IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 17 August 2023, accepted 30 August 2023, date of publication 7 September 2023, date of current version 21 September 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3312993

RESEARCH ARTICLE

HHSD: Hindi Hate Speech Detection Leveraging Multi-Task Learning

PRASHANT KAPIL[1], GITANJALI KUMARI[1], ASIF EKBAL[1], (Senior Member, IEEE), SANTANU PAL[2], ARINDAM CHATTERJEE[1,2], AND B. N. VINUTHA[2], (Member, IEEE)

[1]Department of Computer Science and Engineering, Indian Institute of Technology Patna, Patna 800013, India
[2]Wipro AI, Bengaluru 560035, India

Corresponding author: Prashant Kapil (prashant.pcs17@iitp.ac.in)

This work was supported by Wipro AI, India, as part of the Project HELIOS-Hate, Hyperpartisan, and Hyperpluralism Elicitation and Observer System.

ABSTRACT Hate speech is now a frequent occurrence on social media. Recently, the majority of study was devoted to identifying hate speech in languages with abundant resources (e.g., English). However, relatively few works are developed for languages with limited resources (e.g., Hindi, the third most widely used language on earth). In this study, Hindi Hate Speech Dataset (HHSD) is created following a novel hierarchical fine-grained four-layer annotation approach. The top layer separates the posts into hateful and non-hateful categories. The second layer further categorises hateful posts into explicit hateful and implicit hateful. The third layer is the multilabel tagging of the post into topics, such as political, religion, racism, or sexism. The fourth layer involves the identification of the targeted named entity, either explicitly or implicitly. Additionally, a thorough evaluation of the data annotation schema for trustworthy annotation is provided. The HHSD data is the largest multi-layer annotated corpora in Hindi compared with the existing multi-layer annotated data. Experiments on the dataset using the transformer-based approaches in single-task learning (STL) attain encouraging performances in accuracy and weighted-f1 score. The experiment leveraged multi-task learning (MTL) by including multiple related hate speech detection tasks from high-resource English and languages from the same linguistic family such as Urdu and Bangla with a transformer encoder as the shared layers to obtain a significant increment of 5.31% and 5.35% over STL in accuracy and weighted-f1 for layer A, 8.20%, and 22.83% for layer B. The MTL surpasses STL by 8.98% and 4.07% in exact match and hamming loss for layer C.

INDEX TERMS Transformers, multi-task learning, F1 score, accuracy, Shared layers.

I. INTRODUCTION

With the advancement of the Internet and the widespread acceptance of opinion-rich online resources, users have many options to express their thoughts in real time. However, these platforms are frequently abused to disseminate harmful and hateful messages that target specific people or groups. The prevalence of unpleasant and abusive content on social media sites is posing a significant problem for the government and technology firms. Thus, it is crucial to create automatic methods to stop the spread of hateful content and filter it out. Hate speech is commonly defined as any communication that disparages a person or a group based on some characteristics such as race, color, ethnicity, gender, sexual orientation, nationality, religion, or other characteristics [1].

Hate language can vary from offensive, aggressive, abusive, harassing, toxic or violent. The Google project named *Perspective*[1] defines *toxicity* as a rude, disrespectful, or unreasonable comment that makes the user leave the conversation. Therefore, it is crucial to identify detrimental posts and stop their spread over social networks to preserve social peace. The identification of hate speech on social media sites like Twitter, Facebook, etc., has received a lot of attention in recent years. Due to lesser regulation of hate speech in non-English speaking countries, the platform is more vulnerable to abuse. A new law in India requires social media companies to remove any illegal content within 36 hours of receiving it.[2]

The associate editor coordinating the review of this manuscript and approving it for publication was Aysegul Ucar.

[1]https://www.perspectiveapi.com/
[2]https://www.washingtonpost.com/technology/2021/10/24/india-facebook-misinformation-hate-speech/

Current research on hate speech analysis is oriented toward monolingual corpora. Even Hindi, the first language of 528 million people (43.63%) in India,[3] does not have sufficient labelled corpora. There will likely be 650 million internet users in India by 2023, which would cause the number of Hindi posts to rise dramatically. Recently, the Hindi-English code mixed data annotated for three subtasks were made publicly available via [3] and [4]. A Devanagari-based data (D-HOT) is created by [5] to establish a hate speech classifier in Hindi. Following the work of [5], the primary motivation of this paper is to create a novel data set covering multiple aspects of hateful posts in Hindi. The script for Hindi is Devanagari which is written as इस तरह से. References [6] and [7] argues that due to the tremendous variability in annotating hate speech, including definition, categories, annotation standards, types of annotators, and agreement of annotations, the nature and content of the datasets are more significant than the models developed. The majority of social media platforms use reporting and manual review methods, which are constrained by the reviewer's speed, ability to understand the evolution of slang, jargon, and familiarity with multilingual content [8]. In this study, the models are trained to leverage single-task and multi-task learning paradigms. To increase the performance metric of the classifier, the training data is further augmented with the existing English, Hindi, Urdu, and Bangla hate speech data in the multitask framework.

The key contributions of this work are as follows:

1) **Dataset**: A novel Hindi Hate Speech Dataset (HHSD) is created following a hierarchical fine-grained four-layer annotation approach. The first two are binary classification tasks, the third belongs to multi-label classification tasks and the fourth layer is named entity tagging of the targets. This dataset will be made available to the community for research purposes.
2) **Model**: The experiments are conducted using numerous cutting-edge models, such as convolution neural network (CNN), bidirectional long short-term memory (Bi-LSTM), multilingual-bert (M-BERT), language-agnostic bert sentence embeddings (LaBSE), multilingual representations for Indian languages (MuRIL), XLM-RoBERTa, and IndicBERT on the newly created HHSD in a single task learning fashion. The multi-task learning framework results are reported by taking two best-performing transformer encoders, viz., MuRIL and M-BERT as the shared layers.
3) **Analysis**: The model efficacy in a 5-fold cross-validation approach is examined by presenting qualitative and quantitative analysis. The statistical significance test is also performed to check whether the best model is indeed significant.
4) **Auxiliary data**: In the multi-task learning setup, low-resource languages with a high degree of resemblance to Hindi, such as Urdu and Bangla, are also used to expand the training set. Bangla, Urdu, and Hindi translations and transliterations are likewise derived from the English data that is readily available. A human evaluation score depicting the quality of translation and transliteration are also shown in Table 9.

[3]https://en.wikipedia.org/wiki/List_of_languages_by_number_of_native_speakers

The remainder of the article is structured as follows. The related background literature is presented in Section II. Section III discusses the resource creation and the annotation schema. Section IV describes the state-of-the-art techniques used for the experiment. In Section V evaluation metrics and the experimental setting are described. The results and error analysis are reported in Section VI, and the conclusion and suggested future work are presented in Section VII.

II. RELATED WORK

The advancement in deep learning techniques has widened the application of natural language processing tasks such as classification. The task of solving hate speech detection is overgrowing, but most of the data sets are available in English [2], [9], [10]. In general, the resource available for hate speech detection can be categorized into three settings [11]: (i) high resource setting, (ii) low resource setting, and (iii) zero resource setting. The majority of current research focuses on English and other high-resource languages. However, recently, a few attempts have been made to make the Hindi resources publicly available through shared tasks [3], [4], [12], [13] but due to the less available labelled data, detecting hateful content is a challenging task. In this section, the approaches leveraged to solve Hindi, Urdu, and Bangla hate speech detection is discussed.

A. HINDI

The existing work on Hindi mainly deals with the data mixed with Hindi and English. An annotated corpus of 4575 Hindi-English code-mixed text is presented by [14]. The experiment is done on a support vector machine (SVM) and random forest (RF) by utilizing features such as character n-grams, word n-grams, punctuations, negations, and hate lexicon. For the purpose of identifying hate speech in social media code-mixed text, [15] studied a number of techniques, including the sub-word level long short-term memory (LSTM) model and the Hierarchical LSTM model with attention based on phonemic sub-words. Reference [16] explored deep learning architectures like CNN, LSTM, and variants of BERT like M-BERT, IndicBERT, and monolingual RoBERTa to solve the hate and offensive detection on the data by [17]. The detection of code-mixed Hindi-English data is improved by including social media-based features in the model and additionally capturing the features of profane words [18]. A bias elimination algorithm is also developed to mitigate any bias from the proposed model.

A Hinglish offensive tweet (HOT) dataset was introduced by [19] for the multiclass categorization of offensive textual tweets in the Hindi-English code-switched language.

The proposed multi-input multi-channel transfer learning (MIMCT) based model uses multiple embeddings and secondary semantic features in a CNN-LSTM parallel channel architecture to outperform various transfer learning models. Recently, multi-layer annotated data such as [3], [4], [13], [17], and [38] has been released. A suite of functional tests i.e. HateCheckHIn is presented by [22] for Hindi hate speech detection models by combining the existing monolingual and multilingual functionalities. Reference [23] concluded that character level embedding, GRU, and attention layer are novel to hate speech detection in Hinglish code-mixed language. A dataset of 10,000 samples from different sources is created by [24] to train the model with Facebook pre-trained word embedding library to classify between hate and non-hate. Reference [25] experimented by aggregating six datasets in English, Hindi, and code-mixed Hindi to conclude that logistic regression added with TFIDF and POS features outperformed other monolingual models such as CNN-LSTM, BERT, and RoBERTa. A thorough examination of multilingual abusive speech in eight Indic languages from fourteen publicly accessible sources is shown by [26]. The experiments were carried out for numerous languages in a variety of circumstances, including ELFI (each language for itself), zero-shot learning, few-shot learning, model transfer, instance transfer, cross-lingual learning, etc. The effectiveness of transformer models like IndicBERT, M-BERT and transfer learning from already-trained language models like ULMFiT and BERT in order to identify hateful text in Hinglish is examined by [27]. For the purpose of identifying hate speech in Hinglish, the transformer-based interpreter and feature extraction model (TIF-DNN) beat current cutting-edge techniques. A 150K human labelled data (MACD) for five languages with 49% abusive class is created by [28]. The user comment is crawled from 70K users from the social media platform-Sharechat. An abusive content detection model i.e. AbuseXLMR, pre-trained on a large number of social media comments in 15 Indic languages which outperform XLM-R and MuRIL on multiple Indian datasets is released.

B. BANGLA AND URDU

This section discusses the literature for the two low-resource languages: Bangla and Urdu.

A two-layer manually annotated Bangla aggression dataset (BAD) is presented by [29]. The experiment applies various machine learning algorithms (LR, SVM, RF, NB), deep learning algorithms (CNN, LSTM, CNN+BiLSTM), and deep transformer-based models like M-BERT, Distil-BERT, Bangla-BERT, and XLM-R. A dataset of 30000 Bengali user comments from Facebook and Youtube comments and has 10,000 hate posts is released by [30]. The comments were collected from 7 categories: sports, entertainment, crime, religion, politics, celebrity, tik tok and memes. The experiment leveraged three-word embeddings: word2vec, fasttext, and BengFast, and machine learning models such as SVM, LSTM, and BiLSTM. Reference [8] presented a lexicon of 621 hateful words in Roman Urdu. To identify hate speech, five fine-grained labels were added to the dataset in Roman Urdu. The transfer learning abilities of five pre-existing multilingual embedding models to Roman Urdu through extensive experiments are examined. Reference [31] explored different data augmentation techniques such as synonym replacement, random swap, random insertion, random deletion, MT5 text generation, and M-BERT for the improvement of hate speech classification in Roman Urdu.

C. MULTI-TASK LEARNING

Multi-Task learning (MTL) [32]: It seeks to enhance the learning of a model for the classification task Ti by utilising the knowledge in some or all of the ''m'' learning tasks, given that all of them or a subset of them are connected.

$$\{T_i\}_{i=1}^{m} \tag{1}$$

A deep shared-private multi-task learning framework to leverage valuable information from multiple related tasks such as hate detection, racism detection, aggression detection, harassment detection, etc is presented by [33]. Reference [34] focuses on hate speech detection in Spanish corpora and proposes an MTL model to benefit from associated tasks like polarity and emotion categorization. Reference [35] presented MT-GAN-BERT, a new architecture that extends BERT-based models with semi-supervised learning while using a single encoder in multi-task learning.

III. CORPUS CREATION

The procedure for gathering data is described in the next part, along with the annotation schema that was given to the annotators. Research and development in this area have been hampered by the lack of a significant Hindi annotated corpus for hate messages. We, therefore, set out to create new data.

A. DATA CRAWLING AND PROCESSING

The proposed data set is constructed from Hindi tweets crawled using Twitter search API.[4] As it is a common practice, the stream was filtered based on a list of frequent words in Hindi and by Twitter's language identification mechanism. The data collection covers a wide period from May 2021 to September 2021. The keywords and topics, written in Hindi script related to political, religion, racism, and sexism were identified, which were in the news in recent times and for which hate speech can be expected. The abusive lexicons in Hindi script were also collected to crawl explicit hateful posts. The key objective of this method is to make sure that a balanced mix of hateful and non-hateful tweets makes up the final dataset. Table 1 and Table 2 depicts the important keywords and topics that were used to crawl the posts.

Selecting relevant tweets for Annotation: To select the relevant tweets from the large pool of unlabeled data for the final annotation, a set of tweets was filtered out based on the weakly generated probability value. A classifier based on

[4]https://developer.twitter.com/en/docs/twitter-api/v1/tweets/search/api-reference/get-search-tweets

TABLE 1. Topics to collect the data.

Topics
नागरिकता संसोधन अधिनियम (CAA), एनाअरसी(NRC), अनुच्छेद 370 (article 370), राम मन्दिर (ram mandir), बीफ बैन(beef ban), तीन तलाक् (triple talaaq), अवार्ड वापसी (award wapsi), विमुद्रीकरण(demonetization) जीएसटी (GST), शराबबंदी (liquor ban), मन की बात (mannkibaat), पुलवामा अटैक (pulwama attack), शाहीन बाग (saheenbagh), स्वच्छ भारत (swachh bharat), सबरीमाला मंदिर (sabrimala mandir), फतवा (fatwa), लव जिहाद (love jihad), आजादी मार्च (AzadiMarch)

TABLE 2. Domain specific tokens.

Domain	Tokens
1.Political	सरकार(sarkar/Government), भाजपा(bajpa/BJP), कांग्रेस(congress), मोदइइ(modi), राज्य(rajya/state) केंद्र(kendra/central), संसद(sansad/Parliament) राजनीति(rajneeti/politics), कानून(kanoon/law)
2.Religion	होली(holi), हिन्दू(hindu), धर्म(dharms/religion), मंदिर(mandir/temple), इस्लामा(islam) मस्जिद(masjid/mosque), बाइबिल(Bible), बौद्ध(bauddha) आरती(aarti), भगवान(bagwaan) , अल्लाह(allah)
3.Racism	कश्मीरी(Kashmiri), दलित(dalit), जाति(jaati), रोहिंग्याओ(rohingiyaon)
4.Sexism	छिनाल(C****l,B***h), रखैल(rakhael,wh**e), लिपस्टिक(lipstick) किन्नर(Kinnar, Transgender)

convolution neural network (cnn) C_i is trained using eight publicly available Hindi datasets (see Table 6). The unlabelled tweet i obtained in the crawling is passed through the trained models C_i to generate a weak label based on the probability value p. A set S_h of tweets with p *(hateful)*$\geq$ 0.65, and set S_{nh} of tweets with p *(non-hateful)* $\geq$ 0.85 is filtered out to give to the annotators.

Figure 1 explains the data creation process. To prepare the collected tweets for the annotation, we applied some pre-processing steps, which are as follows:

1) The encoding was converted to UTF-8.
2) The removal of user handles, punctuations, URLs, and numbers (0-9). The emoticons were substituted with relevant text.
3) The tweet should not contain any links, pictures, or videos as they might contain information not available to the annotators.

B. HIERARCHICAL ANNOTATION SCHEMA

The annotation process has been done by five annotators possessing good knowledge of Hindi and linguistics. The annotators were at a higher education level (Masters, PhD.). The annotators were made aware of the posts' hatefulness before they began their annotations. In the HHSD dataset, we use a hierarchical annotation schema for four layers to distinguish whether (A). post is hateful or not, (B). Implicit hateful or Explicit hateful, (C). its associated domain, and (D). it's named entity target. The following subsection goes into further depth about each layer. Figure 2 explains the flow of the annotation covering all four layers.

1) LAYER A: HATEFUL LANGUAGE IDENTIFICATION

Objective: In this layer, classes are divided into two distinct categories i.e. Hateful and Non-hateful.

Hateful: The Language that is intended to be disparaging, humiliating, or insulting to the members of the group or an individual based on race, gender, ethnic origin, sexual orientation, disability, religion, or colour [2], [36].

Non-hateful: Posts that do not contain any hateful content.

2) LAYER B: CATEGORIZATION OF HATEFUL POSTS

Objective: This layer further categorizes hateful tweets into two types of hate i.e. Explicit hateful and Implicit hateful.

Explicit hateful: Any speech or text that displays hate-either through the usage of a particular type of lexical item or lexical feature that is deemed hateful and certain syntactic structures is regarded to be explicit hate.

Implicit hateful: Any post where hate is subtly communicated. It is a hidden attack on the victim and is frequently disguised as (false) courteous interactions (through the use of conventionalized polite structures).

3) LAYER C: MULTI-LABEL TAGGING OF HATEFUL LANGUAGE

Objective: This layer consists of multi-label tagging of the hateful tweets into four domains viz. Political, Religion, Racism, and Sexism. The definition of each domain according to the Cambridge dictionary[5] is given as follows:

Political: The activities of the government, members of law-making organizations, or people who try to influence the way a country is governed.

Religion: The belief in and worship of a god or gods, or any such system of belief and worship.

Racism: Policies, behaviours, rules, etc. that result in a continued unfair advantage to some people and unfair or harmful treatment of others based on race.

Sexism: The belief that the members of one sex are less intelligent, able, skilful, etc. than the members of the other sex, especially that women are less able than men.

[5]https://dictionary.cambridge.org/dictionary/english

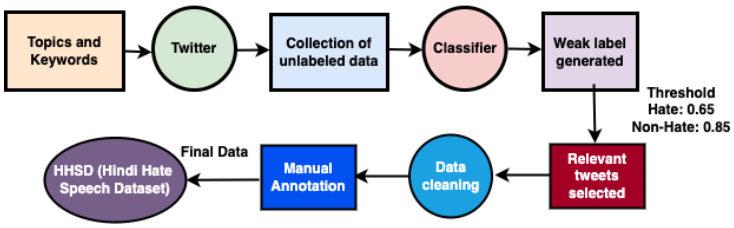

FIGURE 1. Dataset development steps.

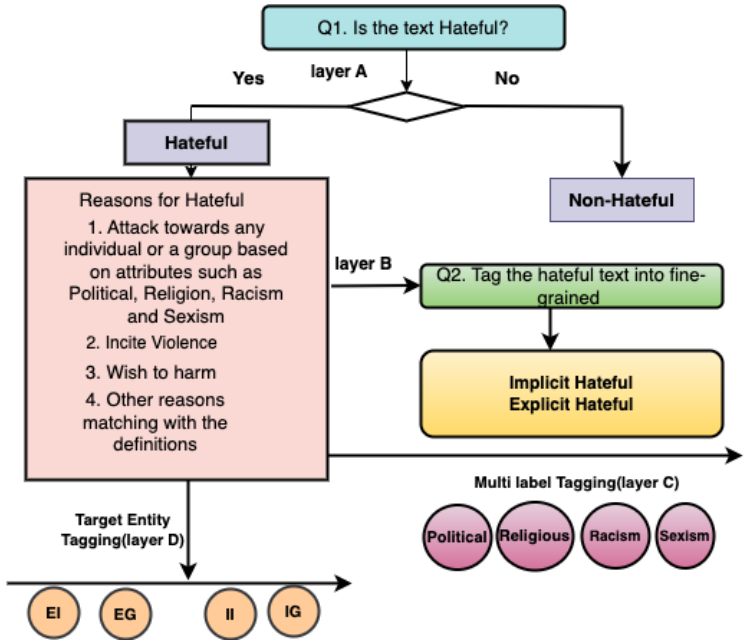

FIGURE 2. Annotation process.

4) LAYER D: TARGET ENTITY IDENTIFICATION

The hateful tweets consist of individuals and groups targeted implicitly or explicitly. We tag all relevant, targeted named entities into four types depending on their presence in the post: Explicit Individual (EI), Explicit Group (EG), Implicit Individual (II), and Implicit Group (IG).

C. ANNOTATION INSTRUCTIONS

Pilot Annotation: Given the subjective nature of the task, the annotators were provided with multiple examples from different classes to get an idea. In the pilot annotation, all five annotators were given the same set of 200 tweets to annotate by following the annotation schema in Figure 2. The purpose of this round was to check the agreement between the annotators. After the first round of the pilot work, we continued with the final annotation and evaluated the quality of the annotation.

Main Annotation: We chose to move forward with a batch of 500 tweets that included distinct samples for each annotator. To create a trustworthy dataset, the annotation quality was examined after each batch.

D. ANNOTATION CHALLENGES AND SOLUTIONS

1) Lots of Unlabelled data and small teams: Annotation is a time-consuming and laborious process. It is very challenging to have the resources capable of handling high-volume labelling.
   Solution: A classifier is trained on existing data to obtain a silver label for the unlabelled set, which will be given to the annotators to get the gold label.
2) Keeping human bias out of AI solution: Human bias is one of the issues in reliable annotation that can hamper the efficacy of the classifier.
   Solution: To mitigate bias, large amounts of training data are collected, and a diverse group of annotators is recruited to ensure the data is as universally applicable as possible.
3) Ambiguity: It is very challenging for the annotators to tag ambiguous tweets. For example:

संजय भैया, कुत्ते के भौकने पर ध्यान नही देते
Transliteration: Sanjay Bhaiya, Kutte ke bhaukne par dhyan nahi dete.
Translation: Sanjay Bhaiya No need to pay attention to the barking dog.
The aforementioned tweet carries two meanings. The former is attacking some human by comparing with the dog, whereas the latter refers to a dog only.
Solution: The two-round annotation discussion is done to resolve this type of issue.

4) Contextual information: The tagging of a tweet requires contextual knowledge for some tweets to correctly tag it. For example:
लुटेरी दुल्हन सुन कर एंटोनियो माइनो की याद आ जाती है
Transliteration: Luteri dulhan sun kar antonio maino kii yaad aa jati hai.
Translation: Hearing the robber bride, one remembers Antonio Maino.
This tweet requires the annotator to know that the second underlined phrase is a name of a political figure who is being targeted by using a derogatory phase.
Solution: The two-round annotation discussion is done to resolve these issues.

E. Inter-Annotator Agreement (IAA)

The Fleiss Kappa score is used [37] to assess the annotator scores for the first three levels. It is a metric for evaluating the degree of agreement between two or more raters known as inter-rater agreement. The high value indicates the correctness of the data. It shows how clear the annotation guidelines are, how uniformly the annotators understood it, and how reproducible the annotation task is. It is a vital part of both the validation and reproducibility of classification results. For the first, second, and third layers, the IAA is 86%, 76%, and 82%, respectively. The Fleiss Kappa's interpretation is shown in Table 3.

F. DATA STATISTICS

In this paper, the newly created HHSD is used to evaluate the performance metric on training with deep neural network-based approach.

The detailed statistics for data set are shown in Table 4. Table 5 enlists the different types of hate attack that is present in the data.

G. AUXILIARY DATA

The experiment also leverages related task data from high-resource languages such as English and other data from semantically similar languages such as Urdu and Bangla. In total, eleven English (E), eight Hindi (H), three Urdu (U), and one Bangla (B) dataset were used to augment the training data. Table 6 shows the information about the existing datasets in Hindi, and Table 7 depicts the statistics for English, Bangla, and Urdu. As the main aim is to increase the performance metric of Hindi data, we increase the training data in Hindi (Devanagari), Bangla, and Urdu by obtaining the translation from English (E) → Devanagari (D), English (E) → Bangla(B) and English (E) → Urdu (U) using Google translate. The transliterated version of Devanagari (D) → Roman Devanagari (RD), and Bengla (B)→ Roman Bangla (RB) is obtained using Indic Trans [45] to increase the training sample. The class-wise statistics for all the English, Hindi, Urdu, and Bangla are shown in Table 8. The eight Hindi data sets were denoted from $H_1 \ldots H_8$, three Urdu data from $U_1..U_3$. The size of English (E) data is the summation of all eleven, and B_1 is the Bangla data.

TABLE 3. Fleiss Kappa interpretation.

K	Interpretation
0.0-0.2	Poor Agreement
0.21-0.4	fair Agreement
0.41-0.60	Moderate Agreement
0.61-0.80	Good Agreement
0.81-1.00	Very good Agreement

TABLE 4. Statistics of HHSD.

layer A	layer B	layer C
Hateful:7311	Implicit hateful: 2432 Explicit hateful: 4879	Political: 4242 Religion : 2266 Sexism : 329 Racism: 255
Non-Hateful:7472	-	-

H. HUMAN EVALUATION

The quality of the translation and transliteration obtained from google translate and IndicTrans is manually measured on a sample of 500 tweets based on fluency, and content preservation. Each tweet was given a Likert scale rating from 1 (worst) to 5 (best) for each of the two evaluation criteria. The total score is averaged to produce the final result.

Fluency: It is used to measure the fluency of the grammar correctness in the output text [46].

Content preservation: It is a measure of degree of the preservation in the translation and transliteration. This also calculates the degree of hatefulness preserved.

Table 9 presents the human score to measure the quality of translation and transliteration.

Challenges in language adaptation: While using data from other languages some challenges are bound to happen. It can be seen from Table 9 that the quality of the transliteration is surpassing the quality of the translation. There is an error rate of 1.9, 1.8, and 2.2 in fluency and 1.9, 1.6, and 1.9 in the content preservation while translating the English posts to Devanagari, Urdu, and Bangla. However there is a significant drop in the error rate in fluency and content preservation while transliteration. The error rate of 1.2, and 0.7, and 1.1, and 0.8 in fluency and content preservation is observed

TABLE 5. Variants of hate attack in the HHSD.

Categories	Examples
Negative stereotypes ethnic slur	जिहादी कौम(Jihadi Kaum) सनातनद्रोही(Sanatandrohi) मुल्लों (Mullo)
Professions and occupations	दलाल मीडिया(Dalal media) दलाल पत्रकार (Dalal Patrakaar) अनपड मंत्री. (Anpad Mantri) बिकाऊ मिडिया (Bikao media)
Physical disabilities and diversity	अपंग (Apang) विकलांग व्यक्ति(Viklaang Vyakti) अंधा(Andhaa) अक्षम(Aksham)
Cognitive disabilities and diversity	अनपड(Anpad) मूर्ख (Murkha) मंदबुद्धित्मअनदहअबउदहइज बुद्धूत्भउददहहउज
Moral and behavioral defects	झूठे (Jhuthe) चमचे (Chamche) चाटुकार(chatukar) छल कपट(Chal Kapat) बेशर्मी(Besharmi)
Social and economic disadvantage	भिखारी(Bhikari) फटेहाल(Phatehaal)
Words related to prostitution	छिनाल(Chinal) रखैल (Rakhael) रंड (Randii) दलाल (Dalaal) भड़ुए(Bhaduye)
Obfuscation of slangs	रे बहिचो (Re Bahicho) भैंसचोर(Bhainschor) त्रुटिया(Trutiya) भोमसडीके (Bhomasdike) चुस्लामी (Chuslami) तिहारी कमीने(Tihari kamine) हुतिया(Hutiya) बकचोक(Bakcho)
Animal Picturization	काले तीतर(Kale titar) आस्तीन के ही सांप निकले(Astin ke hii saanp nikle) पालतू कुत्ता (Paltu Kutta) रंगा सियार(Ranga Siyaar) मगरमच्छ के आंसू बहाता है(Magarmacha ke aansu bahata hai) गंदी नसल के कुत्तों(Gandi nasal ke kutto) काले कौवे(kaale kauye) तेरी बंदर जैसे थोब्दे पर(Teri bandar jaise thobde par)
Implicit Individual	मोमता बानो(Momta bano) सॉपनाथ(Saanpnath) नागनाथ (Naagnath) शूर्पणखा(surpanakha) होलिका(Holika) घुसपैठिए दामाद(Guspaithiye damaad)
Implicit Group	तुम जैसे जयचंदो(Tum jaise jaichando) तुगलकी शासन व्यवस्था (Tuglaki Shashan Vyavhashata) भारत भर के ओछे और लिचडो (Bharat bhar ke oche aur lichado) भरुआ पार्टी(Bharua) सभी मूक दर्शक(sabhi mook darshak) इन झंडुओ (In jhanduon)
Explicit Individual	हलाला की पैदाइश(halala kii Paidaish) रंडी सती सावित्री(Randi sati sawitri) प्रशांत भूषण जैसे देशद्रोही दोगले (Prashant bhushan jaise deshdrhi dogle) स्वार्थी नेता(Swarthi neta) देशद्रोही (Deshdrohi)
Explicit Group	खूखार जिहादी कौम(Khoonkhaar jihadi kaum) दलाल मीडिया(Dalaal media) मोदी सरकार की उदासीनता(Modi sarkaar kii udashintaa) बिकाऊ मिडिया(Bikau media) गोदिमीडिया(godi mdeia) सुअर की औलादों (suwar kii aulado)

TABLE 6. Publicly available Hindi datasets used in the experiment.

Dataset	# layers	classes	IAA	Script	Medium
Bohra et al. [14]	1	Hate, Neutral	84	Roman Hindi	Twitter
Kumar et al. [21]	1	CAG, OAG, NAG	36	Roman+Devanagari	Facebook and Twitter
Kumar et al. [38]	2	1: CAG, OAG, NAG 2: Gendered, Non-gendered	69	Roman+Devanagari	Facebook and Twitter
Mandl et al. [4]	2	1: Hate, Non-offensive 2: Hate, offensive, profanity	65	Roman+Devanagari	Twitter
Mandl et al. [17]	2	1: Hate, Non-offensive 2: Hate, offensive, profanity	72	Roman+Devanagari	Twitter
Jha et al. [5]	1	Hate, Non-Hate	83	Devanagari	Twitter
Bhardwaj et al. [39]	1	Fake, Hate, Offensive, Defame	–	Roman+Devanagari	Reddit
Mathur et al. [19]	1	Hate, Abusive, Non-Hate	98	Roman Hindi	Twitter
HHSD	4	1: Hateful, Non-hateful 2: Implicit hateful, Explicit hateful 3: Political, Religion,Racism, Sexism 4: II, IG, EI, EG	86 76 82 -	Devanagari	Twitter

TABLE 7. Publicly available English, Bangla, and Urdu datasets used in the experiment.

Dataset	# layers	classes	IAA	Script	Medium
Davidson et al. [2]	1	Hate, Offensive Neutral	92	English	Twitter
Waseem et al. [41]	1	Sexism, Racism, Neutral	84	English	Twitter
Kumar et al. [12]	1	CAG, OAG, NAG	72	English	Facebook and Twitter
Zampieri et al. [42]	1	Offensive, Non-offensive	83	English	Twitter
Golbeck et al. [43]	1	Harassment, Non-Harassment	84	English	Twitter
De Gilbert et al. [9]	1	HOF, Non-Hate	61	English	Stormfront
Basile et al. [44]	1	Hate, Neutral	62	English	Twitter
Mandl et al. [3]	3	1: Hate, Neutral 2: Hate, Offensive, Profanity 3: Targeted Insult, Untargeted	36	English	Facebook+Twitter
Bhattacharya et al. [13]	2	1: CAG, NAG, OAG 2: Gendered, Non-gendered	69	English	Facebook+Twitter+Youtube
Mandl et al. [4]	2	1: HOF, NOT 2: Hate, Offense, Profanity	-	English	Twitter
Founta et al. [36]	1	Hate, Abusive, Spam, Normal	70	English	Twitter
Rizwan et al. [8]	1	Hate, Non-Hate	–	Urdu	Twitter
Khan et al. [40]	1	Hate, Non-Hate	87.2	Urdu	Twitter
Khan et al. [40]	1	Hate, Non-Offensive	71.4	Urdu	Twitter
Bhattacharya et al. [13]	1	Hate, Non-Hate	–	Bangla	Twitter

while transliterating the Bangla script to Roman Bangla, and Devanagari to Roman Hindi. As the fluency and content preservation obtained is >2.5 for all the cases, the auxiliary data is augmented.

TABLE 8. Statistics of auxliary datasets used in the experiment.

Datasets	Class
E	Hate: 104056 Non-Hate: 126780
U_1	Hate: 4664 Non-Hate: 5349
U_2	Hate: 880 Non-Hate: 2690
U_3	Hate: 1425 Non-Hate: 3575
B_1	Hate: 2659 Non-Hate: 3312
H_1	Hate: 484 Neutral: 1516
H_2	Hate: 3074 Neutral: 2909
H_3	Hate: 1916 Non-Hate: 3959
H_4	Hate: 12962 Non-Hate: 3008
H_5	Hate: 3042 Non-Hate: 3139
H_6	Hate: 3834 Non-Hate: 4358
H_7	Hate: 2068 Non-Hate: 1121
H_8	Hate: 1299 Non-Hate: 2249
E → D	Hate: 104056 Non-Hate: 126780
E → U	Hate: 104056 Non-Hate: 126780
E → B	Hate: 104056 Non-Hate: 126780
B → RB	Hate: 104056 Non-Hate: 126780
D → RD	Hate: 104056 Non-Hate: 126780

TABLE 9. Human evaluation.

Translation	Fluency	Content Preservation
English → *Devanagari*	3.1	3.1
English → *Urdu*	3.2	3.4
English → *Bengla*	2.8	3.1
Transliteration	**Fluency**	**Content Preservation**
Bengla script → *RomanBengla*	3.8	4.3
Hindi-Devanagari → *Roman − Hindi*	3.9	4.2

IV. METHODOLOGY

The experiment is carried out using seven single-task learning (STL) and eight multi-task learning (MTL) frameworks based on deep neural network-based architectures as shown in Figure 3. A detailed explanation of all the models is as follows:

CNN: This model adopts the architecture proposed by [47], which has five primary layers: the input layer, the embedding layer, the convolution layer, the pooling layer, and the fully connected layer.

BiLSTM [63]: It is a type of long short-term memory (LSTM) that uses two LSTMs to calculate information from the past and the future. The hidden state at each time step is the concatenation of the forward and backward states for the given time sequence. $h_t = [\overrightarrow{h_t^1}, \overleftarrow{h_t^1}]$, hence the input passed

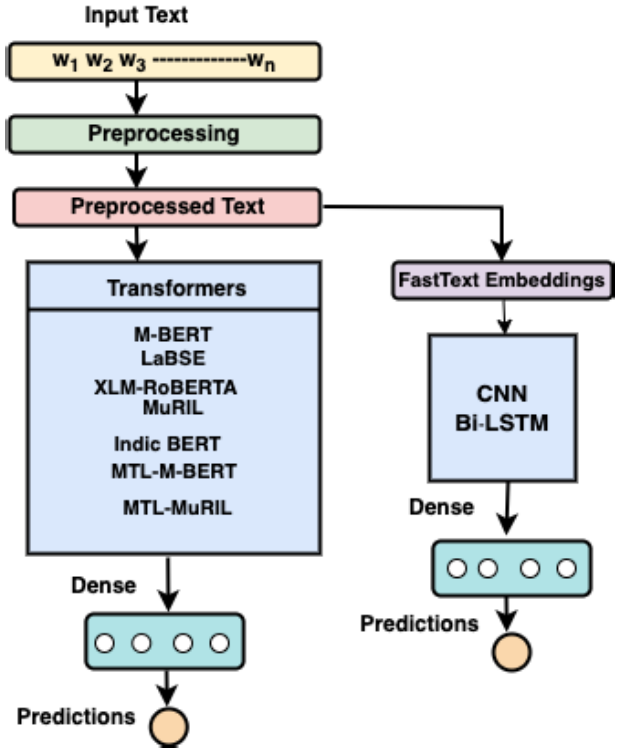

FIGURE 3. Flowchart of proposed methodology.

to the next layer is $[e(w_1);h_1^1],[e(w_2);h_2^1],\ldots..,[e(w_n);h_n^1]$ The next layer output will be $h_2 = (h_1^2,h_2^2\ldots h_n^2)$. The input passed to the next layer will be $[e(w_1);h_1^1h_2^1,e(w_2);h_1^2h_2^2\ldots]$. The FastText [20] is used as word embedding to represent the words into a real-valued vector for CNN and BiLSTM.

Multilingual-BERT: [48] introduced **M-BERT** i.e., Multilingual Bidirectional Encoder Representation from Transformers to pre-train deep bidirectional representations from unlabeled texts by joint conditioning on both left and right context in all layers. The classifier can be created by adding just one more output layer to the pre-trained BERT model. It generally learns from two training objectives described as follows:

1. **Masked Language Modeling (MLM)**: The model randomly masks some of the tokens from the input, and the goal of the model is to fill that mask with an appropriate token. This allows the model to focus on both left and right contexts.
2. **Next Sentence Prediction (NSP)**: It pre-trains text-pair representations to determine whether or not two phrases will follow one another.

The BERT's multilingual version can operate with 104 different languages. Every sequence begins with a distinct classification token as the first token ([CLS]). The aggregate sequence representation for the classification task is the last hidden state corresponding to this token.

XLM-RoBERTa [49]: It is a transformer model trained by sampling streams of text in 100 languages and predicting the masked tokens in the input by MLM objective. 2.5 TB

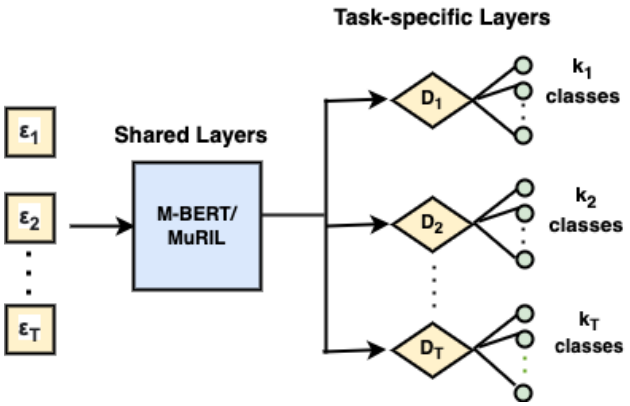

FIGURE 4. Multi task learning (MTL).

Algorithm 1 Training of a MT-DNN Model Θ

1. Load the Encoder parameter acquired during the pretraining
2. Initialize $D_1, D_2, \ldots, D_T$ randomly
3. **for** T in $1, \ldots, T$ **do** //Prepare the data for T tasks
   4. Divide the data of t^{th} task into mini batches so that $\epsilon_t = U_j B_j^t$
5. **end for**
6. **for** epoch in $1, \ldots, epoch_{max}$ **do**
   7. Merge datasets: $\epsilon = \epsilon_1 \cup \ldots \cup \epsilon_T$
   8. shuffle ϵ
   9. **for** B^t in ϵ **do** // B^t is a mini batch of task t
      10. Use the shared BERT encoder to encode $h_{CLS}^{B^t}$
      11. Classify $h_{CLS}^{B^t}$ using D_t against k_t classes
      12: Compute L_t loss as the Cross-entropy w.r.t the k_t classes
      13: Compute gradient: $\nabla(\Theta)$ using L_t
      14: Update the entire model: $\Theta = \Theta$ - v $\nabla(\Theta)$
   15. **end for**
16. **end for**

of common crawl data in 100 languages are used to train it. It outperforms M-BERT across cross-lingual classification, especially for low-resource languages. To apply sub-word tokenization on the raw input, sentence piece [50] with unigram language model [51] is employed. The sample of batches from different languages is selected using the same sampling distribution as in [52].

LaBSE [53]: It adopts multi-lingual BERT to produce language-agnostic sentence embedding for 109 languages. This model combines the masked language model (MLM) and translation language model (TLM) [52].

The training takes place using two types of data.

1. **Monolingual Data**: The data from Wikipedia and Common Crawl is collected, followed by heuristics from [54] to remove the noisy text. After the pre-processing stage, 17B monolingual data were obtained for the training.

2. **Bilingual Translation Pairs**: The web pages were translated using the bitext mining system similar to the approach by [55] to obtain the translated corpus.

MuRIL [56]: A multilingual language model that has been specifically created for Indian languages was trained using text corpora from 16 Indian languages known as "IN." The training objectives include MLM and TLM, among others. The TLM uses pairs of translated and transliterated documents to train the model, whereas the MLM only uses monolingual text. 4096 is the maximum global batch size, 512 is the maximum sequence length, and 1M steps are learned. With a learning rate of 5e-4, the Adam optimizer has a total of 236M learned parameters.

IndicBERT [57]: It is a multilingual ALBERT model that was developed using extensive corpora that included 12 important Indian languages. It has much fewer parameters than M-BERT and XLM-R, but it manages to give a state-of-the-art performance on the classification task. The joint training of all the languages is done using the single shared model to utilize the relatedness of the Indian languages. Table 10 consists of the source of the training data and the number of trained parameters for the transformer encoder leveraged for the experiments.

A. TRAINING OF MTL

The architecture of the Multi-task deep neural network (MT-DNN) model is shown in Figure 4. It adheres to the approach proposed in [58] to solve only classification tasks. An encoder based on BERT represents the shared layers for all T tasks. The shared layers aim to capture common features. The specific categorization tasks are implemented by the output layers $D_1, \ldots, D_T$. The encoder captures the contextual information for each word in each input example (either a phrase or a group of sentences) made up of n word-pieces by using self-attention to generate a sequence of contextual embeddings. These are (n + 2) vector representations in R^d, i.e., (h_{CLS}, h_{w_1}, …, h_{w_n}, h_{SEP}). The h_{CLS} corresponds to the d-dimensional representation of the input sequence, while h_{w_1}, …, h_{w_n} represent the d-dimensional embeddings for the individual word pieces. The h_{CLS} is retained for the sentence-based classification and passed as input to the D_t layer to classify the input sentence w.r.t. the task t = 1, …,T. The training procedure of MT-DNN is reported in Algorithm 1. Input examples generally belong to datasets $\epsilon_1, \ldots, \epsilon_T$ that are specific for each task and have a different set of labels. The MT-DNN requires that each dataset is shuttered in mini-batches B_j^t, each containing valid examples from the same task t. In each epoch, a random mini-batch B_j^t is selected, all the inputs are encoded leveraging the same BERT encoder and the generated representation $h_{CLS}^{B^t}$ is classified by the D_t. The task-specific loss L_t is computed that is used to update the weights for the entire model via back-propagation.

TABLE 10. Transformer encoder.

Encoder	Training data	#Parameters
M-BERT	Wikipedia articles for top 102 languages	110M
XLM-R	Common Crawl	270M
LaBSE	Common Crawl+ Wikipedia	471M
MuRIL	Common Crawl + Wikipedia + Others	236M
IndicBERT	IndicCorp + IndicGlue	18M

Following this way, the output layer D_t is fine-tuned for the t^{th} task but, most importantly, BERT encodings are at the same time optimized for all the tasks. M-BERT and MuRIL were the best two single-task learning models that we chose to employ as the shared-BERT encoder in the multi-task learning framework. We developed four variants of multi-task learning models based on training data in Hindi (H), Bangla (B), Urdu (U), and the combined data from Hindi-Bangla-Urdu (HBU). In this paper, we are reporting only the results obtained on HHSD data from the MTL paradigm. The entire experiment is completed by assigning separate MTL settings to layer A, layer B, and layer C.

V. EXPERIMENTS

This section presents the experimental setups and the evaluation metrics.

A. EXPERIMENTAL SETUP

All deep learning models were created using Keras [59], a neural network tool, with Tensorflow [60] as the backend. We performed 5-fold cross-validation to use 80% for tuning the batch size and learning epochs and test the optimized model on 20% held-out data. The network is optimised using the *Adam* [61] optimizer, with categorical cross-entropy as the loss function. CNN employs 100 filters, with a kernel width that spans from 1 to 4. There are 100 hidden nodes in the BiLSTM. The value for bias is randomly initialized to all zeros, *Relu* activation function is employed at the intermediate layer, and *Softmax* is utilized at the last dense layer. The pre-trained FastText word embeddings [20] is used to initialize the non-BERT model. We use a learning rate of 0.001 for the non-transformer model and 2e-5 for the transformer models. The transformers library is loaded from Hugging Face.[6] It is a Python library providing a pre-trained and configurable transformer model useful for various NLP tasks.

B. EVALUATION METRICS

The Accuracy and Weighted-F1 scores have been used to report the evaluation results for layer A and layer B. The Exact match and Hamming loss [62] were employed as metrics to assess the effectiveness of multi-label classification for layer C in HHSD.

Hamming loss: The fraction of labels that are incorrectly predicted.

Exact match: The percentage of samples that have all their labels classified correctly.

[6]https://huggingface.co/models

VI. RESULTS, COMPARISON AND ANALYSIS

We present 5-fold cross-validation results for HHSD in Table 11 evaluated on state-of-the-art approaches. The results obtained leveraging STL and MTL are discussed in a separate section.

Single-task learning:

layer A: The M-BERT obtained highest accuracy and weighted-f1 of 85.82%, and 85.67%. This is followed by MuRIL which obtained 84.50% and 84.46% accuracy and weighted-f1.

layer B: The M-BERT obtained highest accuracy and weighted-f1 of 72.52%, and 56.97%. This is followed by MuRIL with 70.81% and 56.13% accuracy and weighted-f1 score.

layer C: The IndicBERT surpasses the other models by obtaining the exact match and hamming loss of 53.52% and 14.75%. This is followed by MuRIL with a score of 53.12% and 15.16%.

Multi-task learning:

The multi-task learning set-up leverages multiple data from Hindi, Bangla, and Urdu scripts. The four combinations of MTL are set up for M-BERT and MuRIL. The first three MTL is taking data from three languages one at a time, and the fourth one is taking all the languages.

layer A: The model is performing best when all three languages are simultaneously trained in the MTL fashion. The M-BERT and MuRIL obtained highest accuracy and weighted-f1 of 91.13% and 91.02%. It is interesting to note that M-BERT and MuRIL, when trained only with Hindi data in MTL, obtain a significant score. The reason for this performance is due to a large number of Hindi data available for the training. The inclusion of Bangla is outperforming the results obtained from Urdu.

layer B: The M-BERT trained with only Hindi tasks outperformed the M-BERT trained with all the language tasks by 1.27% and 0.45% in accuracy and weighted-f1 score. the inclusion of Bangla and Urdu hampered the performance of the model. In the MuRIL setup, the model is performing best with joint training of all the languages to surpass the Hindi-only model by 0.30% and 0.96% in accuracy and weighted-f1 score.

layer C: In this layer, the MuRIL with all the languages obtained maximum exact match and hamming loss of 62.10%, and 10.68%. This is followed by Hindi, Bangla, and Urdu. The M-BERT with all the languages obtained slender improvement over the model using only Hindi, Bangla, and Urdu data alone.

A. QUANTITATIVE ANALYSIS

Table 12 and Table 13 present the confusion matrix obtained by the best-performing model for layers A and B. The best performers for layer A and layer B for HHSD are M-BERT fine-tuned with all three languages and M-BERT trained with only Hindi data. It can be seen that the misclassification rate in the best-proposed model for *hateful* is 9.35% in layer A,

TABLE 11. Evaluation results on HHSD.

Models	layer A		layer B		layer C	
	Accuracy	Weighted-F1	Accuracy	Weighted-F1	Exact Match	Hamming loss
CNN	81.28	80.99	66.22	52.30	46.19	17.65
BiLSTM	82.87	82.74	68.85	54.45	49.68	15.80
M-BERT	85.82	85.67	72.52	56.97	50.67	15.72
LaBSE	83.89	83.86	69.47	54.84	47.62	17.61
XLM-RoBERTa	75.07	75.21	66.16	40.71	31.11	23.06
MuRIL	84.50	84.46	70.81	56.13	53.12	15.16
IndicBERT	84.48	84.32	70.52	52.97	53.52	14.75
$M-BERT_H$-MTL	90.36	90.32	**80.72**	**79.80**	59.64	11.62
$M-BERT_B$-MTL	89.10	89.29	76.25	75.63	58.92	12.14
$M-BERT_U$-MTL	88.91	89.20	76.13	75.51	58.46	12.72
$M-BERT_{HBU}$-MTL	**91.13**	91.01	79.45	79.35	59.95	11.68
$MuRIL_H$-MTL	90.61	90.20	79.12	78.82	61.95	10.72
$MuRUIL_B$-MTL	89.73	89.20	75.57	75.31	60.86	11.02
$MuRIL_U$-MTL	89.26	89.20	76.13	75.92	60.42	12.12
$MuRIL_{HBU}$-MTL	90.91	**91.02**	79.42	79.78	**62.10**	**10.68**

TABLE 12. Confusion matrix of D_1 (layer A) ($M-BERT_{HBU}$-MTL).

Class	Hateful	Non-hateful
Hateful	6627	684
Non-hateful	626	6846

TABLE 13. Confusion matrix of D_1 (layer B) ($M-BERT_H$-MTL).

Class	Implicit hateful	Explicit hateful
Implicit hateful	1341	1091
Explicit hateful	368	4511

44.86% for implicit hateful and 7.54% for explicit hateful in layer B. However, the misclassification for *non-hateful* is 8.37%. MuRIL-MTL trained with Hindi, Urdu, and Bangla is performing best for layer C by giving exact match and hamming loss of 62.10% and 10.68%.

B. QUALITATIVE ANALYSIS

This section highlights some of the Type 1 errors (False-Positives) and Type 2 errors (False-Negatives) from HHSD. We showed three cases for each where the model erroneously misclassified the post. The possible human explanation for the wrong prediction is also given.

Type 1 Error: False Positives (*Non−hateful* → *Hateful*)

1) ANIMAL REFERENCING

1) वैसे सूअर का मीट क्या रेट चल रहा है आजकल
Transliteration: waise suwar ka meat kya rate chal raha hai?
Translation: By the way, What is the rate of pork?

2) यह वह वाहन है जिससे कुत्ते को मारा गया है
Transliteration: yeh wah wahan hai jisse kutte ko mara gaya hai.
Translation: This is the same vehicle used to kill the dog.

Human Explainability: The underlined phrase in both the sentences are referring to an animal. Because the language is structured in a way that suggests an implied attack on people, the algorithm incorrectly predicts it to be hate speech.

2) PRESENCE OF EXPLICIT WORD

1) बकचोदी कोरोना के शुरूआती लक्षण दिखने पर मैंने टेस्ट करवाया
Transliteration: Bakchodi corona ke suruaati lakshan dikhne par maine test karaya.
Translation I went for a test after seeing the initial symptoms of the f***ing corona.

2) आराम हराम है शिखा
Transliteration: Aaram haram hai shikha.
Translation: Relaxing is Ba****d shikha!

Human Explainability: In both of the highlighted bigrams, an abusive token is combined with a non-abusive token. The coronavirus is the target of the attack in the first post, whereas a motivational quotation is the subject of the harsh word in the second tweet. However, the model was unable to convey the post's sentiment.

3) INDIRECT REFERENCE

1) एक मेंढक निकलकर बोला पानी में आ तेरी उदासी उतारू साले
Trnasliteration: Ek medhak nikal kar bola paani me aa teri udasi utaru saale.
Translation:A frog came out and said, come in the water, I will remove your sadness.U B****rd.

2) ये कैसी कुत्ते जैसी बिल्ली है
Transliteration: ye kaisi kutte jaisi billi hai.
Translation:What kind of dog-like cat is this?

Human Explainability: In the first tweet, a frog verbally assaults a human by speaking in a negative manner and using vulgar language. Additionally, it appears that a person is being referenced twice in the second post by referring to an animal.

Type 2 Error: False Negatives(*Hateful* → *Non−hateful*)

4) OBFUSCATION OF SLANG WORD

1) इसका हमेशा त्रुटिया कटा है और कटता रहेगा
Transliteration: Iska hamesha trutiya kata hai aur kattha rahega.
Translation: This person will be fooled always.

2) तर्क :- हुतिया है भाजपा व उसके नेता
Transliteration: Tarka: Hutia hai Bhajpa vah uske neta.
Translation: Argument: BJP and its leader are B****rd.

Human Explainability: Users obfuscate the slang term to trick the model and succeed in posting their sentiment. The underlined words in the two posts were used in an insulting manner towards both a person and a group, yet the model did not pick up on the seriousness of the offence.

5) PRESENCE OF SARCASM

1) कुत्ते और बिल्ली एक साथ बिरयानी खा रहे, वाह मोदी आप ने क्या करिश्मा किया है
Transliteration: Kutte aur billi ek sath biryani khaa rahae hai, waah modi aap ne kya karishma kiya hai.
Translation: Dogs and cats are eating Biryani together. Wow Modi, you did a miracle.

2) शुरुवात भाई बहन के रिश्ते से करो, बाद में मौलाना की मर्जी
Transliterate: suruaat bhai bahan ke rishte se karo, baad me maulana kii marjii.
Translation: Start with the relation of brother and sister, later it is the wish of Maulana.

Human Explainability Both posts make an implicit reference to attack with the highlighted term. To fully comprehend the true meaning underlying these posts, contextual information is necessary. The covertness present in the posts is not captured by the model.

6) NAME CALLING

1) वाह पलटूराम वाह आखिर संगति का असर है
Transliteration: Waah Palturaam waah aakhir sangati kaa asar hai.
Translation: Wah Palturam, wow, after all it is the effect of company.

2) लुटेरी दुल्हन सुन कर एंटोनियो माइनो की याद आ जाती है
Transliteration: Luteri dulhan sun kar antonio maino kii yaad aa jati hai.
Translation:Hearing the robber bride, one remembers Antonio Maino.

Human Explainability The phrase in italics is a name-calling reference to a specific individual. Both pieces metaphorically criticise well-known political figures. This is not captured by the model, leading to incorrect classification.

TABLE 14. Bootstrapping test.

Model	#Total posts	#Sample taken	p-value
M1-M2	14783	8870	≤ 0.01
M1-M2	7311	4386	≤ 0.03
M3-M4	14783	8870	≤ 0.04

C. STATISTICAL SIGNIFICANCE TEST

A bootstrap sample test is used to assess whether the difference between the two models is statistically significant (at $p \leq 0.05$). By selecting three confusion matrices out of a possible five at a time, the test determines if the better system is the same as the better system across the entire data set. The outcome (p-) value of the bootstrap test is the proportion of samples where the winner differs from the entire data set. Table 14 displays the results of the statistical significance test conducted on each of the best three pairs of models for both datasets. We measured the score between $M - BERT_H$(M1), $M - BERT_{HBU}$(M2), $MuRIL_H$(M3), and $MuRIL_{HBU}$(M4).

VII. CONCLUSION AND FUTURE WORK

In this study, we developed a benchmark corpus for hate speech identification by crowdsourcing the manual annotation of roughly 15K tweets using a novel four-layer annotation schema. Using keywords and issues related to politics, religion, racism, and sexism, the tweets were crawled. To achieve promising results in terms of accuracy and weighted-f1 score, we undertook an in-depth examination of various experiments carried out on novel-created Hindi data employing deep learning and transformers-based architectures in single-task learning and multi-task learning frameworks. By utilising numerous data from the same domain in different languages, the suggested technique is a long-term approach that typically enhances the adoption of BERT-based models with fewer stringent requirements in terms of annotated training data. Explainability in AI is very important when dealing with sensitive issues which can negatively impact society. There are considerable efforts being made to make sure that AI-based technology does not suffer from any kind of bias introduced by the training data or the training procedure. As a future work, we plan on enriching the dataset with more boosted data, since, as we showed, they carry most of the valuable information about inappropriate speech. Since a lot of tweets require contextual information, localised knowledge graphs can be created for this by collecting intra-user and inter-user tweets to obtain valuable features. The contextual knowledge can easily be verified against this knowledge base.

ACKNOWLEDGMENT

Prashant Kapil acknowledges the University Grant Commission (UGC) of the Government of India for UGC NET-JRF/SRF fellowship.

REFERENCES

[1] J. Nockleyby, "Hate speech in encyclopedia of the American constitution," *Electron. J. Academic Special Librarianship*, 2000.

[2] T. Davidson, D. Warmsley, M. Macy, and I. Weber, "Automated hate speech detection and the problem of offensive language," in *Proc. Int. AAAI Conf. Web Social Media*, 2017, vol. 11, no. 1, pp. 512–515.

[3] T. Mandl, S. Modha, P. Majumder, D. Patel, M. Dave, C. Mandlia, and A. Patel, "Overview of the HASOC track at FIRE 2019: Hate speech and offensive content identification in indo-European languages," in *Proc. 11th Forum Inf. Retr. Eval.*, Dec. 2019, pp. 14–17.

[4] T. Mandl, S. Modha, A. Kumar M, and B. R. Chakravarthi, "Overview of the HASOC track at FIRE 2020: Hate speech and offensive language identification in Tamil, Malayalam, Hindi, English and German," in *Proc. Forum Inf. Retr. Eval.*, Dec. 2020, pp. 29–32.

[5] V. K. Jha, P. Hrudya, P. N. Vinu, V. Vijayan, and P. Prabahara, "DHOT-repository and classification of offensive tweets in the Hindi language," *Proc. Comput. Sci.*, vol. 171, pp. 2324–2333, Jan. 2020.

[6] M. M. Rahman, D. Balakrishnan, D. Murthy, M. Kutlu, and M. Lease, "An information retrieval approach to building datasets for hate speech detection," 2021, *arXiv:2106.09775*.

[7] T. Grndahl, L. Pajola, M. Juuti, M. Conti, and N. Asokan, "All you need is 'love' evading hate speech detection," in *Proc. 11th ACM Workshop Artif. Intell. Secur.*, 2018, pp. 2–12.

[8] H. Rizwan, M. H. Shakeel, and A. Karim, "Hate-speech and offensive language detection in Roman Urdu," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2020, pp. 2512–2522.

[9] O. de Gibert, N. Perez, A. García-Pablos, and M. Cuadros, "Hate speech dataset from a white supremacy forum," 2018, *arXiv:1809.04444*.

[10] B. Kennedy et al., "Introducing the gab hate corpus: Defining and applying hate-based rhetoric to social media posts at scale," *Lang. Resour. Eval.*, vol. 56, no. 1, pp. 79–108, Mar. 2022.

[11] P. Li, "Achieving hate speech detection in a low resource setting," Ph.D. dissertation, Utah State Univ., Logan, Utah, 2021.

[12] R. Kumar, A. N. Reganti, A. Bhatia, and T. Maheshwari, "Aggression-annotated corpus of Hindi-English code-mixed data," 2018, *arXiv:1803.09402*.

[13] S. Bhattacharya, S. Singh, R. Kumar, A. Bansal, A. Bhagat, Y. Dawer, B. Lahiri, and A. K. Ojha, "Developing a multilingual annotated corpus of misogyny and aggression," 2020, *arXiv:2003.07428*.

[14] A. Bohra, D. Vijay, V. Singh, S. S. Akhtar, and M. Shrivastava, "A dataset of Hindi-English code-mixed social media text for hate speech detection," in *Proc. 2nd Workshop Comput. Modeling People's Opinions, Personality, Emotions Social Media*, 2018, pp. 36–41.

[15] T. Y. S. S. Santosh and K. V. S. Aravind, "Hate speech detection in Hindi-English code-mixed social media text," in *Proc. ACM India Joint Int. Conf. Data Sci. Manage. Data*, Jan. 2019, pp. 310–313.

[16] A. Velankar, H. Patil, A. Gore, S. Salunke, and R. Joshi, "Hate and offensive speech detection in Hindi and Marathi," 2021, *arXiv:2110.12200*.

[17] T. Mandl, S. Modha, G. K. Shahi, H. Madhu, S. Satapara, P. Majumder, J. Schaefer, T. Ranasinghe, M. Zampieri, D. Nandini, and A. K. Jaiswal, "Overview of the HASOC subtrack at FIRE 2021: Hate speech and offensive content identification in English and Indo-Aryan languages," 2021, *arXiv:2112.09301*.

[18] S. Chopra, R. Sawhney, P. Mathur, and R. R. Shah, "Hindi-English hate speech detection: Author profiling, debiasing, and practical perspectives," in *Proc. AAAI Conf. Artif. Intell.*, 2020, vol. 34, no. 1, pp. 386–393.

[19] P. Mathur, R. Sawhney, M. Ayyar, and R. Shah, "Did you offend me? Classification of offensive tweets in Hinglish language," in *Proc. 2nd Workshop Abusive Lang.*, 2018, pp. 138–148.

[20] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *Trans. Assoc. Comput. Linguistics*, vol. 5, pp. 135–146, Dec. 2017.

[21] R. Kumar, A. K. Ojha, S. Malmasi, and M. Zampieri, "Benchmarking aggression identification in social media," in *Proc. 1st Workshop Trolling, Aggression Cyberbullying*, 2018, pp. 1–11.

[22] M. Das, P. Saha, B. Mathew, and A. Mukherjee, "HateCheckHIn: Evaluating Hindi hate speech detection models," 2022, *arXiv:2205.00328*.

[23] V. G. Rahul, V. Sehra, and Y. R. Vardhan, "Hindi-English code mixed hate speech detection using character level embeddings," in *Proc. 5th Int. Conf. Comput. Methodologies Commun. (ICCMC)*, Apr. 2021, pp. 1112–1118.

[24] K. Sreelakshmi, B. Premjith, and K. P. Soman, "Detection of hate speech text in Hindi-English code-mixed data," *Proc. Comput. Sci.* vol. 171, pp. 737–744, Jan. 2020.

[25] N. Vashistha and A. Zubiaga, "Online multilingual hate speech detection: Experimenting with Hindi and English social media," *Information*, vol. 12, no. 1, p. 5, Dec. 2020.

[26] M. Das, S. Banerjee, and A. Mukherjee, "Data bootstrapping approaches to improve low resource abusive language detection for indic languages," in *Proc. 33rd ACM Conf. Hypertext Social Media*, Jun. 2022, pp. 32–42.

[27] S. Biradar, S. Saumya, and A. Chauhan, "Fighting hate speech from bilingual Hinglish speaker's perspective, a transformer- and translation-based approach," *Social Netw. Anal. Mining*, vol. 12, no. 1, p. 87, Dec. 2022.

[28] V. Gupta, S. Roychowdhury, M. Das, S. Banerjee, P. Saha, B. Mathew, and A. Mukherjee, "Multilingual abusive comment detection at scale for indic languages," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 35, 2022, pp. 26176–26191.

[29] O. Sharif and M. M. Hoque, "Tackling cyber-aggression: Identification and fine-grained categorization of aggressive texts on social media using weighted ensemble of transformers," *Neurocomputing*, vol. 490, pp. 462–481, Jun. 2022.

[30] N. Romim, M. Ahmed, H. Talukder, and M. S. Islam, "Hate speech detection in the Bengali language: A dataset and its baseline evaluation," in *Proc. Int. Joint Conf. Adv. Comput. Intell.* Singapore: Springer, 2021, pp. 457–468.

[31] U. Azam, H. Rizwan, and A. Karim, "Exploring data augmentation strategies for hate speech detection in Roman Urdu," in *Proc. 13th Lang. Resour. Eval. Conf.*, 2022, pp. 4523–4531.

[32] Y. Zhang and Q. Yang, "A survey on multi-task learning," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 12, pp. 5586–5609, Dec. 2022.

[33] P. Kapil, A. Ekbal, and D. Das, "Investigating deep learning approaches for hate speech detection in social media," 2020, *arXiv:2005.14690*.

[34] F. M. Plaza-Del-Arco, M. D. Molina-González, L. A. Ureña-López, and M. T. Martín-Valdivia, "A multi-task learning approach to hate speech detection leveraging sentiment analysis," *IEEE Access*, vol. 9, pp. 112478–112489, 2021.

[35] C. Breazzano, D. Croce, and R. Basili, "MT-GAN-BERT: Multi-task and generative adversarial learning for sustainable language processing," in *Proc. NL4AI*, 2021, pp. 1–6.

[36] A. Founta, C. Djouvas, D. Chatzakou, I. Leontiadis, J. Blackburn, G. Stringhini, A. Vakali, M. Sirivianos, and N. Kourtellis, "Large scale crowdsourcing and characterization of Twitter abusive behavior," in *Proc. Int. AAAI Conf. Web Social Media*, 2018, vol. 12, no. 1, pp. 1–12.

[37] J. L. Fleiss, "Measuring nominal scale agreement among many raters," *Psychol. Bull.*, vol. 76, no. 5, pp. 378–382, Nov. 1971.

[38] R. Kumar, A. K. Ojha, S. Malmasi, and M. Zampieri, "Evaluating aggression identification in social media," in *Proc. 2nd Workshop Trolling, Aggression Cyberbullying*, 2020, pp. 1–5.

[39] M. Bhardwaj, M. S. Akhtar, A. Ekbal, A. Das, and T. Chakraborty, "Hostility detection dataset in Hindi," 2020, *arXiv:2011.03588*.

[40] M. M. Khan, K. Shahzad, and M. K. Malik, "Hate speech detection in Roman Urdu," *ACM Trans. Asian Low-Resource Lang. Inf. Process.*, vol. 20, no. 1, pp. 1–19, 2021.

[41] Z. Waseem and D. Hovy, "Hateful symbols or hateful people? Predictive features for hate speech detection on Twitter," in *Proc. NAACL Student Res. Workshop*, 2016, pp. 88–93.

[42] M. Zampieri, S. Malmasi, P. Nakov, S. Rosenthal, N. Farra, and R. Kumar, "Predicting the type and target of offensive posts in social media," 2019, *arXiv:1902.09666*.

[43] J. Golbeck et al., "A large labeled corpus for online harassment research," in *Proc. ACM Web Sci. Conf.*, Jun. 2017, pp. 229–233.

[44] V. Basile, C. Bosco, E. Fersini, D. Nozza, V. Patti, F. M. R. Pardo, P. Rosso, and M. Sanguinetti, "SemEval-2019 task 5: Multilingual detection of hate speech against immigrants and women in Twitter," in *Proc. 13th Int. Workshop Semantic Eval.*, 2019, pp. 54–63.

[45] I. A. Bhat, V. Mujadia, A. Tammewar, R. A. Bhat, and M. Shrivastava, "IIIT-H system submission for FIRE2014 shared task on transliterated search," in *Proc. Forum Inf. Retr. Eval.*, 2015, pp. 48–53.

[46] A. Celikyilmaz, A. Bosselut, X. He, and Y. Choi, "Deep communicating agents for abstractive summarization," 2018, *arXiv:1803.10357*.

[47] Y. Chen, "Convolutional neural network for sentence classification," M.S. thesis, Univ. Waterloo, Waterloo, ON, Canada, 2015.

[48] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2018, *arXiv:1810.04805*.

[49] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, and V. Stoyanov, "Unsupervised cross-lingual representation learning at scale," 2019, *arXiv:1911.02116*.

[50] T. Kudo and J. Richardson, "SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing," 2018, *arXiv:1808.06226*.
[51] T. Kudo, "Subword regularization: Improving neural network translation models with multiple subword candidates," 2018, *arXiv:1804.10959*.
[52] G. Lample and A. Conneau, "Cross-lingual language model pretraining," 2019, *arXiv:1901.07291*.
[53] F. Feng, Y. Yang, D. Cer, N. Arivazhagan, and W. Wang, "Language-agnostic BERT sentence embedding," 2020, *arXiv:2007.01852*.
[54] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *J. Mach. Learn. Res.*, vol. 21, no. 1, pp. 5485–5551, 2020.
[55] J. Uszkoreit, J. Ponte, A. Popat, and M. Dubiner, "Large scale parallel document mining for machine translation," Tech. Rep., 2010.
[56] S. Khanuja, D. Bansal, S. Mehtani, S. Khosla, A. Dey, B. Gopalan, D. K. Margam, P. Aggarwal, R. Teja Nagipogu, S. Dave, S. Gupta, S. C. B. Gali, V. Subramanian, and P. Talukdar, "MuRIL: Multilingual representations for Indian languages," 2021, *arXiv:2103.10730*.
[57] D. Kakwani, A. Kunchukuttan, S. Golla, A. Bhattacharyya, M. M. Khapra, and P. Kumar, "IndicNLPSuite: Monolingual corpora, evaluation benchmarks and pre-trained multilingual language models for Indian languages," in *Proc. Findings Assoc. Comput. Linguistics, EMNLP*, 2020, pp. 4948–4961.
[58] X. Liu, P. He, W. Chen, and J. Gao, "Multi-task deep neural networks for natural language understanding," 2019, *arXiv:1901.11504*.
[59] F. Chollet. (2015). *Keras*. [Online]. Available: https://keras.io
[60] M. Abadi et al., "TensorFlow: Large-scale machine learning on heterogeneous distributed systems," 2016, *arXiv:1603.04467*.
[61] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.
[62] R. Venkatesan and M. J. Er, "Multi-label classification method based on extreme learning machines," in *Proc. 13th Int. Conf. Control Autom. Robot. Vis. (ICARCV)*, Dec. 2014, pp. 619–624.
[63] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional LSTM networks," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, 2005, pp. 2047–2052.

PRASHANT KAPIL received the B.Tech. degree from MAKAUT, West Bengal, and the M.E. degree in information technology from Jadavpur University, West Bengal. He is currently a Senior Research Fellow with the Department of Computer Science and Engineering, IIT Patna. He has published papers in various peer-reviewed conferences and journals. His research interest includes hate speech detection and prevention for English and Hindi language. He was a recipient of the University Grant Commission (UGC)-NET-JRF/SRF Fellowship by the Government of India.

GITANJALI KUMARI is currently a Junior Research Fellow with the Department of Computer Science and Engineering, IIT Patna. She has published papers in various peer-reviewed conferences and journals. Her research interest includes hate speech detection and prevention for English and Hindi language.

ASIF EKBAL (Senior Member, IEEE) is currently an Associate Professor with the Department of Computer Science and Engineering, IIT Patna. He has been doing research in natural language processing (NLP), information extraction, text mining, and machine learning (ML), for the last 16 years, and has made significant contributions in these areas. He has authored around 280 papers in top-tier journals and conferences. He has been involved in several sponsored research projects, funded by private and the government agencies. He is an awardee of the Best Innovative Project Award from the Indian National Academy of Engineering, Government of India, the JSPS Invitation Fellowship from the Government of Japan, and the Visvesvaraya Young Faculty Research Fellowship Award from the Government of India.

SANTANU PAL is currently a Lead Scientist with the Wipro AI Laboratory. His research interests include natural language processing, social AI, machine translation, automatic post-editing, question answering and question generation, human–computer interaction, multi-modal interface, and speech translation.

ARINDAM CHATTERJEE received the M.Tech. degree in computer science from IIT Bombay. He is currently pursuing the Ph.D. degree with a focus on code-mixed or code-switched languages. He is a Principal Data Scientist with Wipro Research and Development, Lab45. He has 11 years of industry experience, with more than ten granted patents and top-tier conference publications. His niche lies in pioneering innovative AI solutions and leading collaborative projects with academia. His research interests include codemixing, generative AI, optimized LLMs, large multimodal models, and hate speech detection.

B. N. VINUTHA (Member, IEEE) is currently the AI Research Head of the Wipro Lab45, working at the intersection of latest research and challenging customer problems, with a good blend of technical and managerial skills. Passionate about AI, believe ethical and transparent AI holds key to its acceptance in society.