Systems and Soft Computing 6 (2024) 200112

Contents lists available at ScienceDirect

Systems and Soft Computing

journal homepage: www.journals.elsevier.com/soft-computing-letters

Telugu language hate speech detection using deep learning transformer models: Corpus generation and evaluation

Namit Khanduja [a,1,4,*], Nishant Kumar [a,2], Arun Chauhan [b,3]

[a] Department of Computer Science & Engineering, Faculty of Engineering & Technology, Gurukula Kangri Deemed to be University, Haridwar, Uttarakhand, India
[b] Department of Computer Science & Engineering, Graphic Era Deemed to be University, Dehradun, Uttarakhand, India

ARTICLE INFO

Keywords:
Offensive text
Hate speech
Deep learning
NLP
Transformers
Low-resource languages

ABSTRACT

In today's digital era, social media has become a new tool for communication and sharing information, with the availability of high-speed internet it tends to reach the masses much faster. Lack of regulations and ethics have made advancement in the proliferation of abusive language and hate speech has become a growing concern on social media platforms in the form of posts, replies, and comments towards individuals, groups, religions, and communities. However, the process of classification of hate speech manually on online platforms is cumbersome and impractical due to the excessive amount of data being generated. Therefore, it is crucial to automatically filter online content to identify and eliminate hate speech from social media. Widely spoken resource-rich languages like English have driven the research and achieved the desired result due to the accessibility of large corpora, annotated datasets, and tools. Resource-constrained languages are not able to achieve the benefits of advancement due to a lack of data corpus and annotated datasets. India has diverse languages that change with demographics and languages that have limited data availability and semantic differences. Telugu is one of the low-resource Dravidian languages spoken in the southern part of India.

In this paper, we present a monolingual Telugu corpus consisting of tweets posted on Twitter annotated with hate and non-hate labels and experiments to provide a comparison of state-of-the-art fine-tuned deep learning models (mBERT, DistilBERT, IndicBERT, NLLB, Muril, RNN+LSTM, XLM-RoBERTa, and Indic-Bart). Through transfer learning and hyperparameter tuning, the models are compared for their effectiveness in classifying hate speech in Telugu text. The fine-tuned mBERT model outperformed all other fine-tuned models achieving an accuracy of 98.2. The authors also propose a deployment model for social media accounts.

1. Introduction

Hate speech, is a form of verbal or written communication that targets individuals or groups based on their race, religion, ethnicity, gender, or other characteristics, and has appeared as a significant concern in the digital age [1]. The rise of social media platforms and online communities has provided a fertile ground for the spread of hate speech, posing serious threats and evidence of the harm it can cause to societal harmony, individual well-being, and freedom of expression [2–4]. The lack of controlled tolerance levels and regulations is still a topic of discussion in hate speech [5–8]. The multifaceted technological advancement boosts the availability of the models that aim to distinguish hate speech from non-hateful content, empowering content moderators, social media platforms, and law enforcement agencies to take prompt action against hate speech offenders [9].

English and other widely spoken languages have seen considerable progress in the hate speech classification due to the availability of large corpora making them resource-rich languages [10–13]. However, for languages like Telugu, which has limited available resources and research on hate speech, the task becomes even more challenging. Telugu, a widely spoken Dravidian language in the Indian states of Andhra Pradesh and Telangana, has received little attention in hate

* Corresponding author.
E-mail address: namit.khanduja@gmail.com (N. Khanduja).
[1] https://scholar.google.com/citations?hl=en&user=uCt3D90AAAAJ
[2] https://scholar.google.com/citations?hl=en&user=uCt3D90AAAAJ
[3] https://scholar.google.co.in/citations?user=cyRErR8AAAAJ&hl=en
[4] 0000-0003-0848-66260

https://doi.org/10.1016/j.sasc.2024.200112

Available online 19 June 2024

speech research and development.

Hate speech detection has its own set of challenges, with data scarcity and non-availability of models for downstream NLP tasks being of significance [14,15,16,17].

Different strategies have been utilized for hate speech identification, including traditional rule-based approaches, traditional machine learning classifiers [18–20], deep learning-based classifiers [21–23], and hybrid approaches [21,24]. Transformers [25] is a deep learning architecture based on a multi-head attention mechanism [26]. Transformers have no recurrent units and therefore require much less time than RNN [14] and LSTM [27].

While large language models (LLMs) [28] have shown remarkable progress in hate speech recognition for rich-resource languages like English, their application to low-resource languages poses more challenges. The multilingual state-of-the-art transformer models perform well for resource-constrained languages, it is seen that transformer models trained on monolingual datasets perform well provided sufficient data is available [29].

In this paper, we have attempted to answer the two important challenges: Data Scarcity and availability of the model. Our contribution to the data scarcity challenge is the creation of a monolingual labeled and balanced Telugu corpus of approximately 38,000 tweets annotated for hate and non-hate labels. The second challenge of the availability of models was handled by employing transfer learning and hyperparameter fine-tuning of seven state-of-the-art transformers and one deep learning method. All the models were evaluated, and results were compared and analyzed to find the best-suited one. The rest of the organization of the paper is as follows: Section 2 presents a literature review, Section 3 presents the methodology, Section 4 presents the creation of a dataset, Section 5 presents an experimental study, Section 6 presents Result analysis, a deployment model is proposed in Section 7and finally the work concludes in Section 8.

2. Related work

Hate speech detection has become a pressing concern in the era of social media and online communication platforms. The rise in hate speech incidents has led researchers to explore various techniques, including machine learning, deep learning, and transformer-based models, to combat this issue effectively. Most research work circles around widely spoken languages like English. However, Indian languages are less explored due to the non-availability of a large corpus, thereby hindering the progress of models for NLP tasks. Recently, transformer models have been released for a few Indian languages [30,29,31] like Hindi, Marathi, and Bengali. The work done also suggests that models trained on monolingual datasets perform better [29,27]. Many competitions of notable importance have been organized such as SemEval 2018, SemEval 2019 [32,33], HASOC 2020, and GermEval 2018 [34] to dive into finding the improved result for NLP tasks. In response, researchers have curated datasets from multiple sources for non-English-based languages and fueled the path to explore ways and contrast the feature sets and approaches for hate speech detection like machine learning methods incorporating supervised, unsupervised, and semi-supervised approaches [35,36] and different classification algorithms like Logistic Regression(LR), Support Vector Machines [37,38], deep learning approaches like Convolutional Neural Networks (CNN) [39], Recurrent Neural Networks (RNN) [14], and Long Short-Term Memory (LSTM) [36,40], and hybrid approaches harnessing the power of deep learning models and transformers [29,41,42,31,40,43]. The rapid evolution of neural networks to detect hate speech in multilingual data and analysis of the relationship between classification accuracy and other parameters like vocabulary size and quality have also been considered [44,45]. The literature review points to two major approaches used for automatic hate speech detection. First, the traditional machine learning approach and second deep learning approach.

2.1. Traditional machine learning approaches

The initial stages of hate speech detection heavily relied on traditional machine learning algorithms, which laid the foundation for subsequent research in the field. Multinomial Naïve Bayes classifier was widely adopted for classification tasks and proved to be effective in hate speech detection for Indian languages [46,47,48]. In 2017, Davidson et al. made a pivotal contribution by introducing a dataset and various features specifically designed for hate speech detection [49]. This seminal work marked the inception of systematic hate speech detection research and initiated a trajectory of advancements in the domain.

One key aspect of traditional machine learning approaches was feature engineering. Researchers leveraged features such as n-grams, sentiment analysis, and lexical characteristics to represent textual content effectively. N-grams captured the sequential nature of language, while sentiment analysis gauged the emotional tone within the text. Lexical characteristics, encompassing vocabulary and linguistic patterns, offered valuable insights into hate speech [50].

In conjunction with these features, traditional machine learning algorithms employed classifiers like Support Vector Machines (SVM) [37,38], K-nearest neighbour, and Random Forests [43,51,52]. These classifiers played a pivotal role in distinguishing hate speech from other forms of communication. SVM, known for its effectiveness in binary classification tasks, showed promise in hate speech detection by creating decision boundaries that separated hateful content from non-hateful content [37].

The approach to ensemble decision trees-based models like random forest classifier, gradient boosting, and XGboost, etc., proved to attain better accuracy than single Machine learning algorithms [30,35,53,54].

While traditional machine learning approaches achieved promising results, they encountered challenges in dealing with the complexity of language and context. Hate speech often manifests in subtle linguistic nuances and can heavily rely on context for interpretation. Traditional approaches struggled to capture these subtleties effectively, leading to limitations in their accuracy and generalization [38].

These machine learning algorithms convert the input text to feature vectors having values and a classification model is trained on these vectors to accomplish the desired task. A general framework for machine learning-based hate speech classification models used in the mentioned research papers is shown in Fig. 1.

In recent years, researchers have been exploring ways to bridge the gap between traditional machine learning and more advanced techniques, incorporating elements of deep learning and transfer learning to enhance hate speech detection capabilities. This transition reflects the evolving nature of the field as it adapts to the changing landscape of online communication and the increasing sophistication of hate speech tactics.

Traditional machine learning approaches played a pivotal role in the initial stages of hate speech detection research, providing valuable insights and paving the way for subsequent developments. While these approaches achieved promising results, their limitations in handling linguistic complexity and context prompted the emergence of more advanced techniques. The integration of deep learning and transfer learning signifies a shift toward more robust and context-aware hate speech detection systems.

2.2. Deep learning techniques for hate speech detection

Deep learning has emerged as a powerful tool for hate speech detection, owing to its ability to capture intricate patterns and context within text data. The adoption of deep learning methods has significantly improved the accuracy and robustness of hate speech detection systems.

In 2018, Zhang et al. presented a groundbreaking approach to hate speech detection by introducing a Convolutional Neural Network (CNN) model [39]. This model demonstrated superior performance compared

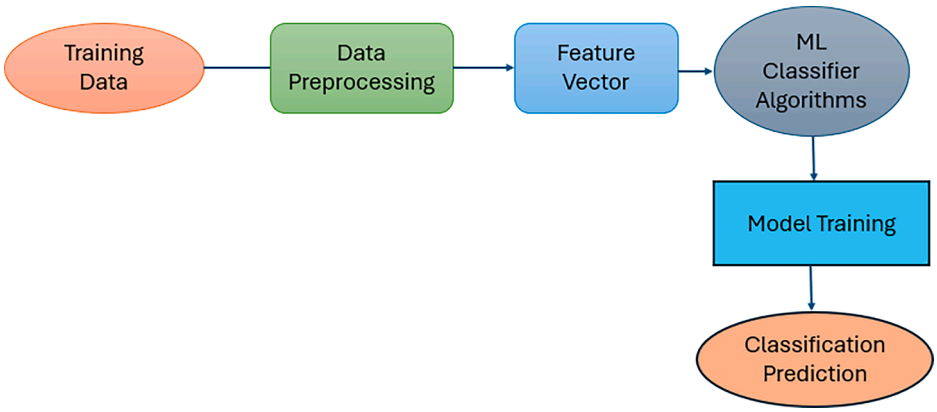

Fig. 1. A general model for ML algorithms.

to traditional methods. CNNs excel at capturing local patterns within the text, making them particularly effective at detecting hate speech, which often involves the use of specific phrases, keywords, and linguistic cues.

Simultaneously, Ribeiro et al. introduced a hierarchical attention-based model in 2018 [55]. This approach addressed the need to capture nuanced hate speech by focusing on hierarchical representations and attention mechanisms. Hierarchical attention models allowed for the exploration of both word-level and sentence-level information, enabling a more in-depth analysis of hate speech content.

The combination of the CNN-LSTM hybrid model for hate speech detection was proposed by Dutta et al. in 2021. Harnessing the power of the models the authors improved the detection accuracy to 88 percent [56].

Hostility detection using word embedding from fastText with CNN, BiLSTM, and GRU was done by Joshi et al. (2021) [54]. The use of domain-specific word embedding is suggested in combination with CNN, LSTM, and BiLSTM by Kamble and Joshi (2018) [57].

Since these seminal works, deep learning has continued to evolve in the context of hate speech detection. Researchers have explored various neural network architectures, including recurrent neural networks (RNNs), transformer-based models, and BERT, to further improve the accuracy of hate speech detection [14]. A general framework for deep learning models is shown in Fig. 2.

Another key development has been the application of transfer learning and fine-tuning. Pre-trained language models like BERT have been adapted for hate speech detection tasks [15]. These models leverage large-scale pre-training on vast text corpora to understand and identify hate speech more effectively. Fine-tuning domain-specific data further refines their hate speech detection capabilities.

In recent years, researchers have recognized that hate speech is not limited to text alone and have explored multimodal approaches that combine text, images, and videos. These approaches leverage deep learning techniques to analyze and understand the diverse content that constitutes hate speech in the digital age [16].

The integration of deep learning techniques and approaches has significantly enhanced the field of hate speech detection. From CNNs and hierarchical attention models to the latest advances in transfer learning and multimodal analysis, deep learning continues to drive progress in identifying and mitigating hate speech in online communication.

2.3. *Transformer-Based models for hate speech detection*

Transformer-based models have revolutionized the field of natural

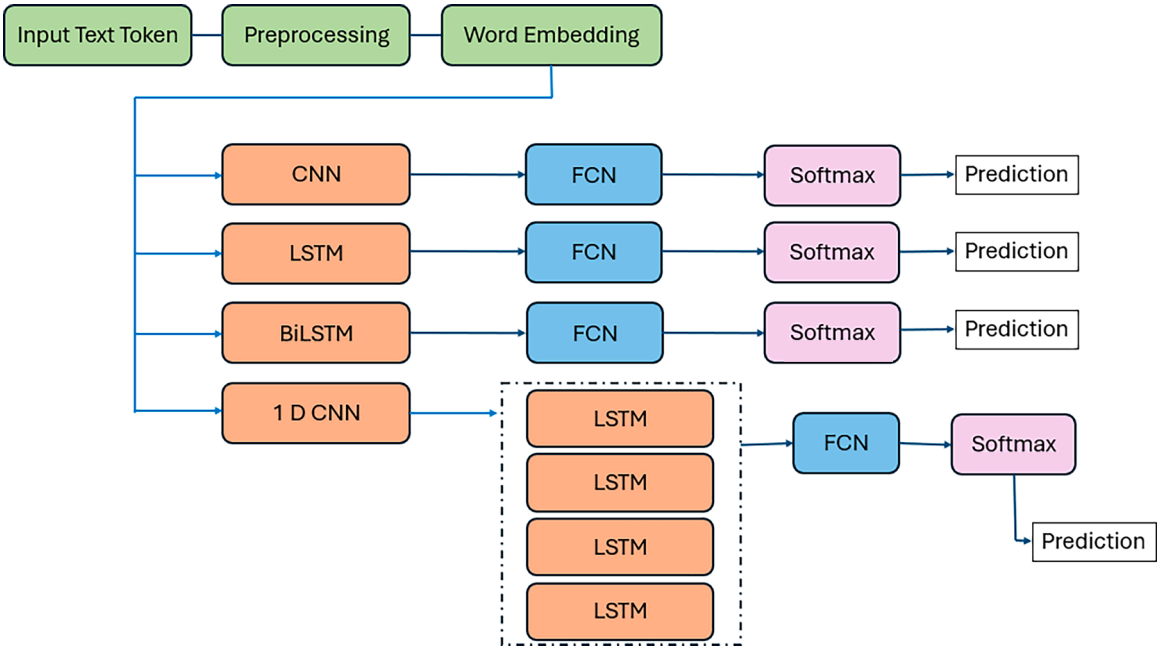

Fig. 2. A general framework for Deep Learning Algorithms.

language processing (NLP) and have proven to be highly effective in various NLP tasks. In the realm of hate speech detection, these models have brought about substantial improvements in accuracy and context awareness.

In 2019, Devlin et al. introduced BERT (Bidirectional Encoder Representations from Transformers), a breakthrough in NLP [58]. BERT's innovation lies in its ability to understand the context of words by considering the entire sentence. This contextual understanding is crucial in the nuanced and often subtle world of hate speech. By capturing the full context, BERT can identify hate speech more accurately and comprehensively.

Researchers quickly recognized the potential of BERT in the context of hate speech detection. By fine-tuning BERT on hate speech datasets, they harnessed its powerful language understanding capabilities to distinguish between hateful and non-hateful content effectively. This adaptation led to state-of-the-art results in hate speech detection [59].

Since BERT's introduction, the Transformer family has continued to expand. Models like RoBERTa [60], ALBERT [61], and DistilBERT [62] have emerged, offering variations and enhancements to the original BERT architecture. These models have been explored and adapted for hate speech detection tasks, demonstrating the adaptability of Transformer-based architectures in addressing the evolving landscape of hate speech [63]

Additionally, Transformer-based models have facilitated hate speech detection in various languages. Multilingual variants of BERT and other Transformers have been developed, enabling researchers to combat hate speech on a global scale [64]. These models consider linguistic nuances across different languages, making them invaluable for cross-cultural hate speech detection efforts.

As the field of hate speech detection continues to evolve, researchers are exploring ways to optimize Transformer-based models further. Techniques such as model distillation and model ensembling are being investigated to improve efficiency and robustness. The adaptability and contextual understanding of Transformers position them as pivotal tools in the ongoing fight against online hate speech (Fig. 3).

Transformer-based models, particularly BERT, have ushered in a new era of hate speech detection, enabling systems to comprehend the context and nuances of hateful language more effectively. Their versatility, multilingual capabilities, and adaptability to evolving challenges make them indispensable in the quest to maintain safe online environments.

2.4. Multimodal approaches for hate speech detection

Hate speech is not limited to text alone; it often incorporates images, videos, and other media formats. Addressing this multifaceted challenge requires innovative approaches that can analyze and understand the diverse content that constitutes hate speech. Multimodal approaches have emerged as a powerful solution to this problem.

In 2020, Lee et al. introduced a pioneering multimodal hate speech detection approach that combined textual and visual information [65]. This approach recognized that hate speech often relies on the synergy between text and accompanying media. By jointly analyzing textual content, images, and videos, researchers achieved significant improvements in detection accuracy.

Multimodal approaches employ text-image fusion techniques that allow the model to understand the relationships and context between the various modalities. These fusion techniques enable the system to recognize hate speech more effectively, even when it is implicit or relies on visual cues.

Deep learning plays a crucial role in multimodal hate speech detection. Models like VGG, ResNet, and their variants are employed to extract features from images and videos, while transformer-based models, such as BERT, are used for textual analysis. The extracted features are used to provide a holistic understanding of the content [66].

While multimodal approaches have shown promise, they also come

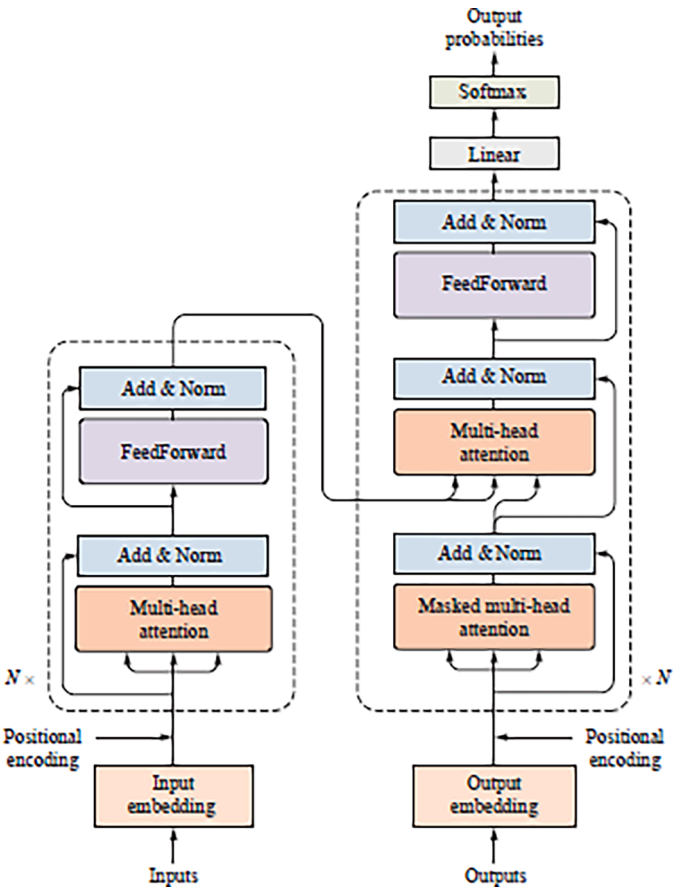

Fig. 3. A General Transformer model [26]

with unique challenges, such as data acquisition and model complexity. Researchers are actively exploring strategies to overcome these challenges. Recent advancements in transfer learning, pre-trained models, and multimodal datasets have further propelled the field [67].

Multimodal hate speech detection has practical applications in content moderation on social media platforms and online communities. It enables more comprehensive and effective monitoring and removal of harmful content, contributing to safer online environments.

As the field continues to evolve, future research may focus on refining multimodal models, expanding to additional languages, and addressing ethical considerations related to hate speech detection. Multimodal approaches are poised to play a pivotal role in the ongoing battle against hate speech on digital platforms.

Multimodal Approaches represent a significant advancement in hate speech detection, as they enable systems to consider not only textual content but also visual and auditory elements. By comprehensively analyzing the multiple facets of hate speech, these approaches contribute to a safer and more inclusive online environment.

Hate speech detection using machine learning, deep learning, and transformer-based models has seen considerable progress in recent years. Traditional machine learning approaches laid the foundation, but deep learning and transformer-based models, particularly BERT, have emerged as state-of-the-art solutions. Multimodal approaches and the use of pre-trained models have further enhanced the accuracy and robustness of hate speech detection systems. As the digital landscape evolves, continued research in this field is crucial to effectively combat hate speech and maintain safe online environments.

3. Methodology

3.1. Data collection

To investigate hate speech in the Telugu language, we collected a comprehensive dataset of tweets. The data collection process involved the following steps:

- **Data Source**: The primary source of data collection is Twitter.
- **Keywords and Hashtags**: We identified relevant keywords and hashtags commonly associated with hate speech in the Telugu language to filter the tweets.
- **Time Frame**: The data was collected from June 2022 to June 2023.
- **Volume**: A total of approximately fifty thousand tweets were collected to ensure a substantial and representative sample.

3.2. Data preprocessing

The collected tweets underwent extensive preprocessing to prepare them for annotation and model training:

- **Language Filtering**: Non-Telugu tweets and tweets with significant code-switching were removed.
- **Duplicate Removal**: Duplicate tweets and retweets were identified and eliminated.
- **Noise Reduction**: Common noise elements such as URLs, emojis, and special characters were removed. Standard text normalization technique was applied to convert all text to lowercase and correct spelling errors.
- **Tokenization**: Tweets were tokenized into individual words or subwords as per the requirements of the Transformer models.

3.3. Data annotation

The preprocessed dataset was annotated by a panel of experts to label instances of hate speech:

- **Annotators**: Five experts, aged between 25 to 35, with proficiency in the Telugu language and experience in social media content analysis, were referred.
- **Annotation Guidelines**: Detailed guidelines were prepared to ensure consistency in identifying hate speech. These guidelines included definitions, examples, and borderline cases.
- **Annotation Process**: Each tweet was examined by the annotators independently, and labels were assigned as 'hate speech' or 'non-hate speech'.

3.4. Model training

Seven different Transformer models were trained on the annotated dataset to evaluate their effectiveness in detecting hate speech:

- **Model Selection**: The following Transformer models were selected for training: mBERT, DistilBERT, IndicBERT, NLLB, MuRil, XLM-RoBERTa, and Indic-Bart and fine-tuned for Telugu.
- **Training Procedure**: Each model was trained using a standard training-validation-test split. The training was conducted on a system having 32 GB RAM and Nvidia RTX 3040 6GB GPU to handle the computational requirements.
- **Hyperparameter Tuning**: A grid search was employed to fine-tune hyperparameters such as learning rate, batch size, and number of epochs.
- **Evaluation:** The performance of the models was evaluated using standard metrics: Precision, recall, F1-score, and accuracy were computed to assess the classification performance. Fig. 4 shows the diagrammatic view of Methodology.

4. Dataset collection & preprocessing

4.1. Data acquisition

In this evaluation study, we have embarked on a fascinating journey of harnessing the power of social media data to tackle the pertinent issue

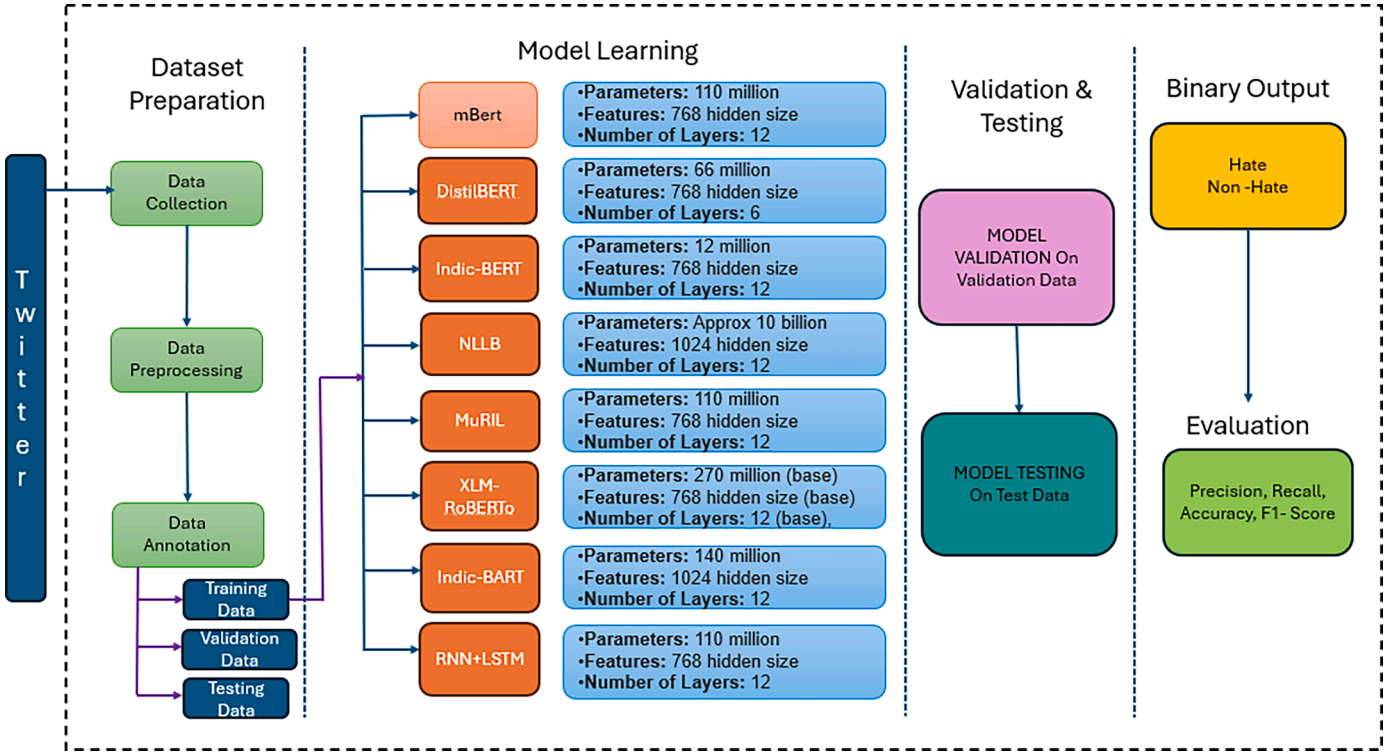

Fig. 4. Diagrammatic Representation of Methodology

of abusive language detection in the Telugu language. To accomplish this, we harnessed the vast expanse of Twitter as our primary data source, utilizing a powerful data collection social networking service in Python known as Snscrape. Our first step involved carefully curating our dataset by identifying specific keywords that are relevant to abusive language in Telugu. By inputting these keywords into Snscrape, we were able to retrieve an impressive corpus of approximately 50,000 tweets, all brimming with the potential to shed light on the prevalence of abusive language within this linguistic domain. However, the journey had just begun. Recognizing the need for pristine data, we dedicated considerable effort to cleaning and preparing our dataset for further analysis. This involved implementing robust data cleaning techniques to remove duplicate entries, manage missing values, correct formatting errors, and eliminate any irrelevant information that may have cluttered our dataset. This meticulous process ensured that our subsequent analyses were based on accurate and reliable data.

Next came the crucial step of annotation, a task that required human touch. Each tweet in our dataset was carefully evaluated and manually labeled by a team of five annotators having skills in reading and writing in the Telugu language, out of these five annotators, three were male and two females, all being in the age group of 25 to 35. The annotators discussed and annotated the tweets by unitedly agreeing to the decision to classify them as either positive or negative in terms of their language usage. This annotation process was conducted meticulously to ensure consistent and accurate labeling, empowering us with a valuable dataset that would serve as the foundation for training our abusive language detection model. This study holds immense significance, as abusive language detection plays a pivotal role in creating a safer and more inclusive online environment. By focusing specifically on the Telugu language, we aim to address the unique linguistic nuances and challenges associated with abusive language in this context. As we progress further, our annotated dataset will serve as a valuable resource for training and refining sophisticated machine-learning models capable of automatically identifying and flagging abusive language in Telugu. By leveraging the power of data and innovative technologies, we strive to contribute to the creation of effective solutions that foster respectful and responsible communication online.

4.2. Data cleaning & preprocessing

During the data cleaning process, we employed regular expressions to effectively clean each sentence. This involved the removal of mentions, user IDs, and emojis from the text. Additionally, we eliminated English words that were deemed irrelevant, as well as any non-relevant text that might have been present. By utilizing regular expressions, we ensured that the data was refined and prepared for further analysis or processing. The Dataset contains around 50,000 tweets, but in the preprocessing phase, we have found that the dataset is biased towards negative labels. We then removed around 10,000 negative tweets randomly, which made the whole dataset balanced. Table 1 showcasts the balanced dataset.

Total of 135 stop words relevant to Telugu language were used. Some of Stop words used are: అందుకే', 'అందులో', 'అందులోని', 'అందులోను', 'అని', 'అను', 'అవ్వడు', 'అయితే', 'అలా', 'ఆ', 'ఆంగ్లం', 'ఆంధ్ర', 'ఆంధ్రవ్రదేశ్', 'ఆగష్టు', 'ఆగస్టు', 'ఆది', 'ఆధారంగా', 'ఆన్ని', 'ఆరంభం', 'ఆరు', 'ఆలోచనలకి', 'ఇంకా', 'ఇంగ్లీష్', 'ఇది', 'ఇదే', 'ఇన్ని', 'ఇప్పటికే', 'ఇవ్వడం', 'ఇవ్వాలి', 'ఇవ్వాల్సిన', 'ఈ', 'ఈగ', 'ఈరోజు'

Table 1
Balancing the dataset

Before Preprocessing		After Preprocessing	
Label	**No. of. Tweets**	**Label**	**No. of. Tweets**
Negative	28,001	**Negative**	18,521
Positive	19,957	**Positive**	19,514
Total	47,958	**Total**	38,035

We have divided the data into three splits: Train dataset, Validation Dataset, and Test Dataset. The proportions for the dataset are given in Table 2.

Samples of Dataset:

Dataset Sample

Content	Label
ఏరా ఎర్రిపుక ఒట్ నీటార్ మోడ్డ గూడు	1
ఒరీ మబ్బు పుక బిబి అభిమానులు నా మోడ్లో ఓటు వేస్తే ఏంటి వేయకపోతే నాకేంటి జీవితం బాగుండాలి అంటే ఓటు వేస్తారు మోడ్డ గుడిసివోవాలి అంటే జగన్ గదికి వేసుకుంటారు	1
మా కోడల ఏంట్రూడింగితే పుచ్చకాయ పగిలివొద్ది	1
పుకు నాకు రా కొజ్జా	1
ట్విట్టర్ నోపేన్ బాగుండాలి అంటే వైఫై ఉండాలి నాధారణ మొబైల్ డేటా అయితే నెమ్మదిగా ఉంటుంది	0
నాకు ఎవ్పుడో అద్రుష్టం	0
అలా అయ్యేనా హాయావీ నీ అయితే మ్యాచ్ బాగుండాలి	0
అదేంటి ఫ్యాన్స్ ఎమో షూట్ అవ్వక వాయిదా పడింది అన్నారు బాబు మంచితనం ఆహా	0

5. Model learning & experimental setup

5.1. Model description

In our hate speech recognition, we employed multiple models, including RNNs, LSTMs, and state-of-the-art Transformers. By using these different model architectures, we conducted extensive experimentation to fully leverage the potential of our collected dataset. The objective was to develop a robust and accurate system for detecting and classifying hate speech. By exploring various models, we aimed to identify the most effective architecture that would enable us to achieve high performance in recognizing and mitigating hate speech instances.

5.2. Experimental setup & learning approach

5.2.1. Approach using RNN & LSTM

In this step, we initially utilized the RNN + LSTM model, which has gained significant recognition for its robust performance in Natural Language Processing (NLP) tasks. The combination of RNN and LSTM layers enhances the model's ability to capture sequential patterns and long-term dependencies within text data. The model is a sequential model, composed of various layers. It begins with an embedding layer that maps the input sequences to a lower-dimensional space (in this case, sixteen dimensions). This layer helps represent the words in a continuous vector space, capturing their semantic relationships.

Next, a dropout layer is applied to prevent overfitting by randomly deactivating some neurons during training, promoting better generalization of the model. Following the dropout layer, an LSTM layer is added. LSTM (Long Short-Term Memory) is a type of recurrent neural network that excels at capturing long-term dependencies in sequential data, making it particularly suitable for NLP tasks. To facilitate further processing, a flattened layer is employed, reshaping the LSTM layer's output into a one-dimensional vector. This enables the subsequent dense layers to receive the sequential information in a more manageable format. The model then proceeds with a dense layer, comprising 512 neurons, which applies a linear transformation to the input. This layer helps learn higher-level representations of the data. Another dropout

Table 2
Proportions of the Dataset for Training, Validation and Testing.

Dataset	No. of. Tweets	Ratio
Train	30,428	4
Validation	7607	1
Test	3804	1

layer follows to mitigate overfitting, and finally, a dense layer with a single neuron is used for binary classification (1 for "positive" and 0 for "negative"). In total, the model has approximately 1.2 million parameters, which are learned during the training process. These parameters allow the model to adapt and make predictions based on the provided input data. Please refer to the given figure for specific metric values obtained during the model evaluation and training process.

5.2.2. Approach using multiple transformers

Transformer [68] models have revolutionized the field of Natural Language Processing (NLP) by introducing a groundbreaking architecture that has reshaped the way we approach language understanding. Unlike traditional recurrent neural networks, transformers leverage self-attention mechanisms to capture global dependencies and relationships within a sequence of words. This enables them to effectively model long-range dependencies, making them highly effective in tasks such as machine translation, sentiment analysis, question answering, and more. With their ability to learn contextual representations and capture fine-grained semantic relationships, transformer models have pushed the boundaries of NLP, leading to remarkable advancements in language understanding and generation, and playing a pivotal role in shaping the future of AI-powered language applications.

The author's approach to implementing multiple transformers for the evaluation work involved the utilization of various models, each having its unique characteristics and advantages. We began by implementing the mBERT (multilingual BERT) model [69], which has gained considerable recognition for its effectiveness in multilingual natural language processing tasks. mBERT leverages a transformer architecture, allowing it to capture contextual information and semantic relationships between words in different languages. We observed promising performance with mBERT, motivating us to explore further.

Next, we experimented with DistilBERT-multilingual [70], a distilled version of BERT that offers similar performance while being more computationally efficient. This model retains the essential characteristics of BERT but with a reduced number of parameters. The DistilBERT-multilingual model demonstrated favorable results, further encouraging us to diversify our transformer selection.

We also incorporated the XLM-Roberta model [71], which is a robust variant of BERT designed for cross-lingual understanding. XLM-Roberta is pre-trained on a vast corpus containing multiple languages and exhibits impressive performance across various language tasks. By leveraging XLM-Roberta, we aimed to leverage its multilingual capabilities to enhance our language understanding and classification tasks.

Additionally, we explored models specifically trained for Indic languages, such as IndicBERT [72], and MuRIL (Multilingual Representations for Indian Languages) [73]. These models are tailored to capture the unique linguistic characteristics and nuances of Indic languages, making them well-suited for the experiment. To implement all these models, we utilized the Hugging Face Transformers library, which provides pre-trained models and tokenizers for easy integration into our pipeline. Each model in this Indic language group had its tokenizer, ensuring optimal handling of language-specific tokens and vocabulary. For models like Indic-BERT, we utilized the Albert model and loaded the weights from the Indic-BERT model since Indic-BERT is based on the Albert architecture. This approach allowed us to leverage the benefits of the Indic-BERT training while utilizing the Albert model framework. For the MuRIL we have used a BERT tokenizer and model, as the [72] mentioned using BERT as the base model.

It's worth mentioning that not all models we experimented with had a sequence classification layer directly attached by Hugging Face. For models like NLLB (No Language Left Behind by Meta) [74] and IndicBART [75], which lacked a sequence classifier, we implemented a custom sequence classifier using PyTorch [76]. We employed a linear classification head to enable the models to perform sequence classification tasks effectively.

Throughout our experimentation, all the models exhibited powerful performance on our classification tasks. Detailed results and performance metrics for these models can be found in the Evaluation and Metrics Section of the paper. The diverse set of transformers allowed us to manage multilingual and Indic language data effectively, highlighting the versatility and effectiveness of these models in our study.

5.2.3. Hyperparameter settings and fine-tuning

The hypermeter settings and fine-tuning of the model are crucial for achieving optimal performance. In this paragraph, we will delve into the learning rate, training configuration, optimization strategies, and additional techniques employed.

Firstly, the learning rate is set at 2e-5, which is commonly used for fine-tuning pre-trained models. The number of training steps is fixed at 10, with a linear scheduler for warmup comprising 10 % of these steps.

For optimization, the AdamW optimizer is utilized with a weight decay parameter of 0.0001. Both training and evaluation batch sizes are set to 16, and the model is trained for 5 epochs. Gradient accumulation steps are configured to 2, which can help stabilize training and enable the use of larger batch sizes without running out of memory.

In addition to these strategies, custom training arguments are provided, specifying output directory, weight decay, number of epochs, batch sizes, gradient accumulation steps, and reporting configuration for Weight & Biases integration. This integration enables tracking and logging of various metrics such as train and validation losses, accuracy, precision, recall, and F1 score, providing valuable insights into model performance.

Furthermore, a custom training loop is incorporated for enhanced control over the training and evaluation process. This allows for detailed logging and evaluation at each epoch, facilitating better monitoring and adjustment of the training procedure. Overall, these hypermeter settings and fine-tuning techniques contribute to optimizing the model's performance for the specific task at hand.

6. Evaluation result and metrics

6.1. Evaluation metrics

Hate speech detection is a binary classification problem; the commonly used evaluation metrics of any model for binary classification problems are F1 score, recall, precision, accuracy ROC-AUC, and confusion matrix, to gain a comprehensive understanding of their performance.

F1 Score: The F1 score is a measure that combines both precision and recall into a single metric, providing an overall assessment of the model's effectiveness. It balances the trade-off between precision (the ability of the model to correctly identify positive instances) and recall (the ability of the model to correctly capture all positive instances). A higher F1 score indicates better overall performance in terms of both precision and recall. The harmonic mean of precision and recall. It provides a balance between precision and recall.

$$\mathbf{F1Score} = \mathbf{2} * (\mathbf{Precision} * \mathbf{Recall})/(\mathbf{Precision} + \mathbf{Recall})$$

Recall: Recall, also known as true positive rate or sensitivity, measures the proportion of actual positive instances correctly identified by the model. It quantifies the model's ability to capture all positive instances, minimizing false negatives. A high recall indicates that the model is effective in identifying positive instances, thereby reducing the number of missed positive cases. The proportion of correctly predicted hateful instances out of all actual hateful instances. It measures the model's ability to capture all hateful instances.

$$\mathbf{Sensitivity/TPR/Recall} = \mathbf{TP}/(\mathbf{TP} + \mathbf{FN})$$

Precision: Precision assesses the proportion of instances identified as positive that are truly positive. It quantifies the model's ability to avoid false positives. A high precision indicates that the model has a low rate of falsely labeling negative instances as positive. The proportion of

correctly predicted hateful instances out of all instances predicted as hateful. It measures the model's ability to avoid false positives.

$$\textbf{Precision} = \textbf{TP}/(\textbf{TP} + \textbf{FP})$$

Accuracy: Accuracy measures the overall correctness of the model's predictions by comparing the total number of correct predictions to the total number of instances. It provides a general evaluation of the model's performance across all classes. However, it may not be an appropriate metric if the dataset is imbalanced. The proportion of correctly classified instances (hateful or non-hateful) out of the total instances.

$$\textbf{Accuracy} = (\textbf{TP} + \textbf{TN})/(\textbf{TP} + \textbf{TN} + \textbf{FP} + \textbf{FN})$$

ROC-AUC: The Area under the Receiver Operating Characteristic curve measures the model's ability to distinguish between hateful and non-hateful instances across various threshold settings. The Area under the Receiver Operating Characteristic curve measures the model's ability to distinguish between hateful and non-hateful instances across various threshold settings.

Confusion Matrix: Provides a detailed breakdown of true positives, true negatives, false positives, and false negatives, allowing for deeper analysis of model performance. Provides a detailed breakdown of true positives, true negatives, false positives, and false negatives, allowing for a deeper analysis of model performance.

By utilizing these metrics, we gained a detailed understanding of each model's performance. The F1 score allowed us to assess the overall effectiveness of the models, while recall and precision provided insights into their performance in positive and negative instances. Accuracy provided a broader perspective on the model's overall correctness. These metrics played a crucial role in evaluating and comparing the models, enabling us to make informed decisions regarding their suitability for this study.

6.2. Test dataset performances of the models

Table 3 shows the comparision of the performance of all fine-tuned models. Figs. 5–9 showcasts the performance of all fine-tuned models on F1-score,Precision, Recall, and Accuracy.

6.3. Comparison and result analysis

The experimental models display varying degrees of performance in hate speech identification, as reported by their F1Score, Precision, Recall, and Accuracy metrics. mBERT, DistilBERT, and Indic-BERT lead the pack with consistently high scores across all metrics, highlighting their robustness and reliability in classifying hate speech. Amongst them DistilBERT shows, efficient performance with reduced computational resources while managing Telugu language text. NLLB and MuRIL also show commendable performance, albeit with slightly lower scores compared to the top performers. Indic-BERT is trained in 12 Indian Languages, including Telugu, displaying robust performance in hate speech identification. RNN + LSTM shows decent performance with acceptable scores across metrics, including Recall, in hate speech identification with Telugu language text. However, has lower Precision compared to transformer-based models and might indicate a higher likelihood of false positives. Fine-tuned XLM-RoBERTa, although has low Precision and high Recall indicates the model's ability to identify a considerable proportion of actual positive instances of hate speech in Telugu language text. Extremely low Precision may result in a higher number of false positives, leading to potential misclassification of non-hate speech as hate speech. Fine-tuned Indic-BART has moderate Recall suggests reasonable performance in identifying positive instances of hate speech in Telugu language text, due to Very low Precision and F1Score indicating a high number of false positives and lower overall performance compared to other models, potentially impacting effectiveness in hate speech identification.

Table 3
Performance of all fine-tuned models.

Model	F1 Score	Precision	Recall	Accuracy
mBERT	98.2	98.2	98.2	98.2
DistilBERT	98	98	98	98
Indic-BERT	98.1	98.1	98.1	98.1
NLLB	97.3	97.3	97.3	97.3
MuRIL	97.9	97.9	97.9	97.9
RNN + LSTM	91	92	91	91
XLM-RoBERTa	85.3	85.5	85.3	85.3
Indic-BART	33.9	25.7	50	51.3

7. Deployment model for social media

The authors propose a deployment model (HSCS-SMMT) to use the best-suited model learned from the evaluation study explained earlier. It starts with a platform that can post any text for the user. The user will write the content for the post, the HSCS-SMMT will call the API of the HS model to detect whether the content or text is Hate or Non- hate. Once the model detects positive HS, the system will generate a warning for the user and will prompt the open-gen AI model to generate suggestions to correct the sentences. If the user agrees with the suggestion, the API call to the social media platform will be done and the content will be posted. Diagramatic view of the deployment model is shown in Fig. 10.

8. Conclusion

With the proliferation of digital platforms and social media, hate speech can quickly spread and negatively affect individuals' safety and well-being, particularly in low-resource language communities. Effective identification of hate speech in low-resource languages like Telugu helps create safer online spaces and mitigates the harmful effects of online harassment and discrimination, thereby ensuring digital safety and well-being. It helps in preserving Cultural Integrity by promoting responsible communication practices and discouraging harmful language use that could erode cultural values and traditions. Hate speech often targets marginalized communities based on ethnicity, religion, gender, or other factors. Hate speech classification in Telugu empowers these communities by providing them with tools to combat discriminatory language and promote inclusivity and equality. It also provides valuable insights for policymakers and regulators to develop effective strategies for combating online hate speech and enforcing content moderation policies. By understanding the prevalence and nature of hate speech in Telugu, policymakers can tailor interventions to protect users and promote responsible online behavior.

Developing hate speech identification models for low-resource languages like Telugu challenges NLP researchers to innovate and adapt existing techniques to resource-constrained environments. This advancement contributes to the broader field of NLP by expanding the applicability of algorithms and models to diverse linguistic contexts. It helps in promoting linguistic diversity in AI research and development by addressing social issues in non-English languages, NLP researchers contribute to creating more inclusive AI technologies that serve a global user base.

In this study, we focused on hate speech classification in the Telugu language, we created a monolingual dataset and conducted a comprehensive evaluation of various transformer-based models fine-tuned. Our findings provide valuable insights into the performance of these models and their potential applications in addressing hate speech in Telugu text.

The results of experiments indicate that transformer models, when fine-tuned for hate speech detection in Telugu, can achieve impressive results. Among the models evaluated, fine-tuned mBERT (Tel-mBERT) consistently outperformed the others in terms of accuracy, precision, recall, and F1 score. The model demonstrated the ability to effectively identify hate speech in Telugu text, highlighting the robustness and versatility of transformer-based architectures.

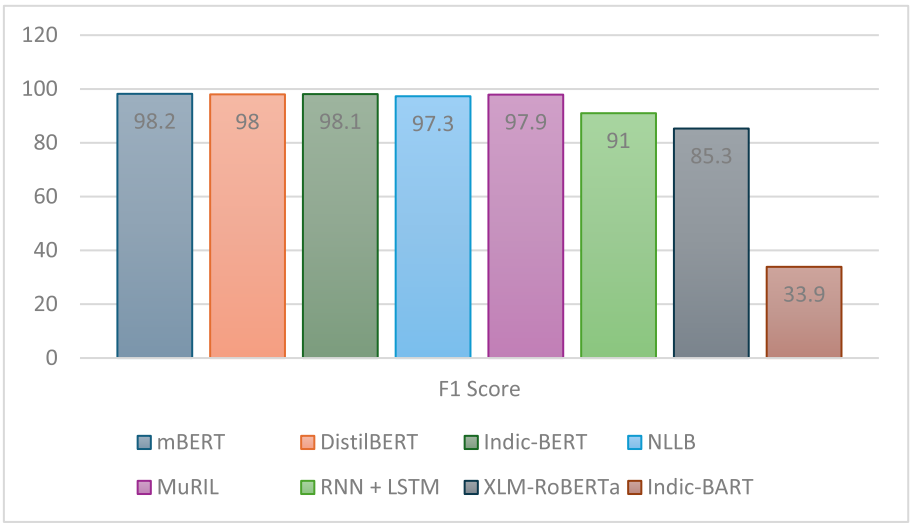

Fig. 5. F1 Score for all the fine-tuned models.

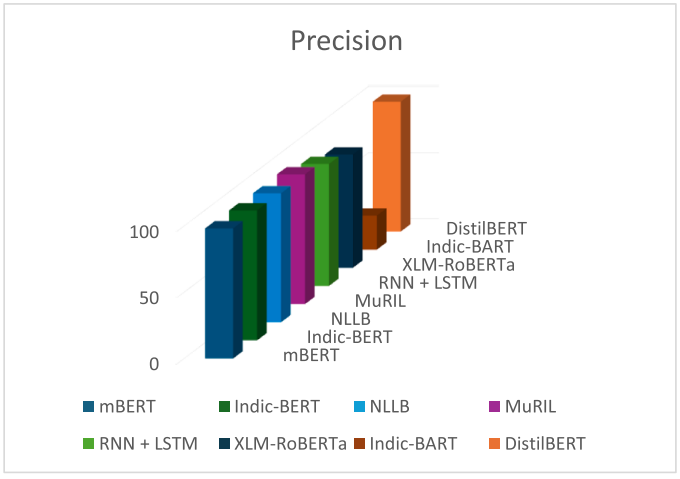

Fig. 6. Precision Performance of all the fine-tuned models

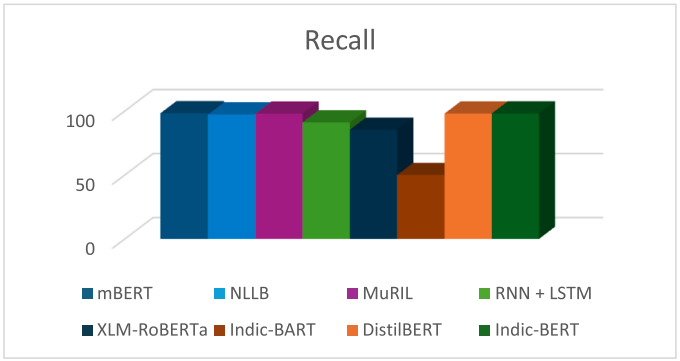

Fig. 7. Performance of Recall of all fine-tuned models

However, it is worth noting that no single model is a one-size-fits-all solution, and the choice of model may depend on specific use cases and requirements. Fine-tuned DistilBERT (Tel-DistillBERT) and Fine-tuned IndicBERT (Tel-IndicBERT) also performed well and may be suitable for scenarios where computational resources are in question.

Additionally, our experiments revealed that the custom Telugu dataset played a crucial role in the performance of these models. The quality and diversity of training data significantly impact a model's ability to generalize and detect hate speech accurately. Therefore, ongoing efforts to curate and expand hate speech datasets in Telugu and other languages are essential for improving hate speech detection systems.

Furthermore, traditional deep learning approaches, such as RNN+LSTM, did not yield significant performance improvements. This

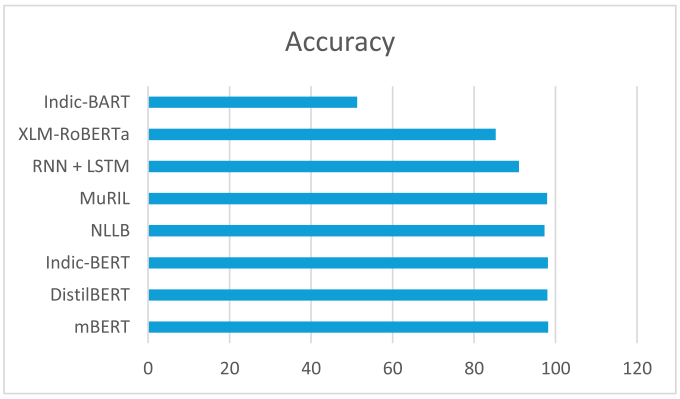

Fig. 8. Accuracy for all fine-tuned models

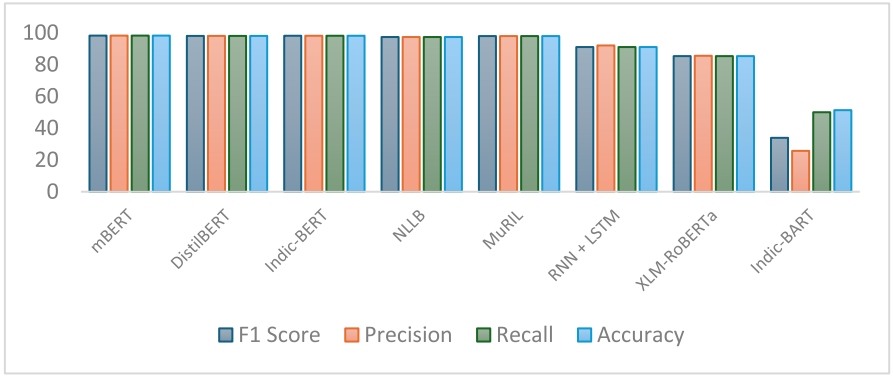

Fig. 9. Combined Performance Comparison of Test data with 8 Transformer models

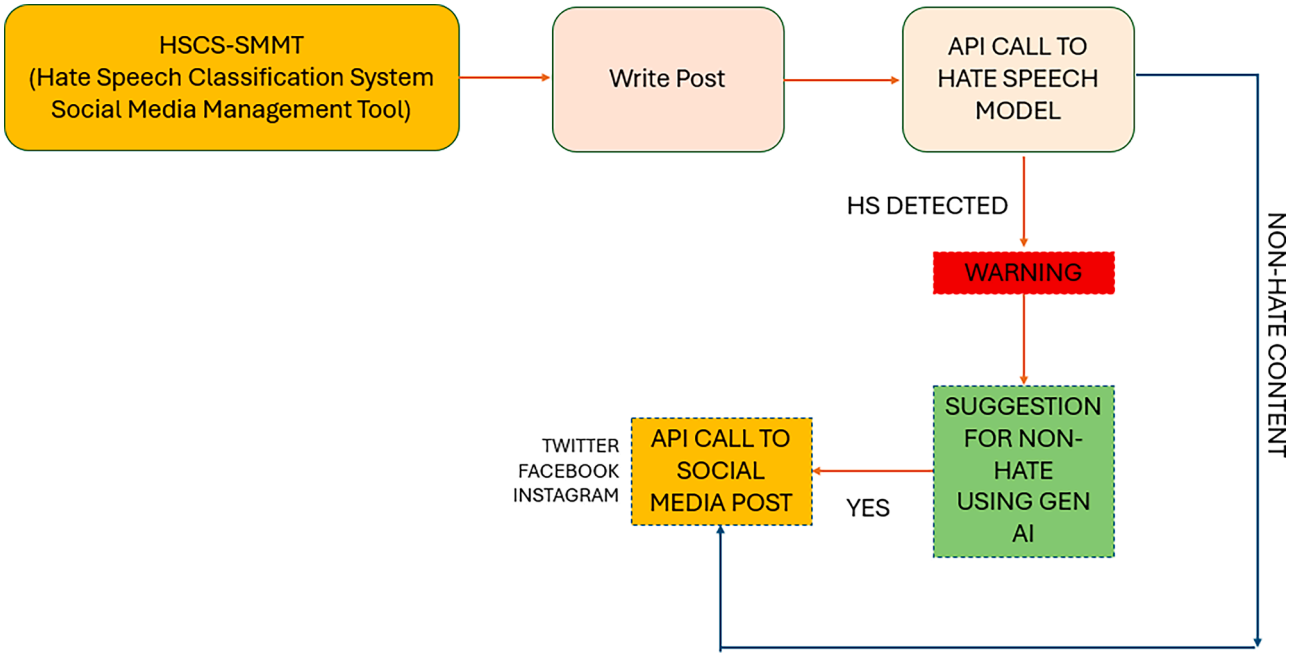

Fig. 10. Proposed Deployment Model

suggests that fine-tuned transformer models are capable of accurately capturing complex linguistic patterns and context required for hate speech detection in Telugu.

In conclusion, this study contributes to the field of hate speech detection by providing a comprehensive evaluation of transformer-based models on a custom monolingual Telugu dataset. While Tel-mBERT and Tel-DistillBERT emerged as the top-performing models, achieving an F1 score above 98. A deployment model is proposed to integrate the fined-tuned model in social media management systems. Future research should continue to explore and refine methods for hate

speech detection in various languages, considering the cultural and linguistic nuances that influence online discourse.

CRediT authorship contribution statement

Namit Khanduja: Writing – original draft, Validation, Software, Methodology, Formal analysis, Data curation. **Nishant Kumar:** Supervision. **Arun Chauhan:** Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

[1] A. Schmidt, M. Wiegand, A survey on hate speech detection using natural language processing, in: Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media, Association for Computational Linguistics, Valencia, Spain, 2017, pp. 1–10, pagesApril.

[2] K. Gelber, L. McNamara, Evidencing the harms of hate speech, Soc. Identiti. 22 (2016) 324–341.

[3] K. Saha, E. Chandrasekharan, M. De Choudhury, Prevalence and psychological effects of hateful speech in online college communities, in: Proceedings of the 10th ACM Conference on Web Science, WebSci '19, Association for Computing Machinery, New York, NY, USA, 2019, pp. 255–264, https://doi.org/10.1145/3292522.3326032.

[4] K. Müller, C. Schwarz, Fanning the flames of hate: social media and hate crime, SSRN Electron. J. (2017), https://doi.org/10.2139/ssrn.3082972.

[5] E. Barendt, What is the harm of hate speech? Ethic. Theory Moral Pract. 22 (2019) https://doi.org/10.1007/s10677-019-10002-0.

[6] Dworkin R. A new map of censorship. Index Censorship. 2006;35(1):130–3. https://doi.org/10.1080/03064220500532412.

[7] S. Heyman, Hate speech, public discourse, and the first amendment, in: I Hare, J Weinstein (Eds.), ***Extreme Speech and Democracy***, Oxford Scholarship, 2009, https://doi.org/10.1093/acprof:oso/9780199548781.003.0010. Online.

[8] Matsuda M.J. Public response to racist speech: considering the victim's story. In: R. D. M. J. Matsuda C. R. Lawrence III, K. Williams (eds.) ***Words That wound: Critical race theory, Assaultive speech, and the First Amendment***, pp. 17–52. Routledge, New York; 1993.

[9] D. Walsh, As content booms, how can platforms protect kids from hateful speech?, 2022. URL: https://mitsloan.mit.edu/ideas-made-to-matter/content-booms-how-can-platforms-protect-kids-hate-speech.

[10] H.H. Saeed, K. Shahzad, F. Kamiran, Overlapping toxic sentiment classification using deep neural architectures, in: 2018 IEEE International Conference on Data Mining Workshops (ICDMW), 2018, pp. 1361–1366, https://doi.org/10.1109/ICDMW.2018.00193.

[11] A. Vaidya, F. Mai, Y. Ning, Empirical analysis of multi-task learning for reducing identitybias in toxic comment detection, in: Proceedings of the International AAAI Conference on Web and Social Media 14, 2020, pp. 683–693, https://doi.org/10.1609/icwsm.v14i1.7334. URL, https://ojs.aaai.org/index.php/ICWSM/article/view/7334.

[12] S.M. Carta, A. Corriga, R. Mulas, D.R. Recupero, R. Saia, A supervised multi-class multi-label word embeddings approach for toxic comment classification, in: International Conference on Knowledge Discovery and Information Retrieval, 2019. URL, https://api.semanticscholar.org/CorpusID:204754719.

[13] T. Tran, Y. Hu, C. Hu, K. Yen, F. Tan, K. Lee, S. Park, Habertor: an efficient and effective deep hate speech detector, 2020. arXiv:2010.08865.

[14] P. Fortuna, S. Nunes, A review of deep learning techniques for hate speech detection, in: European Conference on Information Retrieval, 2021, pp. 201–213.

[15] M.S. Akhtar, A. Ekbal, P. Bhattacharyya, Survey on hate speech detection: challenges and opportunities, ACM Comput. Surv. (CSUR) 53 (4) (2020) 1–38.

[16] T. Silva, P. Carvalho, R.L. Santos, arXiv preprint, 2021.

[17] G. Kovács, P. Alonso, R Saini, Challenges of hate speech detection in social media, SN Comput. Sci. 2 (2021) 95, https://doi.org/10.1007/s42979-021-00457-3.

[18] Z. Waseem, D. Hovy, Hateful symbols or hateful people? predictive features for hate speech detection on Twitter, in: Proceeding of NAACL student research Workshop, 2016, pp. 88–93, pages.

[19] F.E. Ayo, O. Folorunso, F.T. Ibharalu, I.A. Osinuga, A. Abayomi-Alli, A probabilistic clustering model for hate speech classification in Twitter. Expert systems with applications, in: Proceedings of the NAACL student research workshop 173, 2021 pages 88–93, 201.

[20] S. MacAvaney, H.-R. Yao, E. Yang, K. Russell, N. Goharian, O. Frieder, Hate speech detection: challenges and solutions, PLoS ONE 14 (8) (2019) e0221152.

[21] P. Pinkesh Badjatiya, S. Gupta, M. Gupta, V. Varma, Deep learning for hate speech detection in tweets, in: Proceedings of the 26th international conference on World Wide Web companion, 2017, pp. 759–760, pages.

[22] D. Bahdanau, K. Cho, and Y. Bengio.Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473, 2014.

[23] C. Bharathi Raja, Multilingual hope speech detection in English and Dravidian languages, Int. J. Data Sci. Analyt. 14 (4) (2022) 389–406.

[24] Z. Mossie, J.-H. Wang, Vulnerable community identification using hate speech detection on social media, Inf. Process. Manag. 57 (3) (2020), 102087.

[25] Bahdanau; Cho, K.; Bengio, Y. (September 1, 2014). "Neural Machine Translation by Jointly Learning to Align and Translate". arXiv:1409.0473 [cs.CL].

[26] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Adv. Neural Inf. Process. Syst. (2017) 30.

[27] S. Hochreiter, J. Schmidhuber, Long Short-Term Memory, Neur. Comput. 9 (8) (1 November 1997) 1735–1780, https://doi.org/10.1162/neco.1997.9.8.1735.

[28] W.X. Zhao et al., "A Survey of Large Language Models," March 2023.arXiv:2303.18223 [cs.CL]. 10.48550/arXiv.2303.18223.

[29] R. Joshi, L3Cube-MahaCorpus and MahaBERT: marathi monolingual corpus, Marathi BERT language models, and resources, in: Proceedings of the WILDRE-6 Workshop within the 13th Language Resources and Evaluation Conference, Jun 2022, 6 R. Joshi 97–101European Language Resources Association, Marseille, France, https://aclanthology.org/2022.wildre-1.17.

[30] A. Velankar, H. Patil, A. Gore, S. Salunke, and R. Joshi, "Hate and Offensive Speech Detection in Hindi and Marathi," Oct. 2021.

[31] A. Bhattacharjee, T. Hasan, W.A. Uddin, K. Mubasshir, M.S. Islam, A. Iqbal, M. S. Rahman, R. Shahriyar, *Banglabert: Language model Pretraining and Benchmarks For Low-Resource Language Understanding Evaluation in Bangla. Findings of the North American Chapter* of the Association for Computational Linguistics, NAACL, 2022.

[32] M. Zampieri, S. Malmasi, P. Nakov, S. Rosenthal, N. Farra, R. Kumar, arXiv preprint, 2019.

[33] M. Zampieri, P. Nakov, S. Rosenthal, P. Atanasova, G. Karadzhov, H. Mubarak, L. Derczynski, Z. Pitenis, Ç. Çöltekin, Semeval-2020 task 12: multilingual offensive language identification in social media, Semeval 2020 (2020) arXiv preprint, arXiv:2006.07235.

[34] M. Wiegand, M. Siegel, and J. Ruppenhofer. 2018. Overview of the GermEval 2018 shared task on the identification of offensive language.

[35] M.S. Tash, Z. Ahani, A. Tonja, M. Gemeda, N. Hus sain, O. Kolesnikova, Word level language identification in code-mixed kannada-english texts using traditional machine learning algorithms, in: Proceedings of the 19th International Conference on Natural Language Processing (ICON): Shared Task on Word Level Language Identification in Code mixed Kannada-English Texts, 2022, pp. 25–28, pages.

[36] M. Tash, J. Armenta-Segura, Z. Ahani, O. Kolesnikova, G. Sidorov, A. Gelbukh, Lidoma@ dravidianlangtech: convolutional neural networks for studying correlation between lexical features and sentiment polarity in tamil and tulu languages, in: Proceedings of the Third Workshop on Speech and Language Technologies for Dravidian Languages, 2023, pp. 180–185, pages.

[37] J. Bjerva, S. Ruder, I. Augenstein, arXiv preprint, 2021.

[38] S. Mukherjee, L. Bing, arXiv preprint, 2016.

[39] Y. Zhang, D. Robinson, J. Tepper, arXiv preprint, 2018.

[40] A.L. Tonja, M.G. Yigezu, O. Kolesnikova, M.S. Tash, G. Sidorov, A. Gelbuk, arXiv preprint, 2022.

[41] M. Mozafari, R. Farahbakhsh, N. Crespi, A bert-based transfer learning approach for hate speech detection in online social media, in: International Conference on Complex Networks and Their Applications, Springer, 2019, pp. 928–940, pages.

[42] Katharopoulos, A.; Vyas, A.; Pappas, N.; Fleuret, F. (2020). "Transformers are RNNs: fast autoregressive Transformers with linear attention". ICML 2020. PMLR. pp. 5156–5165.

[43] B. Bharathi, J. Varsha, Ssncse nlp@ tamilnlp-acl2022: transformer based approach for detection of abusive comment for Tamil language, in: Proceedings of the 2nd workshop on speech and language technologies for Dravidian languages, 2022, pp. 158–164.

[44] L. Dhanya, K. Balakrishnan, Hate speech detection in Asian languages: a Survey, in: 2021 International conference on communication, control, and information sciences (ICCISc) 1, 2021, pp. 1–5 (IEEE).

[45] S. Dowlagar, R. Mamidi, A survey of recent neural network models on code-mixed Indian hate speech data, Forum Inform. Retriev. Evaluat. (2021) 67–74.

[46] S. Akhter, et al., Social media bullying detection using machine learning on Bangla text, in: 2018 10th International conference on electrical and computer engineering (ICECE). IEEE, 2018, pp. 385–388.

[47] H. Al Kuwatly, M. Wich, G. Groh, Identifying and measuring annotator bias based on annotators' demographic characteristics, in: Proceedings of the 4th Workshop on online abuse and harms, 2020, pp. 184–190.

[48] P. Rani, S. Suryawanshi, K. Goswami, B.R. Chakravarthi, T. Fransen, J.P. McCrae, A comparative study of different state-of-the art hate speech detection methods in Hindi–English code-mixed data, in: Proceedings of the 2nd workshop on trolling, aggression and cyberbullying, 2020, pp. 42–48.

[49] T. Davidson, D. Warmsley, M. Macy, I. Weber, arXiv preprint, 2017.

[50] M. Pavlou, Y. Zhou, L. Derczynski, A deep learning approach for hate speech detection, in: Proceedings of the 15th International Workshop on Semantic Evaluation, 2021 (SemEval-2021).

[51] S. Barnwal, R. Kumar, R. Pamula, IIT DHANBAD CODE CHAMPS at SemEval-2022 task 5: mAMI—Multimedia automatic misogyny identification, in: Proceedings of the 16th international workshop on semantic evaluation (SemEval-2022),

Association for Computational Linguistics, Seattle, 2022, pp. 733–735, https://doi.org/10.18653/v1/2022.semeval-1.101.
[52] A.M. Ishmam, S. Sharmin, Hateful speech detection in public Facebook pages for the Bengali language, in: 2019 18th IEEE international conference on machine learning and applications (ICMLA), IEEE, 2019, pp. 555–560.
[53] M. Sarker, M.F. Hossain, F.R. Liza, S.N. Sakib, A. Al Farooq, A machine learning approach to classify anti-social Bengali com ments on social media, in: 2022 International conference on advancement in electrical and electronic engineering (ICAEEE), IEEE, 2022, pp. 1–6.
[54] S. Kamble, A. Joshi, arXiv preprint, 2018.
[55] M.T. Ribeiro, S. Singh, C. Guestrin, Why should I trust you?" Explaining the predictions of any classifier, in: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, 2018, pp. 1135–1144.
[56] S. Dutta, U. Majumder, S.K. Naskar, sdutta at comma@ icon: a CNN-LSTM model for hate detection, in: Proceedings of the 18th international conference on natural language processing: shared task on multilingual gender biased and communal language identification, 2021, pp. 53–57.
[57] 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.
[58] J. Devlin, M.W. Chang, K. Lee, K. Toutanova, arXiv preprint, 2019.
[59] A. Mukherjee, L. Bing, A BERT-based transformer model for hate speech detection, in: Proceedings of the International Conference on Computational Linguistics (COLING), 2020.
[60] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, V. Stoyanov, arXiv preprint, 2019.
[61] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, R. Soricut, arXiv preprint, 2019.
[62] V. Sanh, L. Debut, J. Chaumond, T. Wolf, arXiv preprint, 2019.
[63] Z. Jiang, Z. Zhang, C. Gan, Hate speech detection with comment embeddings, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL), 2020.
[64] T. Pires, E. Schlinger, D. Garrette, arXiv preprint, 2019.
[65] J. Lee, J. Kim, C. Yoon, Multimodal hate speech detection on Twitter, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL), 2020.
[66] A. Zadeh, M. Chen, S. Poria, E. Cambria, L.P. Morency, arXiv preprint, 2021.
[67] X. Zhang, S. Wang, X. Zhang, H. Ji, arXiv preprint, 2021.
[68] T. Wolf et al., "HuggingFace's Transformers: state-of-the-art Natural Language Processing," Oct. 2019.
[69] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: pre-training of Deep Bidirectional Transformers for Language Understanding, CoRR (2018) vol. abs/1810.04805[Online]Available, http://arxiv.org/abs/1810.04805.
[70] V. Sanh, L. Debut, J. Chaumond, T. Wolf, DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter, CoRR (2019) vol. abs/1910.01108[Online]. Available, http://arxiv.org/abs/1910.01108.
[71] A. Conneau, et al., Unsupervised cross-lingual representation learning at scale, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online: Association for Computational Linguistics, Jul. 2020, pp. 8440–8451, https://doi.org/10.18653/v1/2020.acl-main.747.
[72] D. Kakwani, et al., IndicNLPSuite: monolingual corpora, evaluation benchmarks and pre-trained multilingual language models for Indian languages. ***Findings of the Association for Computational Linguistics: EMNLP 2020***, Online: Association for Computational Linguistics, Nov. 2020, pp. 4948–4961, https://doi.org/10.18653/v1/2020.findings-emnlp.445.
[73] S. Khanuja, et al., MuRIL: multilingual representations for Indian languages, CoRR (2021) vol. abs/2103.10730[Online]Available, https://arxiv.org/abs/2103.10730.
[74] N.L.L.B. Team et al., "No Language Left Behind: scaling Human-Centered Machine Translation," Jul. 2022.
[75] R. Dabre, H. Shrotriya, A. Kunchukuttan, R. Puduppully, M.M. Khapra, and P. Kumar, "IndicBART: a Pre-trained Model for Indic Natural Language Generation," Sep. 2021, doi: 10.18653/v1/2022.findings-acl.145.
[76] A. Paszke, et al., PyTorch: an imperative style, high-performance deep learning library, in: ***Advances in Neural Information Processing Systems***, 32, Curran Associates, Inc., 2019, pp. 8024–8035 [Online]. Available, http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.