International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 08 Issue: 04 | April - 2024 SJIF Rating: 8.448 ISSN: 2582-3930

Hate Speech Detection in Hindi language using Machine Learning and Pre-Trained Models

Apurva Pagadpalliwar[1], Rahul Rathod[2], Prabhit Chaugule[3], Rasika Ransing[4]

[1] Vidyalankar Institute Of Technology, Mumbai, India
[2] Springer Heidelberg, Tiergartenstr. 17, 69121 Heidelberg, Germany
lncs@springer.com

***Abstract*:**

Detecting hate speech is a crucial aspect of creating a safe and inclusive virtual space. As Hindi digital communication expands, it has become necessary to devise effective techniques for detecting hate speech in this language. This abstract presents a machine learning and natural language processing (NLP) approach for hate speech detection in Hindi texts. To this end, the research employs a dataset that consists of labeled Hindi text examples divided into two classes: hate speech and non-hate speech. This means that specific pre-processing techniques such as tokenizing, stemming and stop-word removal must be used to handle linguistic nuance in Hindi. For feature engineering- word embeddings, character level embeddings and TF-IDF vectors are extracted to represent the semantic meaning of the text. These features can then be used to train different machine learning algorithms including Support Vector Machines (SVM), Naïve Bayes and Random Forests. The models were trained using standard performance metrics like precision recall F1-score accuracy The proposed system for detecting hate-speech could be extended to social media platforms or online forums or content sharing websites with an aim of curbing hateful content spread to more tolerant online communities among people who speak Hindi.

Keywords: *Hate speech detection, Machine learning, Natural language processing (NLP), Pre-processing techniques, Feature engineering, Word embeddings, Character level embeddings, TF-IDF vectors, Support Vector Machines (SVM), Naïve Bayes, Random Forests.*

1. Introduction:

Social media platforms are now very common and have greatly increased the way we communicate and share information, giving people more chances to speak their minds than in the past. This online world has made it faster to share what we think with others and has helped us connect, but at the same time, it has led to a growth in negative talk on the internet.

According to Nockleyby [1], hate speech covers all types of communication like talking, writing or actions that attack people or use negative and unfair words against them because of their religion, country they come from, skin color, if they are male or female, or different parts of who they are. The increase in such hateful language on the internet is now a big worry. It's a problem not just for those people it hurts directly but also for everyone else because this kind of talk can lead to actual crimes based on hate [2].

To deal with the growing problem, some social media sites are taking steps to fight against hate speech. They do this even if it might limit how freely people can express themselves. These measures usually include stopping accounts and deleting content that is harmful or insulting.

The need to tackle hate speech on the internet has gained a lot of attention from experts studying how computers understand human language. They are searching for good and quick ways to find and reduce this problem in the online world [3], [4], [5], [6], [7]. Initial efforts to recognize hate speech used words and sentence structures as tools to separate hateful content from other kinds of communication. In contrast, Mehdad applied support vector machines and emotional indicators for identifying hate speech.

In the field of advanced studies, sentiment analysis has become very important for identifying hate speech. This is something Zhou and his colleagues have looked into. The authors in reference [10] presented a model for understanding sentiment and sharing knowledge. This model uses a list of

offensive words together with learning that handles multiple tasks at once for the purpose of recognizing hate speech. Although this approach has shown good results, it depends greatly on the belief that using insulting expressions and expressing negative feelings are consistent ways to tell apart hate speech from other types of speech.

As we explore the important conversation about hate speech on the internet, it is clear that fighting this online problem needs new and subtle approaches. The research continues to try to find a middle ground between protecting freedom of speech and maintaining values for a welcoming and diverse online community.

Related Work

Detecting hate speech in languages with few resources, such as Hindi and Marathi, has been difficult because there are not many linguistic tools or studies for these languages. To solve this problem, researchers have looked into different complex machine learning methods like CNN 1D, LSTM, and BiLSTM. They also use special word embeddings that are specific to the subject area.

In a research paper [11], they used advanced deep learning techniques, including CNN 1D, LSTM, and BiLSTM with word embeddings made for specific subjects to study a dataset that mixed Hindi and English. These techniques did better than the usual machine learning methods such as SVM and random forests.

Another study for comparison[12] was about identifying hate speech within tweets written in English. In this research, they tried different ways to create features and used several computer models that learn from data such as Logistic Regression, Decision Trees, Random Forests and Naive Bayes. They also applied TF-IDF and BOW methods for turning text into numerical data. We used both types of word vectors, GLoVe that comes already trained and our own custom ones, for training the LSTM and GRU models.

In the Hindi language area, [13] examined how machine learning and neural network methods can identify hate speech by sorting messages into categories like hateful, offensive or profane. They tested traditional machine learning techniques including Linear SVM, Adaboost, Random Forests and Voting Classifier against deep learning models that use LSTM technology. Interestingly, machine learning models exhibited superior performance in low-resource settings.

The research looked into different methods for categorizing Hindi text, including BOW, CNN, LSTM, BiLSTM, BERT and LASER models. It was found that the CNN model using fast text embeddings for Hindi language was very good at this task; the performance of LASER also came close to that of the best method.

A research paper about finding aggressive posts in Hindi language [15] looked at different ways like CNN, Multi-CNN, BiLSTM, CNN combined with BiLSTM, IndicBert, mBert and FastText word representations from both IndicNLP and Facebook. The findings showed that the models built on BERT did a bit better than the simple ones; among them all though was the multi-layered CNN model that used FastText vocabulary embeddings from IndicNLP as most effective basic model.

The task of finding hate speech has become more complex with the use of deep learning and big pre-trained language models. These models use word embeddings from training without supervision on large text collections, which improves their ability to spot hate speech. For example, they used FastText with stacked Bidirectional Gated Recurrent Units called BiGRUs[16] together, and also found that both FastText and BERT are good for understanding the meaning of words and information about parts of words[17].

To summarize, researchers are looking into many different models, word embeddings, and ways to design features in order to fight against hate speech in languages that don't have much resources. This shows how the area keeps changing as it tries to find better ways of identifying hate speech that take the context into account.

Proposed Work

1. Collecting Datasets

We collected datasets from GitHub which was from a particular repository that is called, “Hate-Speech-Detection-in-Hindi” (https://github.com/victorknox/Hate-Speech-Detection-in-Hindi/tree/main/Dataset).

2. For data preprocessing,

a) Remove English Words: Eliminated English words to focus on Hindi language.
b) Remove Symbols: I removed symbols so as to clean up data and reduce noise.
c) Remove Emojis: I took out emojis in order to make the texts more understandable.
d) Remove Stopwords: This eliminated common words to increase signal-to-noise ratio.
e) Stemming—This was applied in reducing word standardization and dimensionality.

3. Concerning label preprocessing,

a) Label Mapping: The number of label categories got reduced into two namely ‘hate’ and ‘non-hate’ suitable for deep learning models.
b) Label Encoding—The categorical labels were converted into binary format (‘1’ for hate, ‘0’ for non-hate).

4. Model Training:

a) We used different machine learning models like Logistic Regression, Random Forest and Decision Tree etc.
b) Random Forest—The highest accuracy achieved was 80.27%.
c) In addition, we explored deep learning models such as Indic-bert and MuRIL in order to improve accuracy levels within the model.
d) MuRIL—After fine-tuning with hyperparameters (2 epochs, batch size of 8), it demonstrated excellent accuracy of 89.18%.

5. Development of User Interface
a) An interface has been developed for testing the model.
b) Saving Model: Saved the trained MuRIL model.
c) Flask Framework – Used Flask while creating this user interface.

Machine Learning Model Implementation

1. Model Selection:
We have put into use different kinds of machine learning models such as Logistic Regression, Random Forest, Multinomial Naive Bayes, the Support Vector Classifier also known as SVC, LightGBM and Decision Tree Classifier.Focused on models showing promising accuracy for hate speech detection in Hindi.

2. Feature Extraction:

Used TF-IDF vector technique to change text data into numbers by looking at how often words appear and thinking about all the data together.Achieved efficient feature extraction using the TfidfVectorizer from the sklearn library.

3. Training Process:
Developed a reusable function named train to streamline the training process for each model.
Used the train_test_split function to split the dataset into parts for training and tests, giving 20% to testing.

4.Model Evaluation:
Evaluated model performance using metrics such as accuracy, precision, recall, F1-score, and support.Used the classification_report and accuracy_score functions from the sklearn library to create detailed evaluation metrics.

5.Model Hyperparameters:
Adjusted model hyperparameters to optimize performance.For Logistic Regression, tuned parameters such as C, max_iter, penalty, solver, class_weight, and warm_start.For Random Forest, fine-tuned parameters including n_estimators, max_depth, max_features, min_samples_split, min_samples_leaf, bootstrap, and criterion.

6.Model Accuracy:
Different models showed different levels of precision, and the Random Forest was the most precise one with an accuracy rate of 80.27%.
Recorded the accuracy of other models:
Logistic Regression: 79.6%
Multinomial Naive Bayes: 77.61%
Support Vector Classifier (SVC): 78.22%
LightGBM: 79.24%
Decision Tree Classifier: 67.15%

Implementation of Deep Learning Model

In trying to make a strong system for finding hate speech in Hindi, we used deep learning and worked with two advanced models called Indic-bert and MuRIL. We made a flexible function that helps us train our model very effectively.First, we started a new feature called "model_training" to make our deep learning process easier. We divided the data so that 20% is for testing and used a fixed random state of zero to keep it stable.Then we moved on to the area of tokenization, using a tokenizer to change our input data smoothly into a form that works well for deep learning. We thoughtfully set the token size at 128 so that everything in our dataset was consistent.We carefully organized our data into a dataset for TensorFlow, making sure it was set up well for the best results from the model.We kept going with our progress, adjusting the settings of our model very carefully. We planned well and decided to use 2 epochs for training, a batch size of 8 for training too, and a bigger batch size of 16 when we check how good it is doing. Also, we added important details like steps to get ready before real learning starts and slowing down the learning rate over time by applying weight decay.In the wide area of our strategy, we started creating the model to set up a base for our deep learning projects. With strong focus, we made progress in training the model, making use of the great possibilities from our selected designs.After the training ended, we saw our hard work pay off with deep understandings appearing. At first, Indic-bert showed very good results by reaching 70.59% in accuracy. But, our attention was really caught by the MuRIL model because it showed a very good accuracy of 89.18%.Understanding that MuRIL is superior, we made an important choice to keep our model being excellent. We took very careful steps to protect what we built so it remains powerful for future tasks.We started making a user interface using the Flask framework, so we could use our model's functions. We allowed users to put in Hindi text and then our model carefully checked it for hate speech before giving a clear result.On our path to fight against hate speech in

Hindi, using deep learning models has been a major change-making step, creating a path for an online community that is safer and includes everyone.

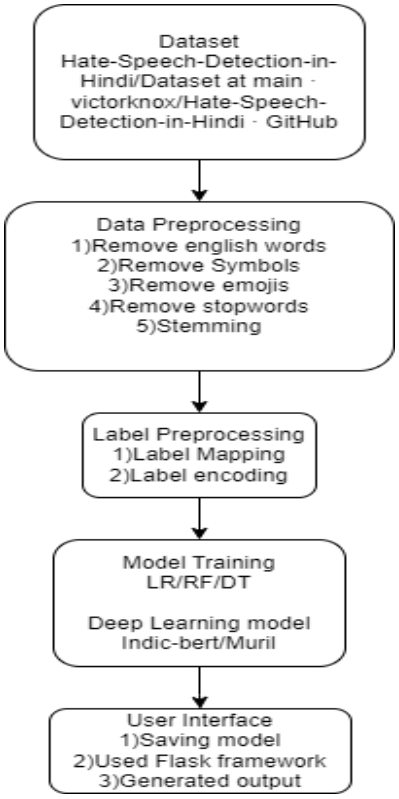

Fig1. Flowchart for workflow of hate speech detection in Hindi

Algorithm for Hate Speech Detection

In the project for finding hate speech in Hindi, we used different kinds of computer algorithms to correctly spot hate speech. We mainly relied on some usual machine learning ways like Logistic Regression and RandomForestClassifier which were very important in our work. Logistic Regression is simple and clear, giving a basic standard for sorting. It can show linear connections between characteristics and the target classifications, which helped us first understand patterns of hate speech in the data we have.

The RandomForestClassifier, which is a method that combines many decision trees for learning from data, showed the best performance among the models. It achieved a high accuracy rate of 80.27%. This classifier was effective because it used lots of decision trees to understand complicated patterns in the data and this helped it to get better at identifying hate speech. Its skill in managing feature spaces with many dimensions and not straight-line relationships made it very fitting for the job we needed to do.

Support Vector Classifier helps to identify hate speech by separating it from non-hate speech using a special boundary in the data. It is good at finding the best line that allows for clear separation between hate and non-hate instances, which improves its ability to tell them apart.

These methods, together with Multinomial Naive Bayes, LightGBM, and decision tree classifier, created a full set of tools for finding hate speech in Hindi. By choosing the right features and models, these algorithms helped each other to identify hate speech well and gave important understanding of negative content in the data collection.

Deep Learning models like IndicBert and MuRIL helped achieve good results. IndicBert achieved an accuracy of 70.59%, while MuRIL attained an impressive accuracy of 89.18%.So,MuRIL become the winner among all,it gets the highest accuracy of 89.18% among all the algorithms .

Result & Discussion

Fig 1: Landing Page

The entry page is for users who want to find hate speech in Hindi. It has a friendly design and shows important details about what the platform does and its principles. When people visit, they see a big title "शब्द शोध: प्रेम और द्वेष की बोलियाँ" (Word Search: Languages of Love and Hate), this shows how the platform really focuses on knowing different cultures and making sure everyone respects each other. As you scroll down, there's more information about what the platform believes in – things like being respectful, staying together as one group, learning new things, not allowing any hate at all and always keeping a positive attitude. The page contains buttons too, for user's easier navigation towards the "About Us" and "Know More" parts, making it simpler to reach more details.

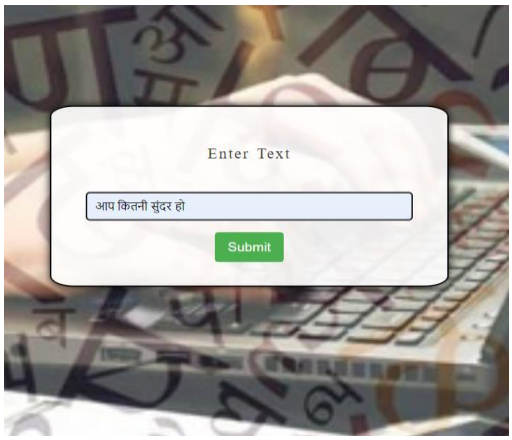

Fig 2: Entering text for detecting hate/non hate

The image shows the screen where people enter words to check for hate speech. They can write or copy and paste Hindi text, so the program decides if it's hate speech or not. The interface offers a smooth experience for users to add text for examination, helping the platform achieve its aim of finding and dealing with hate speech in Hindi.

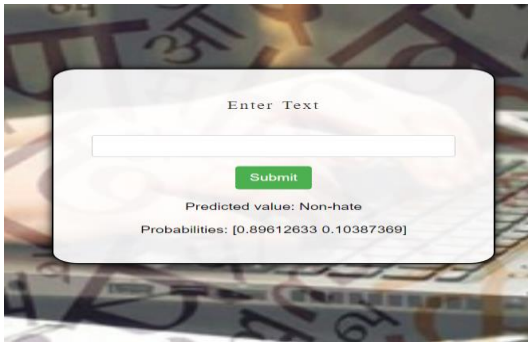

Fig 3: Result displayed is non-hate

In the picture, system shows result of sorting as "non-hate" for words put in. Sorting end point tells that checked writing does not have hate talk, giving people information about what kind of things they entered. This clear feedback system helps users to know how the platform evaluates their text and builds confidence in the process of identifying hate speech.

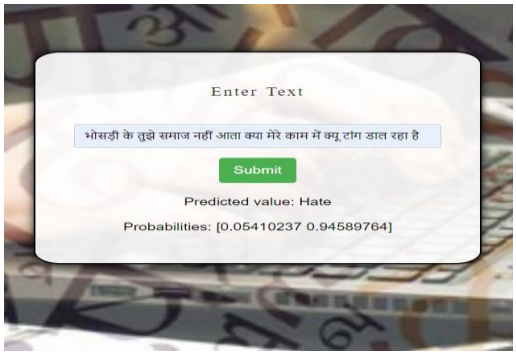

Fig 4: Result displayed is hate

Unlike the earlier picture, this one displays that the input text has been categorized as "hate." The system detects hate speech in the text it checks and shares this finding with users. The platform shows examples of hate speech, helping users to see and deal with bad content; this helps make conversations more respectful among people who speak Hindi.

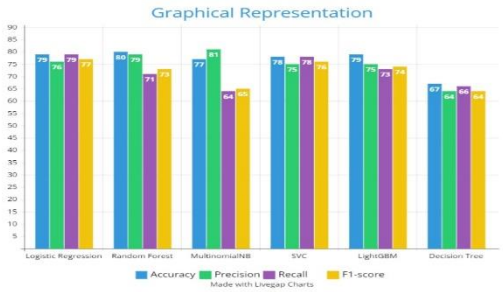

Fig 5: ML Model Performance Overview

This image gives a summary of how well machine learning models work when they are used to identify hate speech. It shows different methods like Logistic Regression, Random Forest Classifier, and Support Vector Classifier (SVC) that were applied in making the models learn. Within these, the Random Forest Classifier came out as the best model, with a high accuracy of 80.27%. This summary shows how well machine learning methods work for finding hate speech in Hindi language documents and points out that choosing algorithms and assessing models is very important when we are working on projects to detect hate speech.

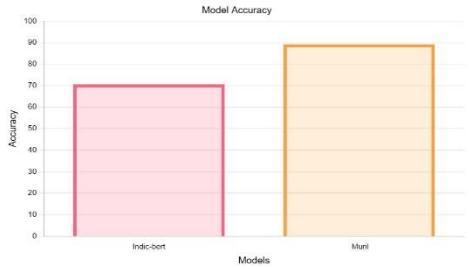

Fig 6:DL Model Performance Overview

This figure presents the accuracy of various deep learning models tested for hate speech detection in hindi. The X-axis represents the models used, namely IndicBert and MuRIL. The Y-axis represents the accuracy achieved by each model, expressed as a percentage.

Conclusion & Future Scope

We started our Hindi hate speech detection project by carefully preparing the data. We took great care in creating the dataset, removing English words, symbols, emojis and usual filler words, and we made sure that the different forms of words were consistent. Simplifying the labels to just 'hate' or 'non-hate' facilitated the training process.

At first, we looked at different machine learning models such as logistic regression, random forest, Multinomial Naive Bayes, Support Vector Classifier (SVC), LightGBM and decision tree classifier. The RandomForestClassifier was the best out of them all with a high accuracy rate of 80.27%.

We sought to increase accuracy and so we explored deep learning methods. Following extensive trials, the MuRIL model was victorious, achieving a remarkable 89.18% accuracy rate, even higher than the Indic-bert model.

Moving forward, we can improve our analysis and make it broader by using advanced deep learning techniques like 1D convolutional neural networks and bidirectional long short-term memory networks. These methods might help the model be better at finding complicated patterns in text data.

Expanding the types of categories to include things like fake, defamatory, peaceful, offensive and safe might give us a better understanding of different kinds of damaging materials that are found in this collection of Hindi texts.

To summarize, our present results are good at detecting hate speech in Hindi language, but there is a big chance to make them better. If we use more complex deep learning methods and add classification systems with many categories, we can improve the accuracy and scope of our research. This will help us fight against hate speech on the internet in a stronger way.

References

[1]J. Nockleyby, “Hate speech in encyclopedia of the american constitution,”
Electron. J. Academic Special Librarianship, vol. 3, pp. 1277–1279, 2000.
[2]M. Williams, “Hatred behind the screens: A report on the rise of online
hate speech,” *J. Exp. Theor. Artif. Intell.*, vol. 1, pp. 1–76, 2019. [Online].
Available: https://hatelab.net/wp-content/uploads/2019/11/Hatred-
Behind-the-Screens.pdf
[3] Y. Ding, X. Zhou, and X. Zhang, “YNU_DYX at semeval-2019 task 5: A
stacked BiGRU model based on capsule network in detection of hate,” in
Proc. 13th Int. Workshop Semantic Eval., 2019, pp. 535–539.
[4] G. Mou, P. Ye, and K. Lee, “SWE2: Subword enriched and significant
word emphasized framework for hate speech detection,” in *Proc. 29th ACM Int. Conf. Inf. Knowl. Manage.*, 2020, pp. 1145–1154,
doi: 10.1145/3340531.3411990.
[5] R. Cao and R. K. Lee, “Hategan: Adversarial generative-based data augmentation
for hate speech detection,” in *Proc. 28th Int. Conf. Comput. Linguistics*,
2020, pp. 6327–6338, doi: 10.18653/v1/2020.coling-main.557.
[6] S. S. Tekiroglu, Y. Chung, and M. Guerini, “Generating counter narratives
against online hate speech: Data and strategies,” in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 1177–1190.
doi: 10.18653/v1/2020.acl-main.110.
[7] X. Zhou et al., “Hate speech detection based on sentiment knowledge
sharing,” in *Proc. 59th Annu. Meeting Assoc. Comput. Linguistics 11th
Int. Joint Conf. Natural Lang. Process.*, 2021, pp. 7158–7166.
[8] Y. Chen, Y. Zhou, S. Zhu, and H. Xu, “Detecting offensive language in
social media to protect adolescent online safety,” in *Proc. IEEE Int. Conf.
Privacy, Secur., Risk Trust, Int. Conf. Soc. Comput.*, 2012, pp. 71–80.

doi: 10.1109/SocialCom-PASSAT.2012.55.

[9] Y.Mehdad and J. R. Tetreault, "Do characters abuse more thanwords ?," in

Proc. 17th Annu. Meeting Special Int. Group Discourse Dialogue, 2016,

pp. 299–303, doi: 10.18653/v1/w16-3638. [10] Artiran, S., Ravisankar, R., Luo, S., Chukoskie, L., & Cosman, P. , "Measuring Social Modulation of Gaze in Autism Spectrum Condition With Virtual Reality Interviews" IEEE.

[10]X. Zhou et al., "Hate speech detection based on sentiment knowledge

sharing," in *Proc. 59th Annu. Meeting Assoc. Comput. Linguistics 11th*

Int. Joint Conf. Natural Lang. Process., 2021, pp. 7158–7166.

[11]S. Kamble, A. Joshi, Hate speech detection from code-mixed hindi-english tweets using

deep learning models, 2018. arXiv:1811.05145."

[12] V. Mujadia, P. Mishra, D. Sharma, Iiit-hyderabad at hasoc 2019: Hate speech detection, in:

FIRE, 2019."

[13] R. Joshi, R. Karnavat, K. Jirapure, R. Joshi, Evaluation of deep learning models for hostility

detection in hindi text, in: 2021 6th International Conference for Convergence in

Technology (I2CT), IEEE, 2021, pp. 1–5.

[14] R. Joshi, P. Goel, R. Joshi, Deep learning for hindi text classification: A comparison, in:

International Conference on Intelligent Human Computer Interaction, Springer, 2019, pp.

94–101.

[15] R. Joshi, R. Karnavat, K. Jirapure, R. Joshi, Evaluation of deep learning models for hostility

detection in hindi text, in: 2021 6th International Conference for Convergence in

Technology (I2CT), IEEE, 2021, pp. 1–5.

[16] Y. Ding, X. Zhou, and X. Zhang, "YNU_DYX at semeval-2019 task 5: A

stacked BiGRU model based on capsule network in detection of hate," in

Proc. 13th Int. Workshop Semantic Eval., 2019, pp. 535–539.

[17] G. Mou, P. Ye, and K. Lee, "SWE2: Subword enriched and significant

word emphasized framework for hate speech detection," in

Proc. 29th ACM Int. Conf. Inf. Knowl. Manage., 2020, pp. 1145–1154,

doi: 10.1145/3340531.3411990.