxmi' used the N-Gram model and machine learning algorithms for dataset learning and testing over the dataset. Several parameters were used such as True Positives, True Negatives, False Positives, and False Negatives to categorize the opinions. Also, they have used a dataset that has 8000 data for calculating accuracy and F1 Score, etc.

For Odia language understanding, Dey et al. [18] developed a Neoteric Contextual model called SOLMAT (Satyajit Odia Language Model for Analysis of Text). It was extremely challenging and time-consuming for him to find out that, out of 100 works, over 67% are based on machine learning and deep learning, while 33% are based on context. Results from contextual analysis were more accurate than those from other methods. Thus, they created a model that was able to interpret Odia utterances in their context with clarity. As a result of this point of view, it forecasted the opinion and defined the opinion by identifying various things.

Gourav et al. [19] discussed a method that was used to create a SentiWordNet for Odia statements. He created the lexicon, which would serve as a tool for Sentiment Analysis of Odia data. Odia WordNet was used to create the lexicon which comprises nouns, verbs, adjectives, etc. The target lexicon comprises word, polarity, POS and to create the Target Lexicon, he used Odia WordNet. The Odia WordNet used synset ID for categorizing the Odia texts into positive and negative.

Pruthwik et al. [20] focused on the sentiment analysis in Odia literature in which, they have created an annotated corpus of various Odia sentences. For the training and testing on the sentiment annotated corpus, they have used Support Vector Machines (SVM), Logistic Regression (LR), and Random Forest (RF) machine learning classifiers. Also, the following features were used to train the classifiers: TF-IDF Word-Level Features, TF-IDF Character-Level Features, and Affective words from the Odia Sentiment Lexicon. They have created an annotated corpus of 2045 Odia sentences from different news articles like the Samaaja News Archive. Also, they have categorized these sentences into three categories i.e., positive, negative, and neutral. They have conducted both binary and ternary sentiment classification on the obtained dataset.

Mohapatra et al. [2] proposed a model for hate speech detection that encapsulates the machine learning and feature extraction of texts. He has collected the hate-speech data of mixed English-Odia from a popular online social platform i.e., Facebook. They also have manually categorized the posts and comments into three classes i.e., hate (HS), offensive (OFS), and neither (OK) speech. For detecting hate speech, they have implemented different methods which include varieties of comments and posts, developing annotation, text pre-processing, feature extraction, etc. For training both datasets they have used SVM, Naïve Bayes, and Random Forest Models, and for calculating the feature extraction they have used TF-IDF, n-grams (unigram, bigram, trigram, combined n-grams, etc.), and word2vec. They have categorized 2 types of datasets i.e. binary class dataset and ternary class dataset. Different classifiers were used to define the performance of the models among which SVM and word2vec achieved better i.e., 73% than the NB and RF classifiers. Also, the ternary models performed better than the binary models.

Parida et al. [21] used a vector space model (VSM), which is used to classify the Odia text document. This model is carried out by the concepts of linear algebra. VSM focuses on queries and documents which are termed the vectors. In this, the vectors are represented in the multidimensional space and the terms are used as the dimensions to design the index key. They have considered the variety of techniques for the information retrieval and then it classifies the term count model, classical vector space model, and normalized vector space model. These three models work on the TF-IDF (Term Frequency- Inverse Document Frequency). The normalized frequency model provides better results than the other model for large documents while the Term Count Model is suited for small documents. The classical vector space model removes the non-meaningful words to increase the score values.

Pattnaik et al. [22] focused on hierarchical clustering which utilizes the cosine similarity measure for segregating the Odia sentences. The model took an extraction technique method for summarizing the Odia sentences. It also provided the best minimization technique for the redundancy and the result was achieved by implementing the cosine similarity matrix. The performance of the model was evaluated by calculating the following parameters: precision (p), recall (r), F score, etc. The average F score value was 60.272%.

Mohanty et al. [23] made an impact on different newspaper articles which comprised different public opinions. News articles can be processed to extract the different sentiments for opinion mining. They have taken Odia news articles from the portal as well as e-paper websites. A total 500 number of news articles were collected. Among them, 350 news articles were used for the training while 150 news articles were used for testing. The news articles were pre-processed and vectorized by TF, and TF-IDF. The SVM (Support Vector Machine) classifier was used for calculating the accuracy F1 score and confidence measure.

Bishwa et al. [24] focused on text classification which used the K-NN classifier to construct the closest matches in training data for forecasting the class of the text. To find out the closest matches, they used Euclidean distance for the prediction of the class levels. They also created a corpus in Odia which belongs to an agricultural domain collected from ILCI, Govt. of India, and was called the training data. To find out the sentiment in the text data, the feature vector method was used. The evaluation was done by calculating the recall, precision, and $F_{\text{-score}}$ parameters.

Sahoo et al. [29] represented an auxiliary feature extraction method for Odia text which determined the features by using Naïve Bayes classifier. This classifier used conditional probability to improve the classification accuracy. They have proposed a method that increases the Naïve Bayes classifier's performance level. For better empirical diagnosis, they have taken nearly about 1000 sentences both for training and testing purposes. Also, they used the Bigram and Trigram concepts to evaluate the probability values of each sentence. For the smooth functioning of the model, three kinds of classifications are used i. e., positive, negative, and neutral. They have focused on a hybrid model to obtain domain flexibility.

Bishwa et al. [35] defined the translation relationship between the two languages Odia and Bangla. Multi-word units are managed by the use of idiomatic expressions, multiple words, and non-compositional complex words. They used a bilingual lexicon, which worked on converting a source text into multiple words. The EM (Expectation and Maximization) algorithm is used to find out the maximum likelihood function between the possible outcome values. A very small size of dataset was used for both training and testing the model.

Table 1: Summary of Text Classification in Odia language-related work

Authors	***Objective***	***Feature Extraction***	***ML Method & Accuracy***
Sagarika et al. [15]	Text summarization for the Odia document	TF-IDF, F score	F-score of 66.858%
Brojo Kishore et al. [16]	Knowledge mining approach for Odia language	Opinion mining, Sentiment analysis	Naïve Bayes, Support Vector Machine
Manoj Kumar et al. [17]	Contextual Opinion Mining in Odia text	Unigrams, Bigrams, Opinion Mining, Sentiment Analysis	Accuracy 80% and F1 score 0.701
Dey et al. [18]	SOLMAT: A Neoteric Contextual Model for Odia Language	Contextual Analysis, Statistical Hypotheses	Odia Language Model for Analysis of Test
Gourav et al. [19]	Building a sentiwordnet for Odia	Sentiment Analysis, Odia SentiWordNet	Fleiss Kappa (k=0.76)
Pruthwik et al. [20]	Annotated corpus for sentiment analysis in Odia language	Odia Sentiment Lexicon	Agreement Score = 0.79
Mohapatra et al. [2]	Hate Speech Detection in English-Odia Code	Tf-Idf, NB, RF	SVM Model F1 Score= 73%
Parida et al. [21]	Odia Text Document using Vector Space Model.	Vector Space Model, Tf-Idf, Text Summarization, Data Mining	Tf-Idf score= 0.699, Weight= 0.096
Pattnaik et al. [22]	Summarize the Odia text document using cosine similarity and clustering	Hierarchical clustering, cosine similarity matrix, Tf-Idf, Precision (p), Recall (r)	Cosine Values = 0 to 1, F score= 60.272%
Mohanty et al. [23]	Odia Newspaper Articles on Public Opinion	Tf, Tf-Idf, Accuracy, F1 score	Accuracy = 0.84, F1 score = 0.79
Bishwa et al. [24]	Odia Text Classification for Sentiment Analysis using K-NN	Tf-Idf, K-NN	Precision, recall, F_{score}
Sahoo et al. [29]	Odia Text Classification Using Naïve Bayes Algorithm	HMM (Hidden Markov Model), CRF (Conditional Random Field), K-NN	
Bishwa et al. [35]	Word Alignment in Bilingual Text for Bangla to Odia Machine Translation	Parallel Corpus, EM (Expectation and Maximization)	

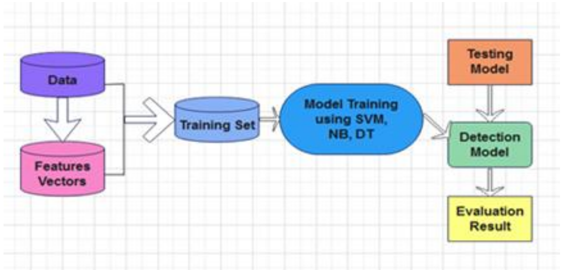


Fig. 2 Machine learning building model

A machine learning model's functionality is shown in the above picture, which includes a training dataset and a testing dataset. The training dataset is applied to the model by applying different machine learning algorithms and observations are observed. Lastly, the model is tested against a testing dataset for the model's prediction as a result of an output.

3. Dataset

Automatic detection of discrimination from Odia text is the topic of this study. Because there isn't a dataset or annotated dataset for this investigation, a fresh Odia HS dataset must be created. The dataset was created in three steps: (1) collecting the texts of the Odia Post and the view from public Facebook pages; (2) processing and filtering the data stored in the dataset; and (3) interpreting the data.

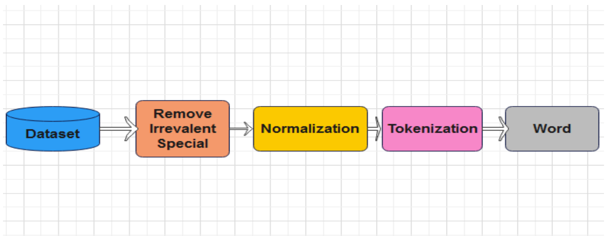
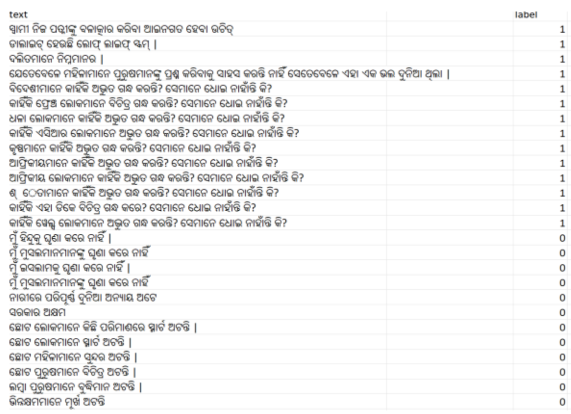


Fig. 3 Dataset Pre-processing

A dataset should be used to train both regular and atypical queries to detect system anomalies. Both their own and publicly available datasets are acceptable for usage by researchers.



text	label
ସ୍ୱାମୀ ନିଜ ପତ୍ନୀଙ୍କୁ ବଳାତ୍କାର କରିବା ଆଇନଗତ ହେବା ଉଚିତ୍	1
ଡାଲାଇଟ୍ ହେଉଛି ଲୋଫ୍ ଲାଇଫ୍ ସ୍କମ୍ ।	1
ଦଳିତମାନେ ନିମ୍ନମାନର ।	1
ଯେତେବେଳେ ମହିଳାମାନେ ପୁରୁଷମାନଙ୍କୁ ପ୍ରଶ୍ନ କରିବାକୁ ସାହସ କରନ୍ତି ନାହିଁ ସେତେବେଳେ ଏହା ଏକ ଭଲ ଦୁନିଆ ଥିଲା ।	1
ବିଦେଶୀମାନେ କାହିଁକି ଅଦ୍ଭୁତ ଗନ୍ଧ କରନ୍ତି? ସେମାନେ ଧୋଇ ନାହାଁନ୍ତି କି?	1
କାହିଁକି ଫ୍ରେଞ୍ଚ ଲୋକମାନେ ବିଚିତ୍ର ଗନ୍ଧ କରନ୍ତି? ସେମାନେ ଧୋଇ ନାହାଁନ୍ତି କି?	1
ଧଳା ଲୋକମାନେ କାହିଁକି ଅଦ୍ଭୁତ ଗନ୍ଧ କରନ୍ତି? ସେମାନେ ଧୋଇ ନାହାଁନ୍ତି କି?	1
କାହିଁକି ଏସିଆର ଲୋକମାନେ ଅଦ୍ଭୁତ ଗନ୍ଧ କରନ୍ତି? ସେମାନେ ଧୋଇ ନାହାଁନ୍ତି କି?	1
କୃଷ୍ଣମାନେ କାହିଁକି ଅଦ୍ଭୁତ ଗନ୍ଧ କରନ୍ତି? ସେମାନେ ଧୋଇ ନାହାଁନ୍ତି କି?	1
ଆଫ୍ରିକୀୟମାନେ କାହିଁକି ଅଦ୍ଭୁତ ଗନ୍ଧ କରନ୍ତି? ସେମାନେ ଧୋଇ ନାହାଁନ୍ତି କି?	1
ଆଫ୍ରିକୀୟ ଲୋକମାନେ କାହିଁକି ଅଦ୍ଭୁତ ଗନ୍ଧ କରନ୍ତି? ସେମାନେ ଧୋଇ ନାହାଁନ୍ତି କି?	1
ଶ୍ େତାମାନେ କାହିଁକି ଅଦ୍ଭୁତ ଗନ୍ଧ କରନ୍ତି? ସେମାନେ ଧୋଇ ନାହାଁନ୍ତି କି?	1
କାହିଁକି ଏହା ଡିକେ ବିଚିତ୍ର ଗନ୍ଧ କରେ? ସେମାନେ ଧୋଇ ନାହାଁନ୍ତି କି?	1
କାହିଁକି ୱେଲ୍ସ ଲୋକମାନେ ଅଦ୍ଭୁତ ଗନ୍ଧ କରନ୍ତି? ସେମାନେ ଧୋଇ ନାହାଁନ୍ତି କି?	1
ମୁଁ ହିନ୍ଦୁକୁ ଘୃଣା କରେ ନାହିଁ ।	0
ମୁଁ ମୁସଲମାନମାନଙ୍କୁ ଘୃଣା କରେ ନାହିଁ	0
ମୁଁ ଇସଲାମକୁ ଘୃଣା କରେ ନାହିଁ ।	0
ମୁଁ ମୁସଲମାନମାନଙ୍କୁ ଘୃଣା କରେ ନାହିଁ	0
ନାରୀରେ ପରିପୂର୍ଣ୍ଣ ଦୁନିଆ ଅନ୍ୟାୟ ଅଟେ	0
ସରକାର ଅକ୍ଷମ	0
ଛୋଟ ଲୋକମାନେ କିଛି ପରିମାଣରେ ସ୍ମାର୍ଟ ଅଟନ୍ତି ।	0
ଛୋଟ ଲୋକମାନେ ସ୍ମାର୍ଟ ଅଟନ୍ତି ।	0
ଛୋଟ ମହିଳାମାନେ ସୁନ୍ଦର ଅଟନ୍ତି ।	0
ଛୋଟ ପୁରୁଷମାନେ ବିଚିତ୍ର ଅଟନ୍ତି ।	0
ଲମ୍ବା ପୁରୁଷମାନେ ବୁଦ୍ଧିମାନ ଅଟନ୍ତି ।	0
ଭିନ୍ନକ୍ଷମମାନେ ମୂର୍ଖ ଅଟନ୍ତି	0

Fig. 4 A sample Odia Hate Speech Dataset

Data Collection

Odia data is collected from posts and comments on various popular Facebook pages and other social media platforms like My Space and Twitter, as well as news channels, which make up the Odia text data because Facebook's privacy policy prohibits access to a private page's public material. The public pages in the directory that have received the most likes and dislikes relative to other pages in that category are included in this dataset. All rumors and opinions are sourced from official pages, which cover the political agenda all year long and cover numerous experiences the nation has had. The country saw a growth in the usage of social media during this time, particularly Facebook. Selected public Facebook pages and news channel comments provide information from each category. Additionally, this study gathers keywords for annotating articles and comments as well as filtering the acquired Odia text data.

Data Preparation

Data preparation follows data collection, including collecting, cleaning, filtering, and compiling Data is stored in a file or database. Cleaning and filtering raw data are important for future work-especially training models for Odia language discrimination and the explanation of the words and phrases in the datasheet. Below is the list of activities performed during the Document Preparation process. Remove all postings and comments that are not in Odia, English, or text. Remove all whitespace, null, values, and empty lines. Use keywords to filter out information that may include rude or hateful language. Add the data from each page to one dataset. To verify that each text in the dataset is unique, eliminate duplication.

4. Hate Speech Detection Architecture

Suggestions for identifying offensive words and expressions in Odia English, profanity, or normal speech are shown in Figure 5. Then preprocesses them according to the nature of the message. It consists of eliminating punctuation, normalizing, tokenizing, and other fundamental pre-processes that are required.

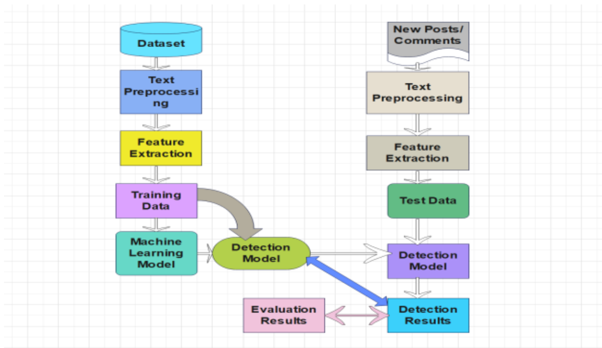


Fig. 5 Hate Speech Architecture

Then, to extract features, utilize n-gram, word2vec, and TF-IDF. The data needed to train the model is extracted to create the base feature vectors (training data). By following feature extractions, SVM [14,7] NB, and RF machine learning techniques are used to train these models. These actions result in the development of attack and hate speech investigational models. Based on the outcomes of the debate on test models, evaluate and choose test models.

Feature Extraction

The act of extracting significant characteristics from a dataset is known as feature extraction. This model has used the retrieved characteristics. The relevant features of the dataset are extracted using the suggested feature extraction methods. It accepts prepossessed and tokenized dataset terms as input and executes extractions.

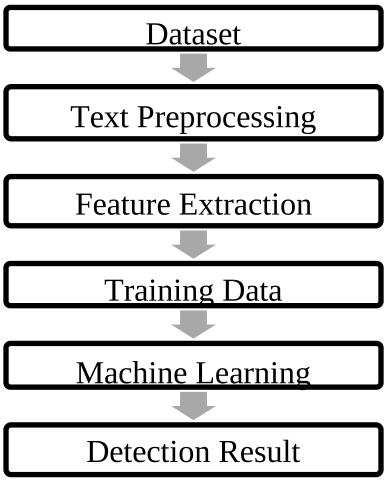


Fig. 6 Data Flow of Feature Extraction

The diagrammatic model of feature extraction is shown in the figure above. It operates on several datasets that are utilized for feature extraction. From the dataset, the data are processed to the next stage which is known as text preprocessing. This involves processing the entire dataset before extracting the pertinent features. Now, using the appropriate machine learning methods, we will train the system in this manner.

TF-IDF Feature Extraction

The frequency is the number of times a word 't' appears. So, it seems plausible that when a word appears in the text, it will be more relevant because it is visible. We can use vectors to describe the text in the form of words because the order of the elements is not important. Dataset Text Preprocessing Feature Extraction Training Data Machine Learning Modeling Detection Result. This is an object with the value "Term Frequency" for each specific term used in the document. For every individual phrase, there is a record in the document with the value "term frequency."

Word2vec Feature Extraction

The model computes the middle of all vectors using standard text properties. in the documents. In this study, feature extraction was performed with a combination of some of these methods, such as TF-IDF and weighted n-gram., rather than just a single method. In the word bag model, since the elements' order is irrelevant, we can describe text using vectors. For each custom word used in the document, there is an item with the value "Control Frequency".

N-Gram Feature Extraction

Feature extraction will be done using composite n-grams. N-gram performance requires the correct selection of the n-value and will also provide different models to show and compare n-gram properties. The choice of N depends on the task and its characteristics of detail. For example, bigrams (N =2) can capture more information than single bundles (N=1) but also introduce space-specific and computational complexity. Triplets (N =3) and higher N-grams can capture more content but also suffer from variability where some N-grams may appear once or very rarely in the information.

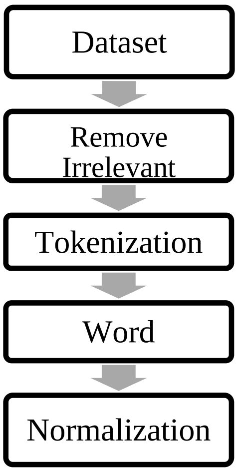


Fig.7 Data Flow of N-Gram Feature Extraction

ML ALGORITHM.

Decision Tree

A well-liked machine learning technique called decision trees can be used to tackle problems with both regression and classification. They are the best option for beginners in the field of machine learning since they are simple to comprehend, interpret, and use.

There are data and attributes for each document in the decision tree model.

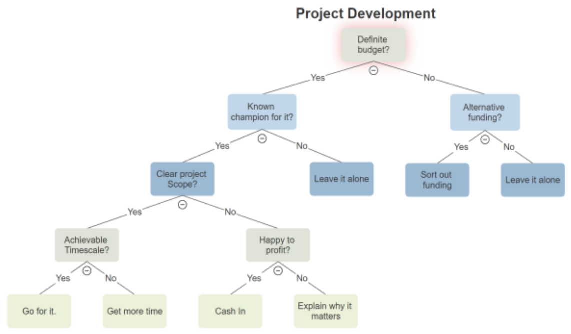


Fig.8 Pictorial representation of decision tree

The above figure represents the pictorial representation of a decision tree, which comprises several branches. At each point, a condition has arrived and based upon the conditions, a decision has to be carried out. Each condition is carried out by giving either 'Yes' or 'No' branches as left child and right child. The above figure is a decision tree of Project Development, in which, decisions are carried out based upon some conditions.

A decision tree's nodes (or features) stand in for features, links (or branches) for decisions (or rules), and leaves (or results) for outcomes. (Values by category or continuous). The objective is to create a tree similar to this one for all the data and finish the results for each leaf or lessen the mistakes on each leaf.

Naïve Bayes Classifier (NB)

The Naive Bayes classifier is a popular supervised machine learning method for applications like text classification [13]. It replicates the input distribution for one class or category and belongs to the family of generative learning algorithms. This tactic is based on the notion that the incoming data's attributes are conditionally independent of the class, allowing the algorithm to generate accurate and timely predictions.

The Naive Bayes classifier is a probability classifier that bases its decisions on the Bayesian posterior probability distribution. It upholds the restriction with the independent relationship between the qualities thanks to conditional probability. Based on the Bayes theorem and featuring, high feature independence criteria, the Naive Bayes algorithm is a machine learning supervised classification method. It is still the best method for classifying documents and texts and is typically used for binary or multiclass classification.

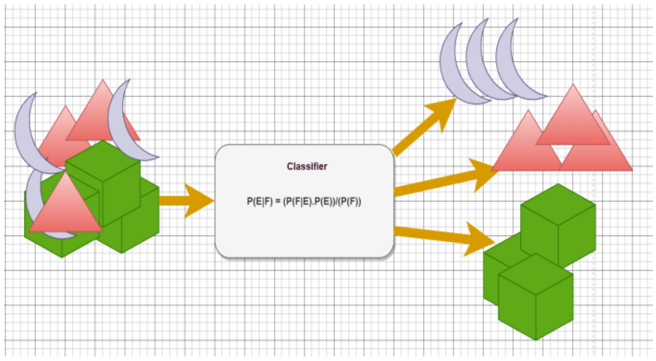


Fig. 9 Data Flow of Feature Extraction using Naïve Bayes

The above figure shows the experimental aspect of the Naïve Bayes classifier. We took 3 varieties of symbols i.e., 3 half-moons, 3 squares, and 3 rectangles. At first, we provide the all symbols. Then we use Naïve Bayes classifiers to classify. The classifier classifies into different classes of half-moon, square, and triangles.

In statistics, Naive Bayes classifiers are considered to be basic probabilistic classifiers that apply the Bayes theorem. This theorem determines the likelihood of a hypothesis based on the information at hand and some prior knowledge. Typically, the Naive Bayes classifier fails to recognize that features in input data are not independent of one another when used in real-world circumstances. The Naive Bayes classifier is widely used in real-world applications because of its efficacy and excellent performance, despite its oversimplifying assumption.

The following formula serves as a representation of the Bayes theorem [14]

$$P(E|F) = \frac{P(F|E).P(E)}{P(F)} \quad (1)$$

Where, F: Unclassified groupings and the facts

E: The F facts statement is a member of a certain category.

P(E|F): The likelihood of a hypothesis E depends on the state being present in state F

P(E): Probability of assumption E

P(F|E): F likelihood varying with states

P(F): F probability

K-Nearest Neighbor Classifier (KNN)

The majority of classifiers in the literature invest more time in the training phase of creating the classification model and are regarded as keen learners [10]. However, the testing phase of the KNN classifier takes more time to predict class names for the new unnamed file data structure. That's why it's called lazy work. The K-Neighbors classifier stores all the training data and their target class during the training of the design. The nearest neighbor algorithm calculates the distance between the test data and all the data when it receives a new test document for classification with an unknown target place.

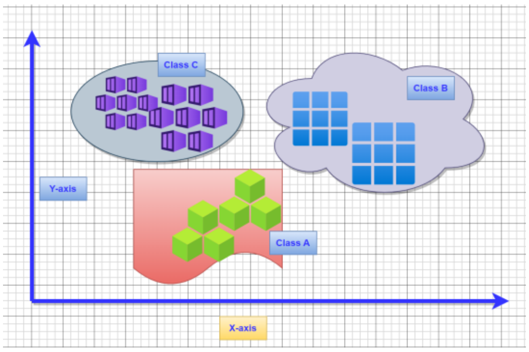


Fig. 10 Pictorial representation of SVM

The above-mentioned figure represents the behavioral aspect of k-NN classifiers. Here the items are categorized into different forms, figures, and objects. We can classify these objects by using the k-NN classifiers. Here the objects are classified as A, B, and C.

Algorithm for K-NN:

Input: A: training dataset, B: class labels of A, k: nearest neighbors in terms of numbers

Output: Class labels of x (test sample)

Step 1: loading the data points (given)

Step 2: initialize the value of A as a training dataset

Step 3: for each x do

Distance d (x, A) = $\sqrt{\sum_{i=1}^{n}(xi - Ai)^2}$

end for

Step 4: Determine the separation between every row in the training dataset and the test data. (i.e., Euclidean distance)

Step 5: Sort the calculated distances in ascending order based on the distance values.

Step 6: get the sorted array's first k rows.

Step 7: find the most common class for these rows

Step 8: give back the expected class

Step 9: end

Support Vector Machine (SVM)

A classifier called SVM can categorize both linear and nonlinear data. The main idea of the SVM classifier [7] is to first convert the data non-linearly to a sufficiently high level, for example, n, and then split the data to a higher level into subsets using a simple n-1 dimensional decision surface called a hyperplane. Learning hyperplanes in linear SVM is done using some linear algebraic transformation problem. This is where the kernel comes into play.

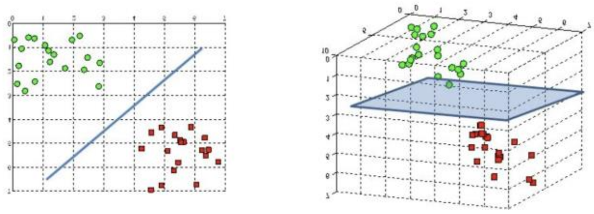


Fig. 11 working of SVM

https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47

Hyperplanes help with the data point classification by serving as decision boundaries. The data points on either side of the hyperplane can be classified into several classifications. The size of the hyperplane is also affected by the quantity of features. When there are only two input features, the hyperplane is depicted as a line. A two-dimensional plane is created when the hyperplane collapses with three input characteristics. When there are more than three aspects, it is harder to imagine.

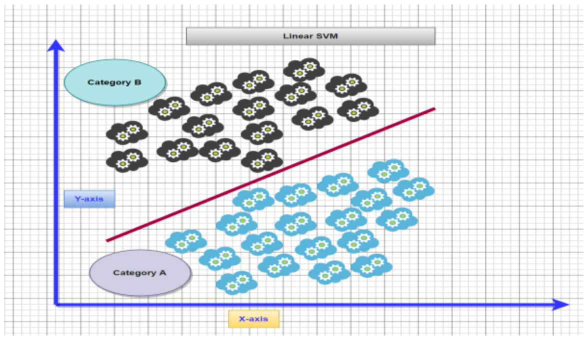


Fig. 12 Pictorial representation of SVM

The above figure clearly defines the working principle of support vector machine, which defines the execution of ideal data transformations that establish boundaries between data points based on preset classes (category A and B), labels, or outputs.

Algorithm for SVM:

A is a vector, w perpendicular to the hyperplane. The point is on the right side if the dot product is greater than "x." The point is on the left side if the dot product is smaller than or equal to "x," and it is on the decision boundary otherwise. Now, according to the rules, define the point as either positive or negative.

Step 1: Compute A.B = |A| cos Θ * |B|

Step 2: $\vec{A} \cdot \vec{w} = x$

Step 3: Compute $\vec{A} \cdot \vec{w} > x$ and $\vec{A} \cdot \vec{w} < x$

Step 4: Compute $\vec{A} \cdot \vec{w} - x \geq 0$

$\vec{A} \cdot \vec{w} + u \geq 0$ [put -x as u]

Finally,

$$B = \begin{cases} +1\ if\ \vec{A}.\vec{w} + u\ \geq 0 \\ -1\ if\ \ \vec{A}.\vec{w} + u < 0 \end{cases}$$

(If the value of A. w + u > 0 then we can say it is a positive point)

(If the value of A. w + u < 0 then we can say it is a negative point)

Step 5: To calculate the distance (say d1) we can assume the two equations are as follows:

Equ1: A. w + u= 1 and

Equ2: A. w + u= -1

Step 6: End

Artificial Neural Network (ANN)

A nonlinear model that simulates the ANN processes the information that powers the brain. To accomplish tasks like categorization, forecasting, decision-making, and visualization, among others, it can learn from big datasets. An ANN processes data within a network of nodes, which are sometimes referred to as neurons. These neurons are composed of three layers: the input layer, the output layer, and the hidden layer.

Artificial Neural Networks (ANN) [17] are algorithms modeled after the brain that are utilized to show complex patterns and foresee problems. The Artificial Neural Network (ANN), a deep learning method, is based on the hypothesis that biological neural networks resemble those seen in the human brain. In an attempt to mimic the functioning of the human brain, ANN was created. Biological neural networks and Artificial neural networks (ANNs) share numerous similarities, although not operating exactly alike. Only structured and numerical data is accepted by the ANN algorithm.

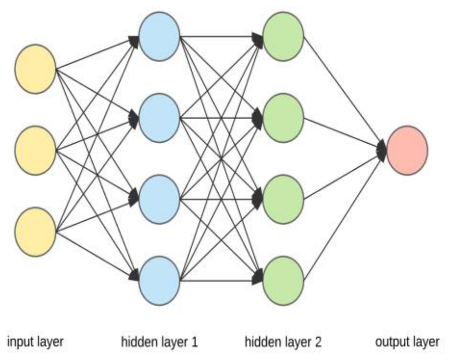


Fig. 13 ANN model in machine learning

https://towardsdatascience.com/applied-deep-learning-part-1-artificial-neural-networks-d7834f67a4f6

Ridge Classifier (Ridge)

The Ridge classification method [9] is predicated on the idea that instances of specific categories exist in a space that is linearly segmented, and that the most recent class can be administered screening of the sample, which is defined as a continuous arrangement of sample training evaluation for the relevant category. The multicollinear data analysis technique known as ridge regression has been used to enhance algorithm performance. L2 regularization is extracted using this technique. When the issue of multicollinearity emerges, the least-squares are random and the variances are considerable, resulting in predicted values that are far lower than the actual value. L1 and L2 might simplify the model and prevent overfitting caused by simple linear regression.

The Ridge classification technique is based on the subspace assumption, which states that samples of a particular class are linearly distributed and that the new test for a class will be represented as a combination of training samples from the different classes. Given its name, cord regression belongs to the latter category. According to the Sklearn Hate sheet, problems can be resolved using ridge regression when there are fewer than 100,000 samples or more parameters than samples.

Random Forest

Multiple random decision trees form a random forest [11]. This tree has two different randomizations. Each tree is initially created using a sample of the original data. Second, a random set of features is chosen for each tree node to ensure the best separation. This Random Forest technique's (a bagging-type ensemble learning technique for classification) basic operation. It creates some decision trees extracted from random subsamples of the text during the run. During testing, each decision tree predicts a fresh test document and assigns the class label that has been largely predicted by each decision tree classifier.

Algorithm for Random Forest Classifier

Step 1: Choose arbitrary samples from a training set or set of data. (i.e., to design a decision tree).

Step 2: Each decision tree is constructed using a subset of ***characteristics*** and a subset of ***data points***. (**n**: *random records*, **m**: *features*, **k**: *number of records*)

Step 3: Every ***decision tree*** will produce a ***result***.

Step 4: the results are evaluated using either an ***average*** or a ***majority vote***.

Step 5: End

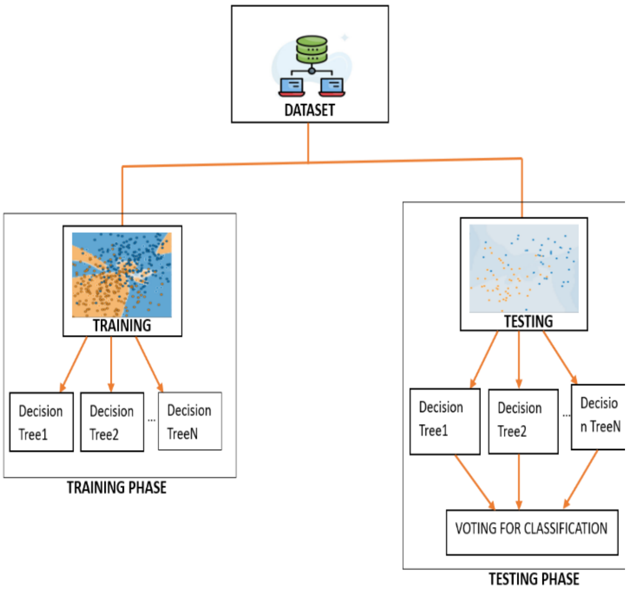


Fig. 14 Pictorial representation of Random Forest

Passive Aggressive Classifier

Learners and even intermediate learning aficionados may not be familiar with the family of Passive-Aggressive algorithms. For particular purposes, they can be quite useful and effective.

Why the Passive-Aggressive Algorithm?

The following explains the nomenclature of passive-aggressive algorithms:

- **Passive:** If the forecast is correct, keep using the model and don't make any changes. Put otherwise, there is not enough data in the case to change the framework in any way.
- **Aggressive:** If the prediction proves to be incorrect, adjust the model. Said another way, it might be corrected by changing the model.

5. Results and Discussions

5.1. Experimental Setup

The following list outlines the different types of graphics published in IEEE journals. We use different types of Python libraries for the implementation of ML algorithms. Scikit-learn python library is used to implement ML algorithms. End-to-end testing is done by Google Integration. The below table shows that we used different types of machine models for HS detection like the dataset HS.

All the machine-learning models run through different machine-learning libraries. The Scikit-Learn libraries in Python are used to implement ML computing methods. The Google Collaboratory cloud service, which has 2496 CUDA cores, a GPU with an integrated Tesla K20, 16GB of RAM, and a 250GB hard drive, is used for all aspects of the testing process. On a laptop running Windows 11 and equipped with an Intel Core i5-11320H processor from the 11th generation, the experiments detailed in this paper were performed. This CPU has a 3.20 GHz clock speed.

Performance Evaluation: The results of the classifier are evaluated using the following measures, to understand how well the classifier performed:

a) Accuracy [45], [14], [8], [7] is the ratio of the correct classified entities to the total number of the dataset.

$$\text{Accuracy} = \frac{\text{Correct Predictions}}{\text{Total Number}} \tag{2}$$

b) Precision [14], [45], [7], [8] is the ratio of the true positive results to the total number of positives.

$$\text{Precision} = \frac{Total\ True\ Positives}{True\ Positives + False\ Positives} \tag{3}$$

c) Recall [8], [7], [14], [45] is the number of the correct classified entity from one class divided by the total entity number of that class

$$\text{Recall} = \frac{TotalTruePositives}{TruePositives + FalseNegatives} \tag{4}$$

d) It is about measuring the test accuracy, the greater the f1 score [45], [14], [8], [7] the better the model is. f1-measure uses both the recall and the precision scores.

$$\text{f1-score} = 2 * \frac{(Recall * Precision)}{(Recall + Precision)} \tag{5}$$

Table 2: Performance of ML Algorithm

Classification Algorithm	**Performance Measure (Mean ± Deviation)**				
	Accuracy	*Precision*	*Recall*	*F1-Score*	*Execution Time*
KNN	0.561±0.029	0.587±0.038	0.561±0.030	0.543±0.031	523.42s
DT	0.577±0.070	0.578±0.072	0.577±0.070	0.574±0.073	122.414s
MNB	0.620±0.118	0.621±0.128	0.620±0.117	0.613±0.123	3.121s
BNB	0.587±0.173	0.597±0.170	0.587±0.173	0.580±0.175	4.08s
RF	0.628±0.120	0.636±0.135	0.628±0.120	0.617±0.127	784.827s

SVM	0.636±0.113	0.642±0.125	0.636±0.113	0.625±0.121	7305.42s
PPN	0.602±0.083	0.603±0.085	0.602±0.083	0.600±0.087	4.133s
SGD	0.620±0.100	0.628±0.111	0.620±0.100	0.603±0.108	6.352s
Ridge	0.630±0.097	0.630±0.104	0.630±0.097	0.623±0.104	13.16s
RC	0.588±0.106	0.596±0.119	0.588±0.106	0.580±0.112	4.289s
PA	0.607±0.080	0.610±0.081	0.607±0.080	0.605±0.083	8.467s
BPN	0.611±0.097	0.612±0.099	0.611±0.097	0.609±0.101	1085.024s

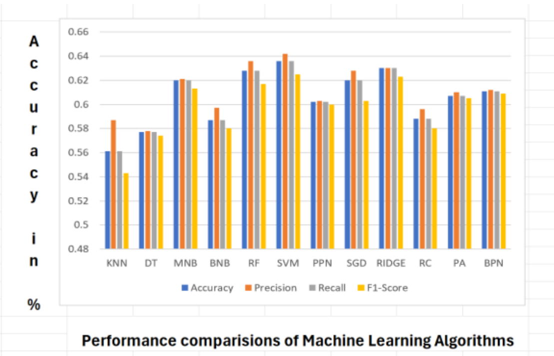


Fig. 15 Comparison of ML Algorithm

The table 2 displays the effectiveness of the various machine-learning algorithms. Accuracy, precision, recall, and F1-score are the four metrics that can be used to measure performance. This study uses several machine learning (ML) algorithms. Based on the data presented, the SVM classifier has greater accuracy than the other classifiers. SVM classifier accuracy is 0.636±0.113 and Ridge classifier accuracy is 0.630±0.097. After Ridge, the RF and SGD classifiers have the same accuracy, followed by the MNB and BPN classifiers. If all classifiers are compared concerning the bar graph depicted in the above figure, Ridge and SVM classifiers demonstrate the best classification performance. Among all classifiers for the HS dataset, the KNN classifier achieves the lowest classification accuracy. In the meantime, the remaining classifiers provide an average classification performance.

6. Conclusion

Hate speech is a serious issue that can cause harm and perpetuate discrimination against certain individuals or groups. The widespread dissemination of hate speech on social media platforms has led to negative impacts on society. This research paper has focused on the detection of hate speech in the Odia language. We have explored various techniques and methods. It is used to identify distinctive words in documents, including machine learning algorithms and language processing techniques. The results of this research demonstrate that can detect hate speech in the Odia language with a reasonable level of accuracy. By analyzing the text data, we have identified several key features that can be used to differentiate HS and non-HS text. Overall, this research highlights the importance of addressing hate speech and promoting a culture of tolerance and inclusivity in our society. It is essential to continue to monitor and fight hate speech to create a safer and more harmonious world for everyone.

Author contributions:

1. **Aloka Natha:** Conceptualization, Methodology, Software, Field study
2. **Dr. Bichitrananda Behera:** Data curation, Writing-Original draft preparation, Software, Validation., Field study
3. **Debaswapna Mishra:** Visualization, Investigation, Writing-Reviewing and Editing.
4. **Saumya Ranjan Sahu:** Documentation
5. **Subhasis Mohapatra:** Analysis

References

[1] S. Sahoo, C. S. Panda, and S. Patnaik, “Part of Speech Tagging in Odia Using Support Vector Machine,” Procedia Computer Science, vol. 48, pp. 507–512, Jan.2015, doi:10.1016/. procs.2015.04.127

[2] S. K. Mohapatra, S. Prasad, D. K. Bebarta, T. K. Das, K. Srinivasan, and Y.-C. Hu, "Automatic Hate Speech Detection in English-Odia Code Mixed Social Media Data Using Machine Learning Techniques," *Applied Sciences*, Vol. 11, no.18,p. 8575, Aug.2021,doi:10.3390/app11188575.

[3] A. M. Ishmam and S. Sharmin, "Hateful Speech Detection in Public Facebook Pages for the Bengali Language, "2019 18th IEEE International Conference on Machine Learning and Application (ICMLA), Boca Raton, FL, USA,2019, pp.555-560, doi: 10.1109//ICMLA.2019.00104

[4] K. Nugroho et al., “Improving Random Forest Method to Detect Hate Speech and Offensive Word,” 2019 International Conference on Information and

Communication Technology (*ICOIACT*), Yogyakarta, Indonesia,2019, pp.514-518, doi: 10.1109/ICOIACT46704.2019.8938451.

[5] N. Atashfaraz “SHORT-TERM WIND SPEED FORECASTING USING DEEP VARIATIONAL LSTM,” Dec.01.2022 http://82.194.3.83:12123/jspui/handel/123456789/128

[6] A.A.-H and H. Ai-Dossari, “DETECTION OF HATE SPEECH IN SOCIAL NETWORKS: A SURVEY ON MULTINOUAL CORPUS,” Computer Science & Information Technology (CS & IT). Feb. 2019. Doi: 10.512 /csit.2019.90208.

[7] B. Behera, G. Kumaravelan, and p. Kumar “Performance Evolution of Deep Learning Algorithm in Biomedical Document Classification,” International Conference on Advanced Computing. Dec 2019, Doi: 10.1109/icoac48765.2019.246843

[8] Saroj, A.; Pal, S. An Indian language social media collection for hate and offensive speech. In Proceedings of the Workshop on Resources and Techniques for User and Author Profiling in Abusive Language, Marseille, France, 11–16 May 2020; pp. 2–8.

[9] Mossie, Z.; Wang, J.H. Social network hate speech detection for Amharic language. In Proceedings of the 6th International Conference on Computer Science and Information Technology, Copenhagen, Denmark, 28–29 April 2018; pp. 41–55.

[10] Ibrohim, M.O.; Budi, I. A dataset and preliminaries study for abusive language detection in Indonesian social media. Procedia Comp. Sci. 2018, 135, 222–229. [CrossRef]

[11] Davidson, T.; Warmsley, D.; Macy, M.; Weber, I. Automated hate speech detection and the problem of offensive language. In Proceedings of the Eleventh International AAAI Conference on Web and Social Media, Montreal, QC, Canada, 15–18 May 2017.

[12] Gambäck, B.; Sikdar, U.K. Using convolutional neural networks to classify hate-speech. In Proceedings of the First Workshop on Abusive Language Online, Vancouver, BC, Canada, 30 July–4 August 2017; pp. 85–90.

[13] Benikova, D.; Wojatzki, M.; Zesch, T. Examining the Impact of Implicitness on the Perception of Hate Speech. In International Conference of the German Society for Computational Linguistics and Language Technology; Springer: Cham, Switzerland, 2017; pp. 171–179.

[14] Tripathy, S. S., & Behera, B. (2023). Performance Evaluation of Machine Learning Algorithms for Intrusion Detection System. Journal of Biomechanical Science and Engineering, 621-640.

[15] Pattnaik, Sagarika & Nayak, Ajit. (2020). A SIMPLE AND EFFICIENT TEXT SUMMARIZATION MODEL FOR ODIA TEXT DOCUMENTS. Indian Journal of Computer Science and Engineering. 11. 825-834. 10.21817/indjcse/2020/v11i6/201106132.

[16] Mishra, B. K., & Sahoo, R. (2018). A hybrid knowledge mining approach to develop a system framework for Odia language text processing. Materials Today: Proceedings, 5(1), 1335-1340.

[17] M.K, Jena & Mohanty, Sanghamitra. (2019). Contextual Opinion mining in online Odia Text using 'Support Vector Machine. 10.13140/RG.2.2.35968.35845.

[18] Dey, G & Maringanti, H. (2023). SOLMAT: A Neoteric Contextual Model for Odia Language Understanding. Journal of Physics: Conference Series. 2571. 012011. 10.1088/1742-6596/2571/1/012011.

[19] Mohanty, G., Kannan, A., & Mamidi, R. (2017, September). Building a sentiwordnet for Odia. In *Proceedings of the 8th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis* (pp. 143-148).

[20] Mohanty, G., Mishra, P., & Mamidi, R. (2020, May). Annotated corpus for sentiment analysis in Odia language. In *Proceedings of the Twelfth Language Resources and Evaluation Conference* (pp. 2788-2795).

[21] U. Parida, M. Nayak and A. K. Nayak, "Ranking of Odia Text Document Relevant to User Query Using Vector Space Model," 2019 International Conference on Applied Machine Learning (ICAML), Bhubaneswar, India, 2019, pp. 165-169, doi: 10.1109/ICAML48257.2019.00039.

[22] S. Pattnaik and A. K. Nayak, "Summarization of Odia Text Document Using Cosine Similarity and Clustering," 2019 International Conference on Applied Machine Learning (ICAML), Bhubaneswar, India, 2019, pp. 143-146, doi: 10.1109/ICAML48257.2019.00035.

[23] Jena, M. K., & Mohanty, S. (2020). Predicting the Impact of Odia Newspaper Articles on Public Opinion. In Progress in Computing, Analytics and Networking: Proceedings of ICCAN 2019 (pp. 265-272). Springer Singapore.

[24] Das, Bishwa & Sahoo, Rekhanjali & Singh, Dilip & Bhoi, Prakash Chandra. (2023). Odia Text Classification for Sentiment Analysis using K Nearest Neighbor.

[25] Fauzi, M.A.; Yuniarti, A. Ensemble method for Indonesian Twitter hate speech detection. Indones. J. Electr. Eng. Comput. Sci. 2018, 11, 294–299. [CrossRef]

[26] Fiok, K.; Karwowski, W.; Gutierrez, E.; Liciaga, T.; Belmonte, A.; Capobianco, R. Automated Classification of Evidence of Respect in the Communication through Twitter. Appl. Sci. 2021, 11, 1294. [CrossRef]

[27] Alshalan, R.; Al-Khalifa, H. A Deep Learning Approach for Automatic Hate Speech Detection in the Saudi Twittersphere. Appl.Sci. 2020, 10, 8614. [CrossRef]

[28] Pereira-Kohatsu, J.C.; Quijano-Sánchez, L.; Liberatore, F.; Camacho-Collados, M. Detecting and monitoring hate speech in Twitter. Sensors 2019, 19, 4654. [CrossRef]

[29] Sahoo, Rekhanjali & Mishra, Brojo & Das, Bishwa. (2022). Odia Text Classification Using Naïve Bayes Algorithm: An Empirical Study. ECS Transactions. 107. 8175-8180. 10.1149/10701.8175ecst.

[30] 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. [Google Scholar] [CrossRef]

[31] Sánchez-Compaña, M.T.; Sánchez-Cruzado, C.; García-Ruiz, C.R. An interdisciplinary scientific and mathematics education, addressing relevant social problems such as sexist hate speech. Information 2020, 11, 543. [Google Scholar] [CrossRef]

[32] Sanoussi, M.S.A.; Xiaohua, C.; Agordzo, G.K.; Guindo, M.L.; al Omari, A.M.M.A.; Issa, B.M. Detection of Hate Speech Texts Using Machine Learning Algorithm. In Proceedings of the 2022 IEEE 12th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, USA, 26–29 January 2022; pp. 266–273. [Google Scholar] [CrossRef]

[33] 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. Comp. Sci. Rev. 2020, 38, 100311. [Google Scholar] [CrossRef]

[34] Corazza, M.; Menini, S.; Cabrio, E.; Tonelli, S.; Villata, S. A Multilingual Evaluation for Online Hate Speech Detection. ACM Trans. Internet Technol. 2020, 20. [Google Scholar] [CrossRef] [Green Version]

[35] Das, Bishwa & Maringanti, Hima & Dash, Niladri. (2022). Word Alignment in Bilingual Text for Bangla to Odia Machine Translation.

[36] Jahan, M.S.; Oussalah, M. A systematic review of Hate Speech automatic detection using Natural Language Processing. arXiv 2021, arXiv:2106.00742. [Google Scholar]

[37] Qureshi, K.A.; Sabih, M. Un-Compromised Credibility: Social Media Based Multi-Class Hate Speech Classification for Text. IEEE Access 2021, 9, 109465–109477. [Google Scholar] [CrossRef]

[38] Siddiqua, U.A.; Chy, A.N.; Aono, M. KDEHatEval at SemEval-2019 Task 5: A Neural Network Model for Detecting Hate Speech in Twitter. Proceedings of the 13th International Workshop on Semantic Evaluation, Minneapolis, MN, USA, 6–7 June 2019; pp. 365–370. [Google Scholar] [CrossRef]

[39] Sachdeva, J.; Chaudhary, K.K.; Madaan, H.; Meel, P. Text-Based Hate-Speech Analysis. In Proceedings of the International Conference on Artificial Intelligence and Smart Systems, ICAIS, Tamilnadu, India, 25–27 March 2021; pp. 661–668. [Google Scholar] [CrossRef]

[40] Tontodimamma, A.; Nissi, E.; Sarra, A.; Fontanella, L. Thirty years of research into hate speech: Topics of interest and their evolution. Scientometrics 2021, 126, 157–179. [Google Scholar] [CrossRef]

[41] Shibly, F.H.A.; Sharma, U.; Naleer, H.M.M. Classifying and Measuring Hate Speech in Twitter Using Topic Classifier of Sentiment Analysis; Springer: Singapore, 2021; Volume 1165. [Google Scholar] [CrossRef]

[42] Ullmann, S.; Tomalin, M. Quarantining online hate speech: Technical and ethical perspectives. Ethics Inf. Technol. 2020, 22, 69–80. [Google Scholar] [CrossRef][Green Version]

[43] Mathew, B.; Saha, P.; Yimam, S.M.; Biemann, C.; Goyal, P.; Mukherjee, A. HateXplain: A Benchmark Dataset for Explainable Hate Speech Detection. Proc. AAAI Conf. Artif. Intell. 2021, 35, 14867–14875. [Google Scholar]

[44] Culpeper, J. Impoliteness and hate speech: Compare and Contrast. J. Pragmat. 2021, 179, 4–11. [Google Scholar] [CrossRef]

[45] Watanabe, H.; Bouazizi, M.; Ohtsuki, T. Hate speech on Twitter: A pragmatic approach to collect hateful and offensive expressions and perform hate speech detection. IEEE Access 2018, 6, 13825–13835. [CrossRef]