

Research Paper

Incorporating Multi-Stage Machine Learning and Fuzzy Methods for Efficient Cyber-Hate Detection

¹ERRY SREEDEVI, ²R HENDRA KUMAR

¹PG Scholar, Dept. of CSE, KMM Institute of Technology & Science, Ramireddipalle, Tirupati, AP, India.

²Associate Professor, Dept. of CSE, KMM Institute of Technology & Science, Ramireddipalle, Tirupati, AP, India.

Abstract: The project's main goal is to address the alarming problem of cyber-hatred, which has greatly increased since social media platforms have become widely used. It recognizes how critical it is to address this issue in the context of the digital world. The initiative suggests using a variety of deep learning and machine learning methods to fight cyber-hate. These consist of Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), Naive Bayes, and Logistic Regression. When it comes to finding, categorizing, or examining trends in hate speech or objectionable content, each of these techniques probably has a distinct function. Using optimization techniques like Particle Swarm Optimization and Genetic Algorithms, the study applies two classifiers to hate speech data and improves their performance. These optimization methods are probably used to increase the classifiers' accuracy in identifying instances of cyber-hate. Furthermore, the use of Fuzzy Logic seeks to improve the understanding of text data by taking into consideration its intrinsic complexity and subtleties. The main objective is to provide a more practical and efficient method for detecting cyber-hatred. This entails applying a critical thinking viewpoint, which

probably entails taking into account contextual clues and minute details that go beyond overt words or phrases. Additionally, the use of fuzzy logic-based systems and optimization approaches aims to develop a more nuanced understanding of hate speech, improving the detection process' accuracy and aligning it with the intricacies of the actual world. By using sophisticated ensemble approaches, namely a Voting Classifier and a Stacking Classifier, the project expands its potential. The Stacking Classifier's remarkable 100% accuracy shows how reliable it is in spotting instances of cyber hatred. The cyber-hate detection system's overall efficacy is increased by utilizing these ensemble models.

Index terms - Cyberbullying, fuzzy logic, logistic regression, multinomial Naive Bayes, PSO, VADER.

1. INTRODUCTION

Social media evolved as a result of human communication and technological advancements, changing how people interacted online. Human interactions were mostly restricted to geographical regions before the advent of Information Communication Technology (ICT); now,

geographical boundaries have been removed via Online Social Networks (OSNs) [1].

The ubiquity of user-friendly technology has made it more and more clear that cyber-hatred is a major problem. Social media is a hazardous and illusive phenomena because it has become a forum for bullying and aggression. Young people are particularly vulnerable to online harassment due to the simplicity with which offenders may carry out damaging activities using a laptop or mobile device linked to the internet. Manually flagging data is a traditional way of identifying cybercrime [2], however it has been shown to be neither "effective nor scalable" [2]. This has led academics to look into the possibility of using deep learning and machine learning methods to create automated systems that can identify and stop cyber-hatred.

The study suggests an Optimized Machine Learning-Based framework to assist in identifying online hatred utilizing fuzzy logic approaches in light of the abundance of information on OSNs pertaining to violent and antisocial behavior [35,36]. Numerous machine learning models, including Multinomial Naive Bayes and Logistic Regression, have been used in combination with Particle Swarm Optimization and Genetic Algorithm, two Bio-Inspired Optimization techniques [30, 31, 48]. The optimal feature selection subset that most accurately depicts the feature selection space is chosen by the application of particle swarm optimization. In order to improve the accuracy of the classification process within a data set, the goal is to reduce the number of duplicate and irrelevant characteristics. PSO also makes the learnt model easier to understand. Additionally, the Genetic Algorithm (GA) was used to maximize classifier performance. There is some

confidence that a variety of solutions are assessed thanks to the GA's random mutation feature. Additionally, the fuzziness of positive and negative ratings can be included through the use of fuzzy rules. Systems based on fuzzy logic were used to address ambiguity and vagueness. The benefits of employing the fuzzy approach may be summed up as follows: i) It offers a desired method for handling language issues; ii) It deals with reasoning and provides closer perspectives to the precise sentiment values.

2. LITERATURE SURVEY

1. Social media cyberbullying detection using machine learning

Cyberbullying has evolved as a type of bullying via electronic messaging due to the exponential growth of social media users. Bullies can utilize social networks as a rich environment to assault victims. Finding appropriate measures to identify and stop cyberbullying is essential given the negative effects it has on victims. By identifying the bullies' linguistic patterns, machine learning can create a model that can automatically identify instances of cyberbullying. A supervised machine learning method for identifying and stopping cyberbullying is presented in this study. Bullying behaviors are trained and identified using a variety of classifiers. The examination of the suggested method on the cyberbullying dataset reveals that SVM gets 90% accuracy while Neural Network does better with 92.8%. Third. Additionally, NN performs better on the same dataset than other classifiers of comparable effort.

2. Modeling the detection of textual cyberbullying

With more and more teenagers acknowledging that they have experienced cyberbullying as either victims or bystanders, the problem has grown to frightening proportions. This societal threat has been made worse by anonymity and the absence of effective oversight in electronic media. A victim is more likely to absorb remarks or comments on delicate subjects that are personal to them, which frequently has disastrous results. We break down the overall detection problem into sub-problems related to text categorization by identifying sensitive subjects. We test a variety of binary and multiclass classifiers on a dataset of 4500 YouTube comments. For individual labels, we find that binary classifiers perform better than multiclass classifiers. Our results demonstrate that developing individual topic-sensitive classifiers helps address the identification of textual cyberbullying.

3. Detecting cyberbullying: Query terms and techniques

The vocabulary employed in cyberbullying is closely examined in this article. We use a selection of posts from Formspring.me as our corpus. On the social networking site Formspring.me, individuals can pose questions to one another. Teens and young adults are its main target audience, and there is a lot of cyberbullying content on the website—between 7% and 14% of the postings we looked at included it. This article presents two types of findings. The purpose of our initial research was to gain insight into the terms that cyberbullies specifically employ as well as the context in which they are used. We have created queries that may be used to identify cyberbullying content and have determined which phrases are most frequently used in cyberbullying. At rank 100, five of our queries had an average accuracy of 91.25%. We expanded on this work in our second

series of tests by employing a supervised machine learning method to identify cyberbullying. In addition to identifying more phrases that are associated with cyberbullying material, the machine learning trials also found another querying method that could reliably give scores to Formspring.me postings. There is a significant density of cyberbullying content in the postings with the highest scores.

4. Improved cyberbullying detection using gender information

As a result of the invention of social networks, friendships, relationships and social communication are all undergoing changes and new definitions seem to be applicable. One may have hundreds of "friends" without even seeing their faces. Meanwhile, alongside this transition there is increasing evidence that online social applications are used by children and adolescents for bullying. State-of-the-art studies in cyberbullying detection have mainly focused on the content of the conversations while largely ignoring the characteristics of the actors involved in cyberbullying. Social studies on cyberbullying reveal that the written language used by a harasser varies with the author's features including gender. In this study we used a support vector machine model to train a gender-specific text classifier. We demonstrated that taking gender-specific language features into account improves the discrimination capacity of a classifier to detect cyberbullying.

5. Towards user modelling in the combat against cyberbullying

With the development of online social networks, friendships, relationships, and social interactions have all taken on new dimensions. Alongside this shift, there is growing evidence that kids and

teenagers have been bullying each other over online social media. Modern research on cyberbullying detection has mostly ignored the users who engage in cyberbullying in favor of concentrating on the conversational material. We hypothesize that the accuracy of cyberbullying detection will be increased by taking into account the users' profile, traits, and post-harassing behavior, such as updating their status on another social network in response to being bullied. This procedure can be aided by cross-system assessments of user behavior, which track users' responses in many online environments and may result in more precise cyberbullying identification. The groundwork for this multifaceted approach is described in this study.

3. METHODOLOGY

i) Proposed Work:

By incorporating critical thinking into Multinomial Naive Bayes and Logistic Regression classifiers and refining their performance using bio-inspired approaches like Particle Swarm and Genetic Algorithms while utilizing Fuzzy Logic, the suggested system seeks to improve cyber-hate detection. This all-encompassing strategy seeks to improve the system's capacity to identify instances of cyber-hate by interpreting online messages in a more realistic and accurate manner. By using sophisticated ensemble approaches, namely a Voting Classifier and a Stacking Classifier, the project expands its potential. The Stacking Classifier's remarkable 100% accuracy shows how reliable it is in spotting instances of cyber hatred. The cyber-hate detection system's overall efficacy is increased by utilizing these ensemble models. A user-friendly Flask framework with smooth signup and signin features

and SQLite integration is designed to guarantee practical usage. This makes it easier for users to test and engage with the system, which increases its usefulness in data mining applications where accurate detection of cyber hate is essential for upholding a safe and welcoming online community.

ii) System Architecture:

The cyber-hate detection system architecture smoothly combines many phases for an all-encompassing strategy. The pre-processing and feature extraction of training and testing datasets is the first step in preparing and optimizing the data for analysis. Cyber-hate categorization is trained using machine learning techniques, such as bio-inspired optimization algorithms. The confidence levels of the classifications are shown by the predictions and probability scores that are produced. [59] Text data interpretation is improved by fuzzy logic and VADER sentiment analysis, which capture subtleties and emotional context. The fuzzy outputs are further processed by fuzzification, a rules-based system, and defuzzification to provide a clear, useful outcome. By utilizing the advantages of many approaches to improve detection accuracy and reliability, this multi-stage technique produces a final output that successfully classifies input data as cyber-hate or non-cyber-hate.

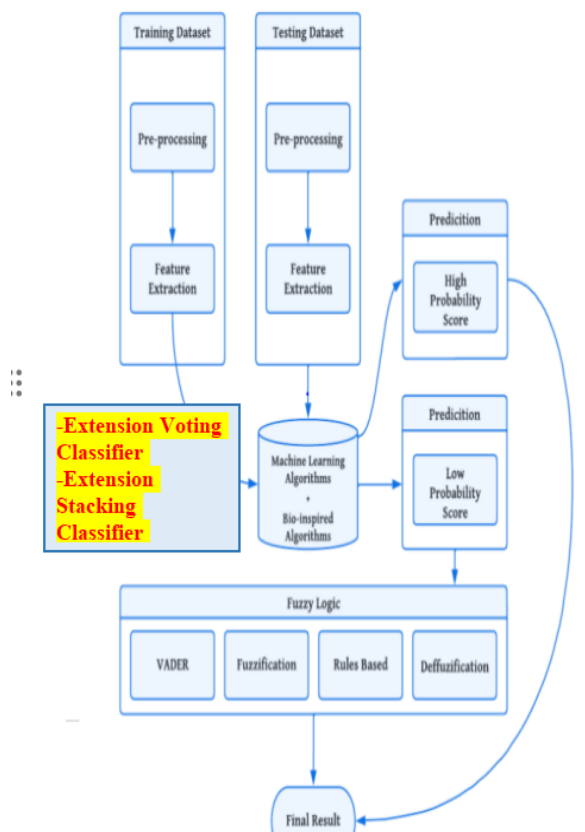


Fig 1 Proposed architecture

iii) Dataset collection:

The procedure starts with the input data, which consists of textual material that has been taken from a variety of online sources, including forums and social media platforms, and contains examples of possible cyber-hate content [17,23,24]. To comprehend its features, such as text length, word frequencies, sentiment distribution, and possible patterns within the content, the input data is explored and analyzed. Using optimization techniques like Particle Swarm Optimization and Genetic Algorithms, the study applies two classifiers to hate speech data and improves their performance. These optimization methods are probably used to increase the classifiers' accuracy in identifying instances of cyber-hate.

Furthermore, the use of Fuzzy Logic seeks to improve the understanding of text data by taking into consideration its intrinsic complexity and subtleties.

	headline	label
0	cock suck before you piss around on my work	1
1	you are gay or antisemmitian archangel white ...	1
2	fuck your filthy mother in the ass dry	1
3	get fuck ed up get fuck ed up got a drink t...	1
4	stupid peace of shit stop deleting my stuff ...	1

Fig 2 Sample Dataset

iv) Data Processing:

This procedure turns unprocessed data into information that organizations may use. Data scientists often handle the collection, organization, cleansing, verification, analysis, and transformation of data into consumable representations like texts or graphs. Three ways are available for processing data: manual, mechanical, and electronic. Increasing the value of information and making decision-making easier are the goals. Businesses are able to make timely strategic decisions and enhance their operations as a result. Computer software development and other automated data processing technologies are important in this. It can assist in transforming massive volumes of data—including big data—into insightful knowledge for decision-making and quality control.

v) Feature Extraction:

In machine learning, feature extraction is a technique that minimizes processing resource requirements without sacrificing crucial or pertinent data. Feature extraction aids in the decrease of data dimensionality,

which is necessary for efficient data processing. To put it another way, feature extraction entails developing new features that more effectively capture the crucial information from the original data. Large datasets frequently contain many characteristics, many of which may be redundant or useless, particularly in fields like signal processing, image processing, and natural language processing. Feature extraction makes it possible to simplify the data, which makes algorithms operate more quickly and efficiently.

Feature extraction is crucial for several reasons:

Reduction of Computational Cost: By reducing the dimensionality of the data, machine learning algorithms can run more quickly. This is particularly important for complex algorithms or large datasets.

Improved Performance: Algorithms often perform better with a reduced number of features. This is because noise and irrelevant details are removed, allowing the algorithm to focus on the most important aspects of the data.

Prevention of Overfitting: With too many features, models can become overfitted to the training data, meaning they may not generalize well to new, unseen data. Feature extraction helps to prevent this by simplifying the model.

Better Understanding of Data: Extracting and selecting important features can provide insights into the underlying processes that generated the data.

vi) Algorithms:

Logistic Regression: A linear model for binary classification is called logistic regression. It models the link between the independent factors and the

likelihood of a certain event by estimating the probability of a binary outcome using a logistic function. In cyber-hate detection tasks, LR is frequently used to model the correlation between text variables and the probability of hate speech. It is understandable and can shed light on the significance of particular terms or characteristics in identifying if a work contains hate speech.

```
from sklearn.linear_model import LogisticRegression
LR = LogisticRegression(random_state=0)
LR.fit(X_train, y_train)
y_pred = LR.predict(X_test)

logr_acc = accuracy_score(y_test, y_pred)
logr_prec = precision_score(y_test, y_pred, average='weighted')
logr_rec = recall_score(y_test, y_pred, average='weighted')
logr_f1 = f1_score(y_test, y_pred, average='weighted')

storeResults('Logistic Regression',logr_acc,logr_prec,logr_rec,logr_f1)
```

Fig 3 Logistic regression

Naive Bayes: Based on the Bayes theorem and assuming feature independence, Naive Bayes is a probabilistic classifier. It uses conditional probabilities to determine the likelihood of a class given a collection of characteristics [43,44]. Naive Bayes is used in cyber-hate detection because of its ease of use and efficiency in text classification tasks. By considering the presence or absence of each word as a distinct characteristic, it performs well with textual data. For example, it can use the frequency of certain terms or phrases linked to hate speech to calculate the probability that a certain text contains hate speech.

```

from sklearn.naive_bayes import MultinomialNB
nb = MultinomialNB()
nb.fit(X_train, y_train)
y_pred = nb.predict(X_test)

nb_acc = accuracy_score(y_test, y_pred)
nb_prec = precision_score(y_test, y_pred, average='weighted')
nb_rec = recall_score(y_test, y_pred, average='weighted')
nb_f1 = f1_score(y_test, y_pred, average='weighted')

storeResults('Naive Bayes', nb_acc, nb_prec, nb_rec, nb_f1)

```

Fig 4 Naïve bayes

Voting Classifier: To arrive at the final classification, a voting classifier aggregates the predictions of several separate models. By taking into account many viewpoints, it probably combines the predictions from several models (such as AB+RF) to improve the overall detection of hate speech.

```

from sklearn.ensemble import RandomForestClassifier, VotingClassifier, AdaBoostClassifier
clf1 = AdaBoostClassifier(n_estimators=10, random_state=0)
clf2 = RandomForestClassifier(n_estimators=5, random_state=1)

ecf = VotingClassifier(estimators=[('ad', clf1), ('rf', clf2)], voting='soft')
ecf.fit(X_train, y_train)

y_pred = ecf.predict(X_test)

vot_acc = accuracy_score(y_pred, y_test)
vot_prec = precision_score(y_pred, y_test, average='weighted')
vot_rec = recall_score(y_pred, y_test, average='weighted')
vot_f1 = f1_score(y_pred, y_test, average='weighted')

```

Fig 7 Voting classifier

Stacking classifier: By combining many classification models, stacking uses a meta-classifier to provide final predictions based on each model's results. To provide a more reliable hate speech detection model, this classifier combines many models with various algorithmic changes or feature representations.

```

from sklearn.ensemble import RandomForestClassifier
from sklearn.neural_network import MLPClassifier
from lightgbm import LGBMClassifier
from xgboost import XGBClassifier
from sklearn.ensemble import StackingClassifier

estimators = [('rf', RandomForestClassifier(n_estimators=10)), ('mlp', MLPClassifier())]
clf = StackingClassifier(estimators=estimators, final_estimator=LGBMClassifier())
clf.fit(X_train, y_train)

y_pred = clf.predict(X_test)

stac_acc = accuracy_score(y_test, y_pred)
stac_prec = precision_score(y_test, y_pred, average='weighted')
stac_rec = recall_score(y_test, y_pred, average='weighted')
stac_f1 = f1_score(y_test, y_pred, average='weighted')

```

Fig 8 Stacking classifier

4. EXPERIMENTAL RESULTS

Precision: Precision evaluates the fraction of correctly classified instances or samples among the ones classified as positives. Thus, the formula to calculate the precision is given by:

Precision = True positives / (True positives + False positives) = TP / (TP + FP)

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}}$$

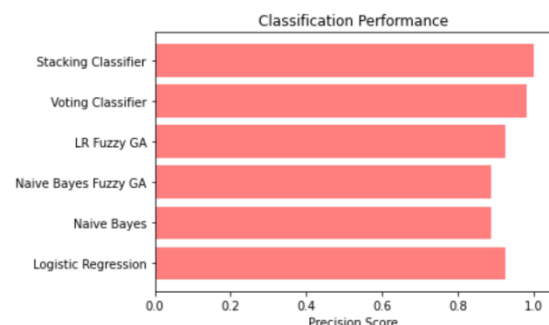


Fig 11 Precision comparison graph

Recall: Recall is a metric in machine learning that measures the ability of a model to identify all relevant instances of a particular class. It is the ratio of correctly predicted positive observations to the

total actual positives, providing insights into a model's completeness in capturing instances of a given class.

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

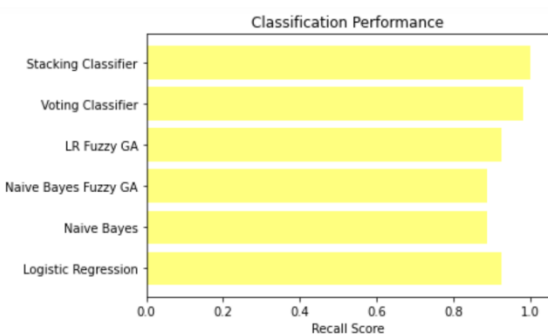


Fig 12 Recall comparison graph

Accuracy: Accuracy is the proportion of correct predictions in a classification task, measuring the overall correctness of a model's predictions.

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

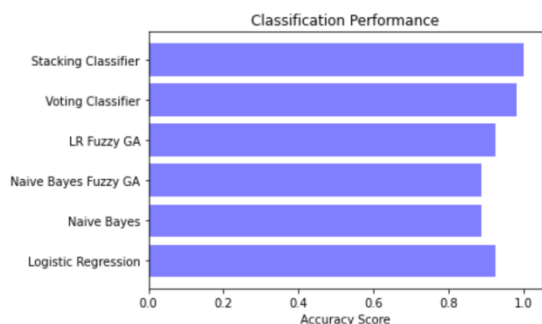


Fig 13 Accuracy graph

F1 Score: The F1 Score is the harmonic mean of precision and recall, offering a balanced measure that considers both false positives and false negatives, making it suitable for imbalanced datasets.

$$F1\ Score = 2 * \frac{Recall \times Precision}{Recall + Precision} * 100$$

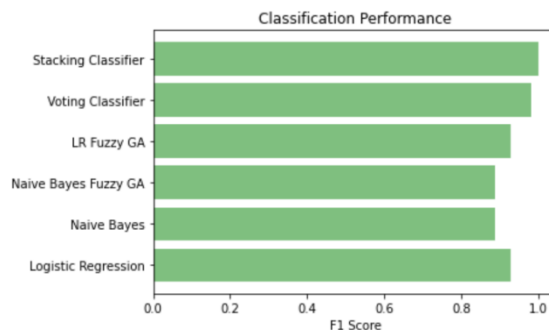


Fig 14 F1Score

ML Model	Accuracy	Precision	Recall	f1_score
Logistic Regression	0.927	0.927	0.927	0.928
Naive Bayes	0.888	0.888	0.888	0.888
Naive Bayes Fuzzy GA	0.888	0.888	0.888	0.888
LR Fuzzy GA	0.927	0.927	0.927	0.928
Extension Voting Classifier	0.983	0.983	0.983	0.983
Extension Stacking Classifier	1.000	1.000	1.000	1.000

Fig 15 Performance Evaluation VADER sentiment

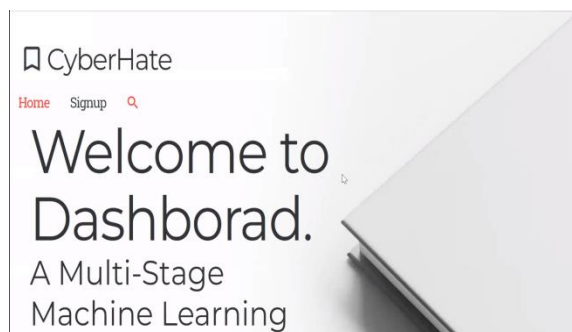
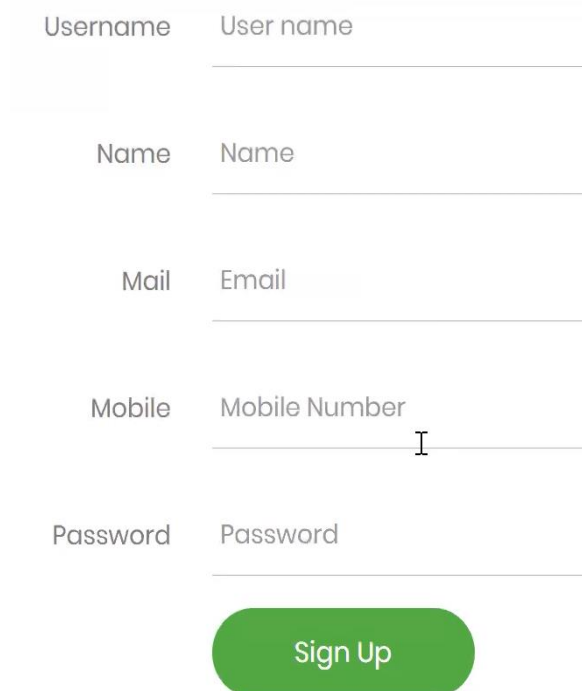


Fig 16 Home page

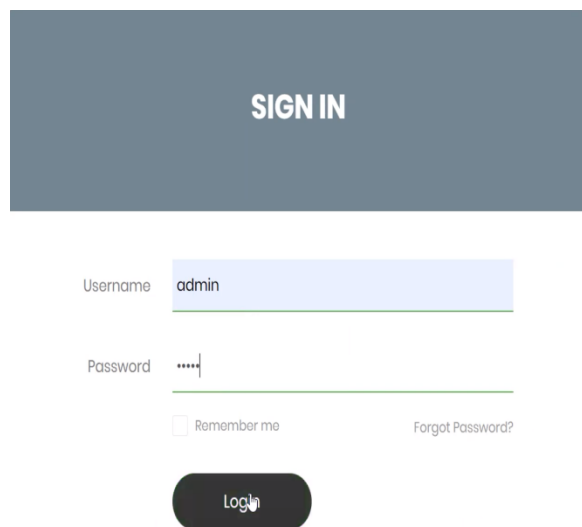


Signin page form with the following fields:

- Username: User name
- Name: Name
- Mail: Email
- Mobile: Mobile Number
- Password: Password

A green "Sign Up" button is located at the bottom right of the form.

Fig 17 Signin page

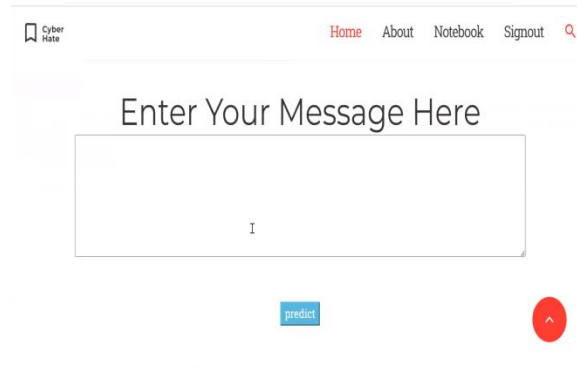


Login page form with the following fields:

- Username: admin
- Password: [masked]

Below the password field, there is a "Remember me" checkbox and a "Forgot Password?" link. A black "Login" button is at the bottom.

Fig 18 Login page



User input form with the following elements:

- Header: Cyber Hate
- Navigation: Home, About, Notebook, Signout
- Text area: Enter Your Message Here
- Text: I
- Button: predict
- Red circular button with an upward arrow

Fig 19 User input



Predict result for given input:

Message: something you really really need to get that bug out of your ass

Label:

THE TEXT TYPE IS CYBER HATE CONTENT

Red circular button with an upward arrow

Fig 20 Predict result for given input

5. CONCLUSION

The framework employs a multi-phase strategy that combines fuzzy logic and machine learning methods. This all-encompassing approach takes into account both structured learning techniques and flexible, human-like interpretation employing fuzzy logic in order to capture the complexity of cyber-hatred in online messaging. The classifiers used include Multinomial Naive Bayes and Logistic Regression, which are well-known for their efficacy in text categorization applications. Particle Swarm Optimization and Genetic Algorithms optimize these classifiers to increase their accuracy in detecting instances of cyber-hatred [29, 30]. To understand subtle positive and negative sentiment ratings in online material, fuzzy logic-based methods are

incorporated. These technologies improve the comprehension of delicate emotions or sentiments by imitating human interpretation, which helps identify cyber-hate. Classifier performance is enhanced by the use of bio-inspired optimization methods such as Particle Swarm Optimization and Genetic Algorithms. By fine-tuning the classifiers, these optimization techniques improve accuracy and interpretability—two critical aspects of cyber-hate detection. Reducing unnecessary characteristics in the data is a priority. By removing extraneous data, this approach seeks to improve the efficiency and efficacy of the cyber-hate detection system by streamlining the categorization process. In the end, this elimination of superfluous characteristics improves the categorization process and produces more precise results when recognizing cases of cyber-hatred.

6. FUTURE SCOPE

By creating artificial examples of angry tweets, GANs, a kind of neural network architecture, might be used to correct imbalances in the dataset. By offering a more diversified and balanced dataset, this augmentation can improve the model's capacity to identify cyber-hatred. Sarcastic content analysis may be enhanced by using a framework for sarcasm detection that combines discriminator and generator networks, which are commonly employed in GANs. This configuration improves the comprehension and detection of sarcasm in online conversations by enabling the model to produce sarcastic material (generator) and distinguish it from non-sarcastic content (discriminator). Cyber-hate detection capacities can be improved by more study on the efficacy of Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs). These architectures are particularly good at learning

sequential dependencies (RNNs) and hierarchical features (CNNs), which may help them identify more intricate patterns in hate speech ([23], [24], [25], [26], [27]). To further increase classification accuracy and interpretability, fuzzy logic-based systems are combined with other optimization approaches beyond Genetic Algorithms and Particle Swarm Optimization. This extension expands the techniques used to improve models and get a deeper understanding of the subtleties of hate speech. To increase the model's generalizability in a variety of cyber-hate detection scenarios, the dataset utilized for training and testing must be expanded. A larger dataset enables the model to learn from a wider range of cyber-hate incidents, improving its capacity to correctly identify such content in a variety of contexts.

REFERENCES

- [1] J. Hani, M. Nashaat, M. Ahmed, Z. Emad, E. Amer, and A. Mohammed, "Social media cyberbullying detection using machine learning," *Int. J. Adv. Comput. Sci. Appl.*, vol. 10, no. 5, pp. 703–707, 2019.
- [2] B. Vidgen, E. Burden, and H. Margetts, "Social media cyberbullying detection using machine learning," Alan Turing Inst., London, U.K. Tech. Rep, Feb. 2022. [Online]. Available: https://www.ofcom.org.uk/_data/assets/pdf_file/0022/216490/alan-turing-institute-reportunderstanding-online-hate.pdf
- [3] 4.4.1 A Sampling of Cyberbullying Laws Around the World. Accessed: Nov. 1, 2023. [Online]. Available: <https://socialna-akademija.si/joining-forces/4-4-1-a-sampling-of-cyber-bullying-laws-around-the-world/>

- [4] The EU code of Conduct on Countering Illegal Hate Speech Online. Accessed: Nov. 1, 2022. [Online]. Available: https://commission.europa.eu/strategy-and-policy/policies/justice-and-fundamental-rights/combating-discrimination/racism-and-xenophobia/eu-code-conductcountering-illegal-hate-speech-online_en
- [5] K. Dinakar, R. Reichart, and H. Lieberman, “Modeling the detection of textual cyberbullying,” in Proc. Int. AAAI Conf. Web Social Media, vol. 5, no. 3, Barcelona, Spain, 2011, pp. 11–17.
- [6] A. Kontostathis, K. Reynolds, A. Garron, and L. Edwards, “Detecting cyberbullying: Query terms and techniques,” in Proc. 5th Annu. ACM Web Sci. Conf., May 2013, pp. 195–204.
- [7] D. Yin, Z. Xue, L. Hong, B. D. Davison, A. Kontostathis, and L. Edwards, “Detection of harassment on web 2.0,” in Proc. Content Anal. Web, Madrid, Spain, 2009, pp. 1–7.
- [8] M. Dadvar, F. D. Jong, R. Ordelman, and D. Trieschnigg, “Improved cyberbullying detection using gender information,” in Proc. 25th Dutch-Belgian Inf. Retr. Workshop, Ghent, Belgium, 2012, pp. 1–3.
- [9] M. Dadvar, R. Ordelman, F. De Jong, and D. Trieschnigg, “Towards user modelling in the combat against cyberbullying,” in Proc. 17th Int. Conf. Appl. Natural Lang. Process. Inf. Syst., 2012, pp. 277–283.
- [10] K. Reynolds, A. Kontostathis, and L. Edwards, “Using machine learning to detect cyberbullying,” in Proc. 10th Int. Conf. Mach. Learn. Appl. Workshops, Honolulu, HI, USA, Dec. 2011, pp. 241–244.
- [11] H. Hosseinmardi, S. A. Mattson, R. Rafiq, R. Han, Q. Lv, and S. Mishra, “Poster: Detection of cyberbullying in a mobile social network: Systems issues,” in Proc. 13th Annu. Int. Conf. Mobile Syst., Appl., Services, May 2015, p. 481.
- [12] D. Chatzakou, N. Kourtellis, J. Blackburn, E. De Cristofaro, G. Stringhini, and A. Vakali, “Mean birds: Detecting aggression and bullying on Twitter,” in Proc. ACM Web Sci. Conf., New York, NY, USA, Jun. 2017, pp. 13–22.
- [13] M. A. Al-Garadi, K. D. Varathan, and S. D. Ravana, “Cybercrime detection in online communications: The experimental case of cyberbullying detection in the Twitter network,” Comput. Hum. Behav., vol. 63, pp. 433–443, Oct. 2016.
- [14] V. S. Babar and R. Ade, “A review on imbalanced learning methods,” Int. J. Comput. Appl., vol. 975, no. 2, pp. 23–27, 2015.
- [15] N. Aggrawal, “Detection of offensive tweets: A comparative study,” Comput. Rev. J., vol. 1, no. 1, pp. 75–89, 2018.