IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 22 August 2024, accepted 29 August 2024, date of publication 2 September 2024, date of current version 10 September 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3452987

RESEARCH ARTICLE

Enhancing Multilingual Hate Speech Detection: From Language-Specific Insights to Cross-Linguistic Integration

EHTESHAM HASHMI[1], SULE YILDIRIM YAYILGAN[1], IBRAHIM A. HAMEED[1], (Senior Member, IEEE), MUHAMMAD MUDASSAR YAMIN[1], MOHIB ULLAH[2], AND MOHAMED ABOMHARA[1]

[1]Department of ICT and Natural Sciences (IIR), Norwegian University of Science and Technology (NTNU), Ålesund, 6009 Møre og Romsdal, Norway
[2]Department of Information Security and Communication Technology (IIK), Norwegian University of Science andTechnology (NTNU), 2815 Gjøvik, Norway

Corresponding author: Ehtesham Hashmi (hashmi.ehtesham@ntnu.no)

This work was supported by the SOCYTI Project through the Research Council of Norway as a Researcher Project for Technological Convergence related to Enabling Technologies under Grant 331736.

ABSTRACT The rise of social media has enabled individuals with biased perspectives to spread hate speech, directing it toward individuals based on characteristics such as race, gender, religion, or sexual orientation. Constructive interactions in varied communities can greatly enhance self-esteem, yet it is vital to consider that adverse comments may affect individuals' social standing and emotional health. The crucial task of detecting and addressing this type of content is imperative for reducing its negative effects on communities and individuals alike. The rising occurrence highlights the urgency for enhanced methods and robust regulations on digital platforms to protect humans from such prejudicial and damaging conduct. Hate speech typically appears as a deliberate hostile action aimed at a particular group, often with the intent to demean or isolate them based on various facets of their identity. Research on hate speech predominantly targets resource-aware languages like English, German, and Chinese. Conversely, resource-limited languages, including European languages such as Italian, Spanish, and Portuguese, alongside Asian languages like Roman Urdu, Korean, and Indonesian, present obstacles. These challenges arise from a lack of linguistic resources, making the extraction of information a more strenuous task. This study is focused on the detection and improvement of multilingual hate speech detection across 13 different languages. To conduct a thorough analysis, we carried out a series of experiments that ranged from classical machine learning techniques and mainstream deep learning approaches to recent transformer-based methods. Through hyperparameter tuning, optimization techniques, and generative configurations, we achieved robust and generalized performance capable of effectively identifying hate speech across various dialects. Specifically, we achieved a notable enhancement in detection performance, with precision and recall metrics exceeding baseline models by up to 10% across several lesser-studied languages. Additionally, our work extends the capabilities of explainable AI within this context, offering deeper insights into model decisions, which is crucial for regulatory and ethical considerations in AI deployment. Our study presents substantial performance improvements across various datasets and languages through meticulous comparisons. For example, our model significantly outperformed existing benchmarks: it achieved F1-scores of 0.90 in German (GermEval-2018), up from the baseline score of 0.72, and 0.93 in German (GermEval-2021), a substantial increase from 0.58. Additionally, it scored

The associate editor coordinating the review of this manuscript and approving it for publication was Wei-Yen Hsu.

© 2024 The Authors. This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License.
For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

0.95 in Roman Urdu HS, surpassing the previous peak of 0.91. Furthermore, for mixed-language datasets such as Italian and English (AMI 2018), our accuracy rose dramatically from 0.59 to 0.96. These outcomes emphasize the robustness and versatility of our model, establishing a new standard for hate speech detection systems across diverse linguistic settings.

INDEX TERMS Hate speech, word embedding, machine learning, deep learning, transformers, natural language processing, explainable AI.

I. INTRODUCTION

With the progression of digital technology, the age of social computing has notably transformed how individuals interact with one another, particularly evident in the widespread utilization of social media platforms and online chat forums [1], [2]. These platforms are playing a significant role in shaping the world into an increasingly globalized scenario, where social media users from various regions can easily exchange information [3]. Hate speech (HS), which emerges from the clashes between different groups within and across societies, is a phenomenon that can spread rapidly on social media due to its intersection with various tensions [4], [5]. The definition of HS, which is sometimes ambiguous and has different meanings in different places and cultures, makes it difficult to identify and regulate, especially in the digital era. It appears in various forms, as identified by researchers, including cyberbullying, flaming, the use of profanity, abusive language, expressions of toxicity, and acts of discrimination [6], [7], [8]. These types of controversial materials spark heated arguments, and the resulting anger can escalate to violent crimes or physical attacks [9].

Preserving a secure and welcoming online space can become particularly challenging due to the intensification of such behaviors and these issues are frequently worsened by anonymity and the absence of real-world consequences. Prompt and precise methods for identifying and addressing these issues demand urgent and careful attention due to their rapid proliferation and evolving nature. To filter out the proliferation of hate content, researchers are using the capabilities of Artificial Intelligence (AI) methods to develop two different types of language-specific algorithms [10]. Most studies have predominantly concentrated on resource-aware languages like English [11], [12]. This prioritization of resource-rich languages has led to a notable gap in HS-related research, especially concerning languages with limited resources, such as Italian, Korean, Portuguese, Turkish, Roman Urdu/Hindi, and Arabic. This study aims to explore multilingual HS detection across 13 languages, including both resource-aware and resource-limited languages. This exploration will be conducted through a series of experiments utilizing ML, DL, and Transformer-based models. The experiments will involve hyperparameter tuning, and optimization techniques, alongside the integration of Explainable AI (XAI) modeling, aimed at providing detailed insights into the text data. The following section provides a summary of the contributions of this research, followed by how the rest of the paper is organized.

A. WORK CONTRIBUTIONS

1) This study pioneers the integration of a wide array of methodologies for HS detection across a multilingual dataset spanning 13 languages, combining pre-trained embeddings, language-specific transformer models, Machine Learning (ML), and Deep Learning (DL) classifiers for comprehensive linguistic coverage. We advance the field with the introduction of robust cross-language evaluation techniques, including a systematic n-1 analysis and incremental learning strategies that enhance model generalizability, adaptability, and scalability across different linguistic contexts.
2) We significantly enhance HS detection systems by employing language-specific transformers such as AraBERT, GermanBERT, TurkishBERT, ENRIE, ItalianBERT, CAMEMBERT, XLNet, RoBERTa, MacBERT, FlauBERT, AlbertSpanish, and multilingual transformer-based models like mBERT, ELECTRA, mBART, and FLAN-T5. These are optimized through regularization techniques, hyperparameter tuning, and generative configurations, demonstrating the superiority of transformer-based architectures in achieving model robustness and scalability.
3) The implementation of prompt-based fine-tuning methods, including few-shot and full fine-tuning with generative configurations, leverages the capabilities of transformer-based models for HS detection tasks, marking a significant contribution to the field by adapting these advanced architectures to specific requirements.
4) A unified multilingual HS classification is achieved by utilizing transformer-based models and supervised FastText in the final evaluation of our research, showcasing a holistic approach towards multilingual HS detection and classification.
5) Our findings underscore the exceptional performance of the Support Vector Machine (SVM) in conjunction with FastText across all 13 languages, based on metrics like accuracy, F1-score, precision, and recall, enhanced by parameter tuning, regularization, and quantization. To provide interpretability, we incorporate Local Interpretable Model-agnostic Explanations (LIME), offering deep insights into the decision-making processes of our models, thereby contributing to the transparency and understanding of HS detection mechanisms.

B. STRUCTURE OF THE PAPER

The rest of the paper is structured as follows: Section (II) discusses the existing research work on HS. Section (III) explains the proposed work methodology. Section (IV) focuses on the results and discussions. Section (V) is based on the comparison of our results with the state-of-the-art. Section (VI) is related to the interpretability modeling with LIME. Section (VII) presents the conclusion and future work.

II. RELATED WORK

The rise in social media users necessitates the development of sophisticated HS detection systems to sift through and eliminate unethical and hateful content. This growing demand highlights the critical need for innovative solutions in moderating online discourse. Recent breakthroughs in AI and Natural Language Processing (NLP) have amplified the significance of the HS detection methodologies [13], [14]. These novel approaches have enhanced our understanding of HS, its implications, and monitoring throughout social networks and public discourse. However, the majority of this research has focused on languages that are resource-aware, such as English. The focus on HS detection has resulted in the underrepresentation of languages that are resource-limited, including Korean, Portuguese, Italian, and South Asian languages [15].

A. MACHINE LEARNING BASED APPROACHES

Akuma et al. [16] conducted a study where they analyzed a dataset of HS and offensive language from Kaggle[1] using four Machine Learning (ML) algorithms: K-Nearest Neighbour (KNN), Decision Tree (DT), Logistic Regression (LR), and Naive Bayes (NB), along with two different word embeddings, Bag of Words (BoW) and Term Frequency Inverse Document Frequency (TF-IDF). Their findings showed that DT combined with TF-IDF achieved the highest accuracy score of 0.92 among the models they tested. Elzayady et al. [17] designed a novel way to identify HS within Arabic dialects. They employed a dual-phase strategy that combines both standard ML-based approaches and Deep Learning (DL) methodologies, with an added focus on personality traits. They initially utilized the AraPersonality dataset to explore the relationship between personality factors and HS through correlation analysis. Subsequently, they employed the TF-IDF technique to extract the featured and then inputted them into a range of ML-based classifiers including Random Forest (RF), LR, DT, Support Vector Machine (SVM), and Extreme Gradient Boosting (XGBoost).

The study by Mittal and Singh [18] exemplifies successful HS identification in English using an ML-based framework, where the combination of XGBoost and Count Vectorizer (CV) for feature extraction, along with the integration of the LIME framework for explanation, led to remarkable performance with an F1-score of 0.94. Agarwal et al. [19] proposed automatic HS detection using parallelized ensemble learning-based models. In this work, parallel variants of bagging, A-stacking, and random sub-space algorithms are constructed and compared to serial counterparts. The comparison analysis is carried out on a set of standard high-dimensional datasets designed expressly for the detection of HS. These statistics include examples of HS transmission in response to major events like as the COVID-19 pandemic, the 2020 US presidential election, and the 2021 farmers' protest in India. Toktarova et al. [20] identified HS across three distinct datasets focused on HS and offensive language, cyberbullying, and Twitter-based HS. The team applied Word2Vec and GloVe for vectorization. Through a comprehensive set of experiments utilizing both ML and DL techniques, their proposed methodology achieved an F1-score of 0.85 across all datasets. These significant advancements in state-of-the-art methods highlight the effectiveness of combining advanced vectorization methods with sophisticated ML-based modelings in addressing the complexities of HS detection. Their work demonstrates a notable advancement in the automated identification and analysis of online HS, paving the way for more refined and accurate detection systems in the future.

B. DEEP LEARNING AND TRANSFORMERS BASED APPROACHES

DL and transformer-based models have transformed the state-of-the-art in textual analysis. These advanced techniques, such as Recurrent Neural Networks (RNNs), Convolutional Neural Networks (CNNs), and Bidirectional Encoder Representations from Transformers (BERT), excel at capturing complex patterns and semantic relationships due to their ability to process large amounts of text. The Transformer is an NLP framework designed to handle sequence-to-sequence tasks using the self-attention mechanism while allowing long-range dependencies. Saleh et al. [21] applied BiLSTM and BERT, a transformer-based model, for binary HS detection. The study used three widely recognized datasets: [22], [23], and [24]. They evaluated three embedding types: domain-specific, Word2Vec [25], and GloVe [26] for better word representation. Recent advancements in AI have seen the development of Large Language Models (LLMs) like GPT-3 and BERT, which possess the capability to generate vast amounts of synthetic textual data [27], [28]. By leveraging these models, researchers can create highly diverse and representative datasets in multiple languages, particularly beneficial for languages that traditionally lack sufficient real-world data for training purposes. A. Bezerra de Oliveira et al. [29] introduced a novel approach for detecting HS across multiple languages using LLMS and Cross-Lingual Learning (CLL). Employing a combination of supervised and unsupervised learning techniques, the method integrates GPT-3 with enhanced cross-lingual adaptation to analyze hate speech in English, Italian, German, and other languages. This approach leverages diverse corpora to refine

[1] https://www.kaggle.com/datasets/mrmorj/hate-speech-and-offensive-language-dataset

model generalizability, achieving remarkable improvements in precision and recall metrics. GPT-3, combined with Joint Learning (JL) and Cascade Learning (CL) strategies, showcased an F1-score of 96.58% for English, demonstrating the potential of CLL in enhancing the robustness of LLMs against complex and diverse datasets. This work highlights the versatility of LLMs in handling nuanced linguistic variations in HS detection tasks. Firmino et al. [30] developed an innovative method to enhance HS detection by leveraging Cross-Lingual Learning (CLL) across Italian, English, and Portuguese, utilizing Pre-Trained Language Models (PTLMs). They engaged with English and Italian corpora as source languages, while the OffComBr-2 corpus served as the target language dataset in Portuguese. This cross-lingual strategy employed PTLMs such as BERT and XLM-RoBERTa, achieving an F1 measure of 92%. In their study, Khan et al. [31] introduced a framework for detecting violence incitation in Urdu tweets, employing a 1D-CNN and semantic word embeddings. Using a newly annotated corpus of 4808 tweets, the approach compares 1D-CNN with traditional machine learning and transformer models. The 1D-CNN model, integrated with word unigram features, demonstrates superior performance, achieving 89.84% accuracy and 89.80% macro F1-score, outperforming all other evaluated models. This work underscores the effectiveness of combining CNN with contextualized language representations for violence detection in social media content.

In their studies, Svetasheva and Lee [32] addressed the issue of HS detection by using the capabilities of LLMs such as ChatGPT-4 to develop synthetic data. They concentrated on sectors with minimal data, such as online gaming communities. Their study included a two-phase approach that combined human and LLM annotations on Dota 2 game match data, which was followed by the improvement of these datasets using machine-generated facts. This technique showed LLMs' ability to improve model performance and dataset quality, providing a unique strategy for resolving the issues of dataset inequalities in the automatic detection of HS. García-Díaz et al. [33] highlighted the issues of detecting HS on social media, focusing on Spanish-language instances of xenophobia, misogyny, and homophobia, with the goal of improving HS detection performance in Spanish by combining linguistic knowledge and transformer-based model capabilities. Evaluation of the results was performed by using several Spanish BERT models, including BERTIN [34], BETO [35], Spanish RoBERTa [36], and mBERT [37], and found BETO to be the most effective. Nagar et al. [38] introduced an innovative method to detect HS by using two publicly available datasets [39] and [40]. Their developed framework, known as Variational Graph Auto-Encoder (VGAC), capitalizes on multi-modal data. This approach integrates two key aspects: the textual content within tweets and the social network framework of the tweeting users.

Multiple agents interact and learn from both real and synthetic data, potentially leading to more sophisticated and contextually aware models. However, this raises significant ethical considerations, as the reliance on synthetic data could influence the agents' behavior and decision-making processes in unpredictable ways [41]. Ziems et al. [42] explored the transformative potential of LLMs within Computational Social Science (CSS). They assessed the capabilities of LLMs to augment human annotation and enhance content analysis through zero-shot performance evaluations across diverse CSS tasks. The study highlights that while LLMs do not consistently surpass finely tuned models, they offer substantial contributions to annotating and generating content, especially in task-specific contexts. Mehmood [3] introduced "Passion-Net," an advanced DL predictor for detecting HS in Roman Urdu (RU) text, offering a significant performance improvement over existing models. It employs advanced language modeling to extract semantic patterns and utilizes an attention-based classifier for precise hate content identification. The model demonstrates superior accuracy, precision, and F1-scores on both coarse-grained and fine-grained datasets, outperforming state-of-the-art models by substantial margins. Additionally, it incorporates interpretability features, providing insights into the contribution of specific words toward classification decisions. Focusing on the resource-limited RU language, Khan et al. [43] developed a deep neural network that leverages CNN for feature extraction and LSTM to understand long-term textual dependencies, using embeddings from Word2Vec CBoW, GLOVE, and FastText. In a comparable manner Nagra et al. [44] developed sentiment analysis for RU using a Faster Recurrent CNN (FR-CNN) on the RUSA-19 dataset, performing binary and ternary classifications to distinguish positive, negative, and neutral classes. Chen et al. [45] created a hybrid cyberbullying detection model that combines XLNet with deep Bi-LSTM and incorporates Enhanced Representation via Knowledge Integration (ERNIE) [46] for analyzing Chinese social media content. Leveraging the strengths of these models, the study enhanced its accuracy in identifying cyberbullying cases. To resolve data imbalances, the strategy involved a full relabeling and enlargement of the COLDATASET, which included actual cyberbullying remarks. This new methodology demonstrated the efficacy of combining advanced NLP models to improve the understanding and processing of complex linguistic subtleties in cyberbullying detection. In their research, Mahajan et al. [47] presented an innovative ensemble DL-based model for detecting multilingual HS and cyberbullying across online social media. Utilizing GloVe embeddings with a mix of BiLSTM, BiGRU, CNN, and LSTM through a bagging-stacking ensemble method, it achieved promising results on nine datasets in English, Bengali, Indonesian, Italian, and Spanish.

The societal implications of using LLM-generated synthetic data for HS detection are profound. On one hand,

this technique promises improved detection accuracy and fairness in moderation across diverse languages and cultures, contributing to more inclusive digital environments. On the other hand, the deployment of these models must be handled with caution to prevent the propagation of biases or the reinforcement of harmful stereotypes. It is crucial that such technologies are developed and implemented in a transparent and responsible manner, ensuring that they serve to support societal values of fairness and respect [48], [49]. De. Zarzà et al. [50] introduced a framework for modeling emergent cooperation and strategic adaptation in multi-agent systems using an extended coevolutionary theory integrated with Large LLMs. They developed a comprehensive model that incorporates the dynamics of adaptive learning and strategic interactions among heterogeneous agents within complex network environments. Their framework goes beyond traditional game theory by including mechanisms for real-time strategy adaptation influenced by LLMs, offering insights into the evolution of cooperation and competition among agents. The simulation results demonstrated the framework's ability to foster adaptive and resilient strategies in dynamic settings, achieving significant advancements in understanding multi-agent interactions. This work provides a solid foundation for future research in enhancing cooperative behaviors across various systems using LLM-based models.

The following table, Table (1), represents the summary of existing research in HS detection. Alongside, Table (2) provides a comparative analysis of HS detection in low-resource languages. The current methods for detecting HS often overlook the use of multilingual transformers and language-specific transformers, especially those utilizing the increasingly popular prompt-based fine-tuning technique within generative AI. Furthermore, many approaches focus primarily on word embedding techniques, often overlooking the crucial roles of regularization and hyperparameter tuning, which are essential for robust algorithm performance. In contrast, our study not only integrates advanced transformer models with machine learning-based classifiers but also emphasizes the importance of regularization. We set a new standard by incorporating hyperparameter tuning and prompt-based fine-tuning using generative configurations, along with explainable AI (XAI) techniques, for hate speech detection across multilingual datasets. An important aspect of our contribution to this paper will be our focus on addressing the limited body of work concerning XAI within HS. Through the integration of explainable AI methods into our research approach, we aim to illuminate the interpretability and transparency of our models.

III. METHODOLOGY

A. DATA PREPROCESSING

Data preprocessing is crucial for boosting the performance of learning classifiers. This process involves eliminating extraneous text and structuring the data into an organized format. By effectively preparing the data, this step markedly enhances the quality and usability of the data for both training and subsequent analysis, leading to significant improvements in the performance of learning models. For the preprocessing of the "text" column in our multilingual dataset, which includes 13 languages, we undertook a series of steps to refine the data. Initially, we transformed all uppercase letters into lowercase to ensure uniformity and removed irrelevant characters, including ASCII symbols. Following this, we executed both word and sentence tokenization to break down the text, while also removing stop words specific to each of the 13 languages to enhance the data's clarity. The process of removing stop words was performed for both ML and DL-based models. Additionally, we utilized the RegEx library in Python to sift through and handle various elements like digits, punctuation, and distinct patterns, such as email addresses, URLs, and numbers. The process of lemmatization was also applied, aiming to simplify words down to their fundamental or root form. This technique enhances uniformity in the application of words and boosts the model's capacity to identify connections among various forms of the same word.

For our transformer-based model data preparation, we carried out a selective set of preprocessing steps, specifically excluding the removal of stop words, as it's generally advised against in this context. Our preprocessing efforts primarily focused on converting uppercase letters to lowercase, discarding unnecessary characters such as ASCII symbols, and conducting word and sentence tokenization. This minimal preprocessing strategy was also designed to confront the problem of syntactic ambiguity, which has been a significant concern in traditional DL-based algorithms. Syntactic ambiguity occurs when words within a sentence might have several interpretations depending on the context, making it a difficult problem to interpret [61].

1) LANGUAGE-SPECIFIC PREPROCESSING TECHNIQUES

Multilingual NLP presents unique challenges, primarily due to the linguistic and structural diversity encountered across languages. Effective preprocessing is crucial as it directly impacts the performance of the machine learning models employed for hate speech detection. Here, we discuss our computational approach to handling these challenges across the distinct languages included in our research.

- **Preprocessing Variations**: Romanized languages (e.g., English, German) typically involve lowercasing, tokenization, and removal of stop words and punctuation. Tokenization in these languages is relatively straightforward due to clear word boundaries marked by spaces. Character-based languages (e.g., Chinese, Japanese) demand more nuanced approaches, such as character segmentation, which involves breaking down text into individual characters or phrases without spaces. Agglutinative languages (e.g., Turkish, Korean) where words are made up of different types of morphemes require morphological analysis to parse complex word forms

TABLE 1. Comparative analysis of state-of-the-art methods.

Ref	Dataset	Feature Set	Method	Results
[3]	RUSHOLD[2]	FastText, Contextual Embeddings	CNN, LSTM, BERT,RoBERTa, SVM	F1-score: 0.93
[15]	[38], [39]	Contextual Embeddings	mBERT, XLM-R, MAML	ROC-AUC: 0.79
[16]	Offensive Tweets	BoW, TF-IDF	KNN, DT, LR, NB	Accuracy: 0.92
[17]	AraPersonality, SemEval 2020 Arabic offensive	TF-IDF	DT, XGBoost, SVM, LR, RF, LSTM, BiLSTM, CNN	Accuracy: 0.82
[18]	Online Tweets	Count Vectorizer	XGBoost, LIME, SHAP	F1-score: 0.94
[20]	Online HS	Word2Vec, GloVe	LSTM, BiLSTM, CNN	F1-score: 0.85
[21]	[22], [23], [24]	Word2Vec, GloVe	BiLSTM, BERT	F1-score: 0.96
[32]	Dota 2	Synthetic	ChatGPT-4	Accuracy: 0.87
[33]	Xenophobia, Misogyny, and Homophobia	Contextual Embeddings	BERTIN, BETO, Spanish RoBERTa, mBERT	F1-score: 0.84
[38]	[39], [40]	Contextual Embeddings	VGAC	Accuracy: 0.85
[43]	RUSA-19, RUSA	Word2Vec CBoW, GLOVE, Fasttext, N-gram	CNN, RNN	Accuracy: 0.92
[45]	Cyberbullying (COL-DATASET)	Contextual Embeddings	BiLSMT, XLNet, ERNIE	F1-score: 0.90
[47]	Multlingual HS	GloVe	BiLSTM, BiGRU, CNN, LSTM, Bagging, Stacking	F1-Score: 0.81
[51]	Online Tweets, Social Media Comments	FastText, Cotextual Embeddings	BiLSTM-GRU, CNN-LSTM, Nor-BERT, Nor-T5, FLAN-T5, ELECTRA, nb-BERT, scandiBERT, mBERT, mBART	F1-Score: 0.98

TABLE 2. Comparative analysis of HS detection in low resource dialects.

Ref	Dataset	Hateful	Non Hateful	Evaluation
Bigoulaeva et al. [52]	German (GermEval-2018)	39.5%	60.5%	F1-Score: 0.98
Pereira et al. [53]	HaterNet	58.5%	42.5%	Acc: 0.83
Ayo et al. [54]	[22]	20%	80%	AUC: 0.96
Garcia et al. [55]	Spanish MisoCorpus 2020	58%	42%	Acc: 0.85
Fersini et al. [56]	Italian and English (AMI 2018)[3]	41%	59%	Acc: 0.83
Del et al. [57]	Italian	58.5%	42.5%	F1-Score: 0.998
Batarfi et al. [58]	HateEval 2019	41%	59%	F1-Score: 0.998
Ptaszynski et al. [59]	Polish (PolEval-2019)	89.71%	20.24%	Acc: 0.90
Trajano et al. [60]	Brazilian Portugese	31.5%	68.5%	F1-Score: 0.78

into their base units or morphemes. This process helps in reducing the vocabulary size and dealing with the rich inflection typical in these languages. Furthermore, to manage the diverse preprocessing techniques this research utilized the following modules,

- **Modular preprocessing pipelines**: In this module, each language is processed through a series of stages specifically designed for its characteristics. This modular approach allows for flexibility and scalability in processing large datasets.
- **Automated language detection**: Automated language detection is implemented initially to route texts to their respective preprocessing modules. This step ensures that each text is treated according to the linguistic rules and conventions pertinent to the language it is written in.
- **Parallel processing techniques**: These techniques are employed to handle the computational load efficiently. By distributing the preprocessing tasks across multiple processors, we can significantly speed up the processing time, making it feasible to handle large volumes of data within reasonable time frames.

B. WORD EMBEDDING

Word embeddings convert text into numerical vectors, enabling learning algorithms to interpret and analyze text data efficiently. This technique retains the essence of word meanings and contextual relevance, making it useful for a variety of NLP applications such as sentiment analysis, text classification, and language model development. These embeddings, which convert words into vectors in

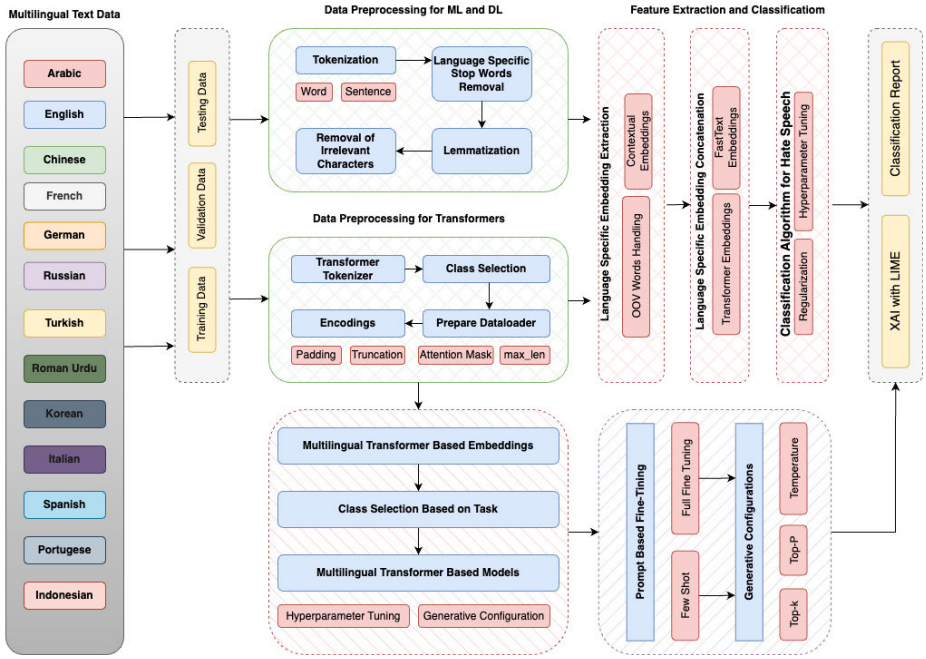

FIGURE 1. Proposed methodology architecture.

a dense space, enable machines to find word similarities, understand the complex nature of word meanings, and generalize learnings from training datasets. This considerably improves the performance of machines in executing a wide range of language tasks. In this study, we have employed FastText embeddings, due to their effectiveness in encapsulating semantic and contextual intricacies within textual datasets. FastText embeddings present significant improvements over conventional word embedding models by incorporating subword elements and offering enhanced handling of words not present in the vocabulary known as Out of Vocabulary (OoV). FastText embeddings are particularly advantageous, especially for languages with intricate morphological structures and variations. Traditional word vectors often overlook the internal structure of words, which contains valuable information. This information can be critical when generating representations for rare or misspelled words, making FastText an excellent choice in these contexts.

FastText is an advanced word embedding technique developed by Facebook's AI research team, capable of operating in both unsupervised and supervised modes. It features a vast vocabulary of 2 million words from the Common Crawl dataset, mapping each to a 300-dimensional vector space. What sets FastText apart is its integration of selected n-grams with individual words, enhancing its linguistic analysis. This technique supports two operational modes: unsupervised for learning word vectors and supervised for text classification. Our study utilizes both to assess their effectiveness across various linguistic tasks, offering insight into FastText's versatility and performance.

1) UNSUPERVISED FASTTEXT

In the unsupervised learning domain, FastText enhances the Word2Vec model by incorporating subword information and analyzing words through their constituent character n-grams. This means that for a given word like "killing", FastText does not only consider the whole word but also examines n-grams such as "kil", "ill", "lli", "lin", "ing", given an n-gram range typically set between 3 to 6 characters. Similarly, for "hate", it would look into segments like "hat", "ate". This method proves beneficial for grasping the significance of prefixes and suffixes, suggesting that words sharing similar subword structures may convey related meanings. FastText's unsupervised technique leverages extensive collections of untagged textual data to construct word vectors. These vectors then serve multiple purposes, including assessing word similarity, solving word analogies, or acting as input features in further NLP tasks. In our work, we utilized FastText's unsupervised word vectors, primarily the pre-trained[4] model which has contains the individual embeddings of 157 languages. This model was built using Common Crawl and Wikipedia using FastText's unsupervised learning technique, which incorporates subword information throughout the training process. By doing so, the model preserves the morphological characteristics of words and represents them as vectors in a 300-dimensional space. In algorithm 1, we convert text data into numerical representations using unsupervised FastText embeddings, adaptable for any language as indicated by the model file 'cc.lang.300.bin'. The core process involves loading a FastText model, and

[4]https://fasttext.cc/

then transforming text from a DataFrame into averaged word embeddings. This transformation handles OoV words by leveraging FastText's subword information, ensuring comprehensive coverage of linguistic elements. For texts without valid embeddings, a zero vector of the model's dimension is used. The result is a matrix *X_fasttext* of these embeddings, enabling the seamless integration of textual data into ML workflows, and offering a robust framework for analyzing text across multiple languages.

Algorithm 1 Create Unsupervised FastText Embeddings for Text Data

Require: FastText model file for text data from *df*
Ensure: Matrix *X_fasttext* of FastText embeddings

Load the FastText model: *ft_model* ← fasttext.load_model('cc.lang.300.bin');
function text_embeddings(*text*, *ft_model*)
  words ← split the *text*;
  embeddings ← initialize an empty list;
  for each *word* in *words* **do**
    vector ← *ft_model*.get_word_vector(*word*);
    if vector is valid **then**
      Append *vector* to *embeddings*;
    end if
  end for
  if *embeddings* is not empty **then return** mean of *embeddings* across axis 0;
  else
    return zero vector of length *ft_model*.get_dimension();
  end if
end function
X_fasttext ← stack vertically the result of **text_to_fasttext_embeddings** for each *text* in *df*;

2) SUPERVISED FASTTEXT

In supervised learning contexts, FastText excels at text classification by leveraging subword information and training on labeled datasets, where texts are associated with specific categories. It employs a hierarchical softmax approach based on Huffman coding to accelerate both training and prediction, allowing for efficient handling of extensive datasets and numerous classes. The model generates text representations by averaging word vectors, facilitating accurate and swift label predictions. This makes FastText highly effective for large-scale text classification tasks which is very suitable in our task. We performed a thorough exploration by deploying both supervised and unsupervised FastText models. While both approaches provided us with significant results, it was observed that supervised FastText consistently beat and outperformed its unsupervised counterpart. This analysis emphasizes the crucial role of labeled training data in text classification challenges, highlighting how supervised learning leverages explicit category data to achieve enhanced precision. The success of the supervised FastText model in this context demonstrates its suitability for classification tasks and underscores its value as a powerful instrument for boosting the accuracy of our studies. In the training processes, the model was trained using different learning rates and epochs to achieve maximum performance. This training technique allowed us to effectively use FastText embeddings to improve our classification performance. Following the model's training, the *quantize* method is employed to perform quantization, utilizing the *qnorm* and *retrain* parameters. This step is crucial for reducing the model size and potentially increasing inference speed without significantly compromising accuracy.

- **Quantization:** In ML, and specifically in relation to models like FastText, quantization refers to the process of lowering the precision of the model's parameters (e.g., weights and vectors). It has two basic types, Post-Training Quantization and Quantization Aware Training. In our paper, we implemented Post-Training Quantization. Post-training quantization was applied after a model had been fully trained. This approach does not require retraining the model, although some techniques may include a fine-tuning step to recover potential losses in accuracy.

C. MODELING APPROACHES

This section will thoroughly explore the ML, DL, and transformer-based models employed in this study. It will provide in-depth details of each model's architecture and its application within our research framework.

1) ML BASED MODELS

In our study on binary class HS detection for multilingual data, FastText embeddings were employed as the foundational input due to their proficiency in capturing semantic nuances across various languages. This approach allows for a robust feature representation, particularly beneficial for models like DT, SVM, LR, and RF, which were chosen for their diverse strengths in handling binary classification tasks. The versatility of these supervised ML-based models ensures a comprehensive evaluation of the linguistic and contextual patterns inherent in HS, facilitating effective detection across multiple languages. The SVM classifier outperformed both ML and DL-based classifiers in detecting HS in our multilingual dataset. This best performance of the SVM classifier is directly related to the strategic use of hyperparameter tuning and regularization approaches. The application of hyperparameter tuning and regularization was crucial in refining the models, aiming to achieve the highest level of optimized performance.

- **Hyperparameter Tuning for ML Based Models:** In the process of hyperparameter tuning, *GridSearchCV* was employed for LR and DT due to its exhaustive

search capability in relatively smaller hyperparameter spaces, ensuring the most optimal parameters are selected for these models. For SVM and RF, *Bayesian_Optimization* was utilized. This approach was chosen for its efficiency in larger hyperparameter spaces, where it outperforms *GridSearchCV* by using a probabilistic model to select the most promising hyperparameters to evaluate based on past results. *Bayesian_Optimization* is particularly adept at handling the complex dependencies between hyperparameters in these more sophisticated models, potentially leading to better performance with a significantly reduced computational cost. In equation 1, $f(x)$ is the objective function, $f(x^*)$ is the best value found so far, and $\mathbb{E}$ denotes the expected value over the posterior distribution. *EI* guides the selection of the next point x by quantifying the anticipated benefit relative to the current best, focusing the search on areas likely to yield improvements and efficiently using computational resources [62].

$$\mathrm{EI}(x) = \mathbb{E}\left[\max(f(x^*) - f(x), 0)\right] \tag{1}$$

Equation 2 illustrates the *GridSearchCV* method used in ML. This algorithm aims to identify the optimal model parameters through a comprehensive search. It operates by exhaustively exploring all possible combinations of hyperparameters, where h_1 belongs to the set H_1, h_2 to H_2, and so on, up to h_n in H_n. The objective is to maximize the evaluation metric, denoted by the *score* function, across these hyperparameter spaces. The *argmax* function is employed to pinpoint the exact combination of hyperparameters that achieves the highest score, reflecting the model's optimal accuracy O or overall performance. Table 3 highlights the hyperparameter and configuration details for ML-based models in this paper.

$$O\left(\underset{h_1 \in H_1,\, h_2,\, \ldots,\, h_n \in H_n}{argmax} score\,(h_1, h_2, \ldots, h_n)\right) \tag{2}$$

In configuring ML-based classifiers, specific hyperparameters were targeted to leverage their unique impact on model performance. For the DT model, the choice of $Split_{min}$ values aims to prevent overfitting by controlling the tree's depth, ensuring a model that generalizes well to new data. The RF model's $N - Estimators$ parameter was optimized to enhance ensemble learning effectiveness, where increasing the number of trees contributes to model accuracy and stability. The regularization parameter C for the SVM and LR models directly influences the strength of the penalty imposed on the magnitude of coefficients, balancing the model's complexity and its ability to fit the data without overfitting. By carefully tuning these hyperparameters, each model is fine-tuned to achieve optimal performance metrics. All these parameters across different models were meticulously optimized using *GridSearchCV* and *BaysianOptimization*.

Figure 2 represents the architecture of unsupervised FastText integration with language-specific transformer-based models.

2) DL BASED MODELS

In our study, we employed Long Short-Term Memory (LSTM) [63] and its variant, Bidirectional LSTM (BiLSTM) [64], [65]. These Recurrent Neural Network (RNN) based models excel in analyzing sequential information. LSTMs are renowned for their proficiency in recognizing long-range dependencies within data. The BiLSTM model further augments this strength by analyzing sequences from both forward and reverse perspectives. Additionally, the LSTM model's unique architecture enables it to effectively mitigate the vanishing gradient problem, ensuring more reliable learning from data over extended sequences. Moreover, we integrated Convolutional Neural Networks (CNN) with BiLSTM layers, capitalizing on CNN in identifying spatial characteristics and BiLSTM's adeptness at managing sequential data from both directions. This hybrid approach facilitates the modeling of intricate patterns within datasets.

1) **CNN-BiLSTM Architecture:** The CNN-BiLSTM architecture leverages the strengths of CNNs and Bi-LSTMs to effectively process textual data. CNNs are adept at capturing hierarchical feature representations from input data, which is crucial for identifying local patterns such as phrases or keywords within text. When combined with Bi-LSTMs, which excel at contextual understanding by integrating information across both previous and subsequent text, the architecture becomes particularly powerful for tasks like hate speech detection. This combination allows our model to discern subtle nuances and dependencies in text data, which are vital for accurately identifying and classifying varying forms of hate speech across different languages. The architecture's ability to handle multilingual datasets with diverse syntactic and semantic structures makes it an excellent choice for our study, enhancing both the accuracy and robustness of our HS detection system. The CNN-BiLSTM model represents a combination of two convolutional and BiLSTM layers for enhanced data analysis capabilities. The model employs two convolutional layers, each with 64 filters and kernel sizes of 4 and 3, respectively, paired with 'relu' activation for effective spatial feature extraction. Following these layers is a MaxPooling layer designed to compress the data size and boost processing efficiency. For capturing the dynamics over time, the model incorporates a BiLSTM component with two layers, containing 50 and 30 units each, vital for the analysis of sequential data. The architecture is rounded off with a dense layer activated by 'softmax', rendering it highly effective for classification purposes. This design is particularly proficient in tasks

TABLE 3. Hyperparameter and configuration details for ML-based models.

Model	Regularization	Hyperparameter Tuning
DT	$Split_{min}$: [5, 10, 15]	$GridSearchCV$
RF	$N - Estimators$: [100, 200, 300]	$Bayesian\_Optimization$
SVM_{linear}	C: [0.1, 1, 10]	$Bayesian\_Optimization$
LR	C: [10, 100, 1000]	$GridSearchCV$

that require simultaneous spatial and temporal data analysis.

2) **Regularization Techniques:** Regularization is a strategy used in learning classifiers to prevent overfitting, which occurs when a model performs well on training data but badly on unseen test data [66], [67]. The robust performance of the CNN-BiLSTM model is considerably enhanced by the implementation of kernel L2 regularization, set at a lambda value of 0.01 for both the BiLSTM and CNN layers. L2 regularization is crucial for reducing the magnitude of the weights, which encourages the model to favor smaller weight values [68]. This method fulfills two crucial objectives: it diminishes the risk of overfitting and maintains the model's generalization capability, ensuring its reliable performance across unfamiliar datasets. The decision to opt for L2 over L1 regularization was made deliberately. L1 regularization can induce sparsity by reducing certain weights to zero, which could potentially result in underfitting, a challenge observed during preliminary experiments. The mathematical expressions (3) and (4) provide the calculations for $L1$ and $L2$ regularization respectively.
In the specified model, **w** represents the weight vector, with each component w_i denoting the i-th weight in the vector. The term λ is used as the regularization coefficient, which helps in controlling the complexity of the model to avoid overfitting. The variable n indicates the total number of weights contained within the vector **w**.

$$L1(\mathbf{w}) = \lambda \sum_{i=1}^{n} |w_i| \tag{3}$$

$$L2(\mathbf{w}) = \lambda \sum_{i=1}^{n} w_i^2 \tag{4}$$

$L1$ regularization incorporates the absolute magnitude of coefficients as a penalty to the loss function. This addition of absolute values introduces a non-linear penalty based on the weights, making $L1$ regularization conducive to sparse outcomes where numerous coefficients become precisely zero. $L2$ regularization introduces the squared magnitude of coefficients as a penalty to the loss function. This squaring process results in a smoother, differentiable penalty, even at $w_i = 0$. Contrary to $L1$ regularization, $L2$ does not lead to sparse models because it generally does not push coefficients to become exactly zero, although it may reduce them to small values.

3) **Hyperparameter Tuning for DL-Based Models:** In the hyperparameter optimization process for DL-based models, we methodically adjusted the model's learning process through targeted experimentation. The training period was set to 10 epochs, a duration chosen to balance effective learning against the risk of overfitting, and ended when the model's loss decreased. For our HS detection task, which is a binary classification problem, we utilized the $cross - entropy$ loss function, renowned for its precision in evaluating the congruence between predicted probabilities and actual binary outcomes. To enhance our model's learning efficiency, we chose the *Adam* optimizer for its capability to adapt the *learning rate* dynamically, set at $2e - 5$. This feature of the $Adam_o ptimizer$ significantly aids in the model's performance toward finding the best set of parameters by adjusting the learning rate according to the needs of the training process. Table 4 highlights the configuration and hyperparameter details for DL-based models.

3) TRANSFORMER BASED MODELS

The Transformer is a model in NLP engineered for sequence-to-sequence tasks, utilizing a self-attention mechanism to handle long-range dependencies. It is structured around two core components: the encoder and the decoder [69].

This paper utilized both language-specific and multilingual Transformer-based models, designed for targeted languages and multiple languages, respectively, to perform text classification tasks.

1) **Language Specific Transformer:** Our research focuses on HS binary classification across 13 distinct languages, where one of our contributions involves integrating language-specific Transformer-based models with ML classifiers to achieve optimal robustness in results. Additionally, this approach allows us to leverage the nuanced understanding of each language's characteristics, enhancing the overall performance and accuracy of the classification. The algorithm 2 shows the steps taken to extract language-specific transformer-based models. These extracted embeddings from 1 and 2 were further combined and inputted to ML classifiers for the evaluation of the results.
Table 5 represents the language-specific transformer-based models along with their configuration details that

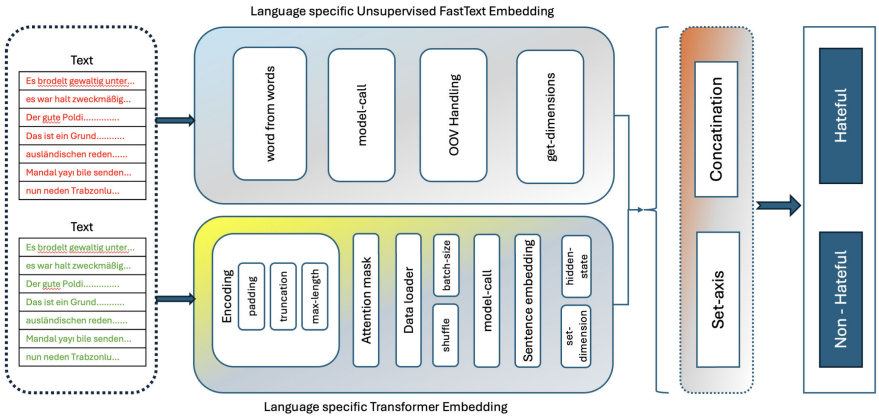

FIGURE 2. Unsupervised FastText with language-specific transformer architecture.

TABLE 4. Configuration details for DL models.

model	model layer	dense layer	dropout layer	pooling layer	Regularization	epochs	function
LSTM	3	2	2	-	L2	10	softmax
BiLSTM	3	2	2	-	L2	10	softmax
CNN-LSTM	3	2	2	2	L2	10	relu, softmax

have been used in our study. To the best of our work, we utilized these transformers from hugging-face where they are easily accessible.

2) **Multilingual Transformer Based Models:** Our research also encompassed the use of various multilingual transformer-based models, such as mBERT, mBART, ELECTRA, and FLAN-T5.

   a) **mBERT:** BERT, a Transformer-based model, was self-trained on a vast, multilingual corpus, utilizing only raw text without human-labeled data through automated techniques to create inputs and labels. Meanwhile, its variant, mBERT, targeted 104 languages using Wikipedia articles and was pre-trained with a Masked Language Modeling (MLM) technique [37].

   b) **ELECTRA:** BERT's pre-training involves masking input tokens and predicting them, whereas Electra introduces a more efficient replaced token detection technique. Unlike BERT, Electra substitutes tokens with alternatives from a generator network, and a discriminative model determines if tokens have been replaced. This method is detailed with probabilities for token generation x_t using a softmax layer [70]. The equations 5 and 6 represent the equations for ELECTRA referring to its discriminator and generator modules [71].

$$P_G(x_t|x) = \frac{e^{x_t^T h_G(x)_t}}{\sum_{x_0} e^{x_0^T h_G(x)_t}} \tag{5}$$

Following is the mathematical expression for the **discriminator** part of Electra,

$$L = -\mathbb{E}[y \ln D + (1-y)\ln(1-D)] \tag{6}$$

   c) **FLAN-T5:** FLAN-T5, building upon the Text-to-Text Transfer Transformer (T5) framework [72], marks a notable development in NLP, designed specifically for instruction-based fine-tuning. By being trained across a diverse set of tasks, FLAN-T5 demonstrates enhanced flexibility and performance in text-to-text tasks [73]. Its capabilities extend to summarizing dialogues and performing text classification with high efficiency, proving essential for practical applications. FLAN-T5's ability to automate the sorting of text into specific categories, such as Sentiment Analysis, spam detection, and topic modeling, underscores its utility in handling a variety of text-based challenges.

   d) **mBART:** The Multilingual Bidirectional and Auto-Regressive Transformers (mBART) model, pre-trained on a variety of monolingual datasets in multiple languages, aims to improve language understanding across different contexts, essential for multilingual task performance. mBART stands out for pre-training a full sequence-to-sequence model by cleaning texts in several languages, moving beyond the previous focus on individual model components. Meanwhile, mBERT focuses on cross-lingual comprehension, suitable for language transfer learning tasks.

Algorithm 2 Extract Transformer-Based Embeddings for Text Data

Require: Language-specific transformer model, text data from *language*
Ensure: Tensor *dataset_embeddings* of Transformer-based embeddings

Load tokenizer and model for the specific language; ▷ Preparation step
texts ← *language*; ▷ Assign text data to process
encoding ← *tokenizer*(*texts*, *padding* = *True*, *truncation* = *True*, *max_length* = 512, *return_tensors* = "*pt*"); ▷ Tokenize text data
input_ids ← *encoding*['*input_ids*'];
attention_mask ← *encoding*['*attention_mask*'];
Create a DataLoader: *data_loader* ← *DataLoader*(*TensorDataset*(*input_ids*, *attention_mask*), *batch_size* = 32, *shuffle* = *False*); ▷ Prepare data loader for batch processing
function extract_bert_embeddings(*data_loader*, *model*, *device*)
    model.config.output_hidden_states ← *True*; ▷ Configure model to output hidden states
    embeddings ← initialize an empty list; ▷ Initialize list to collect embeddings
    device ←' *mps*'; ▷ Set processing device, modify as needed
    for each *batch* in *data_loader* **do**
        input_ids, *attention_mask* ← [*t.to*(*device*)*fortinbatch*]; ▷ Move batch data to device
        outputs ← *model*(*input_ids*, *attention_mask* = *attention_mask*); ▷ Process batch through model
        hidden_states ← *outputs.hidden_states*;
        last_hidden_states ← *hidden_states*[−1]; ▷ Get the last layer's hidden states
        sentence_embeddings ← *torch.mean*(*last_hidden_states*, *dim* = 1); ▷ Compute mean across the sequence length
        Append *sentence_embeddings* to *embeddings*;
    end for
    embeddings ← *torch.cat*(*embeddings*, *dim* = 0); ▷ Concatenate all embeddings into a tensor
    return *embeddings*;
end function
dataset_embeddings ← EXTRACT_BERT_EMBEDDINGS *data_loader*, *model*, *device*;

In our research, we utilized mBART for classification tasks with the *SequenceClassification* class type and *MBartTokenizer*, training it to reconstruct original texts from modified inputs [74]. The equation 7 represents the equation for mBART.

$$L_\theta = \sum_i \sum_x \log P(x|g(x);\theta) \tag{7}$$

4) GENERATIVE CONFIGURATIONS

To enhance our multilingual transformers, we adjusted essential hyperparameters such as batch sizes, learning rates, and epochs, resulting in improved performance. We also implemented generative configuration parameters, including $top-k$ and $top-p$ sampling methods, to refine the model's output creativity and token count during inference [75], [76].

$Top-k$ sampling confines the next word selection to the top k probable words from the model's distribution, striking a balance between creativity and coherence in text generation. This method ensures that choices are confined to a likely subset as determined by the model's predictions.

$$P(w) = \begin{cases} \dfrac{e^{P(w)}}{\sum_{w'} e^{P(w')}} & \text{if } w \in \text{ top-}k \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

$Top-p$ sampling, or nucleus sampling, selects a variable number of words whose cumulative probability exceeds a threshold p. This approach enables more dynamic and contextually relevant text generation by focusing on a variable-sized set of probable outcomes, allowing for richer variations in the generated content [77],

$$P(w) = \frac{1}{\sum_{w' \in V : P(w') \geq p} P(w')}. \tag{9}$$

Additionally, we introduced a *temperature* parameter to modulate the probability distribution for the next token prediction. This parameter scales the softmax output, influencing the randomness of text generation: higher temperatures produce more diverse outputs, while lower temperatures result in more predictable text. This feature is crucial for tuning the balance between variety and reliability in generated text.

$$P(w) = \frac{exp^{(P(w)/\tau)}}{\sum_{w'} exp^{(P(w')/\tau)}} \tag{10}$$

5) PROMPT BASED FINE-TUNING

Prompt-based learning guides transformer models using concise prompts, leveraging pre-existing knowledge, as opposed to traditional algorithms that rely on large datasets. Our research combined few-shot and full fine-tuning methods. Few-shot fine-tuning involves training on a minimal set of examples, ideal for adapting models to tasks with scarce data and promoting effective generalization. Conversely, full

TABLE 5. Configuration details for language specific transformers.

Language	Models	Epochs	lr	Batch Size
Arabic	AraBERT[5], AraElectra[6], Albert Arabic[7]	10	2e-5	32
English	BERT[8], XLNet[9], RoBERTa[10]	10	2e-5	32
Chinese	Chinese BERT[11], ERNIE[12], MacBERT[13]	10	2e-5	32
French	CAMEMBERT[14], FlauBERT[15]	10	2e-5	32
German	GermanBERT[16], DeepestBERTGerman[17]	10	2e-5	32
Russian	RussianBERT[18], XLM-RoBERTa Russian[19]	10	2e-5	32
Turkish	TurkishBERT[20], DistilTurkishBERT[21]	10	2e-5	32
Roman Urdu	Roman Urdu BERT[22]	10	2e-5	32
Korean	KoBERT[23], KRBERT[24]	10	2e-5	32
Italian	UmBERT[25], DehateBERT[26]	10	2e-5	32
Spanish	BETO[27], AlbertSpanish[28], XLM-RoBERTa Spanish[29]	10	2e-5	32
Portugese	DehateBERT Portugese[30], BERTImbau[31]	10	2e-5	32
Indonesian	IndoBERT[32], IndoLEM[33]	10	2e-5	32

TABLE 6. Training arguments for prompt based fine-tuning.

Parameters	Language Model
Learning Rate	2e-5
num_train_epochs	5
evaluation_strategy	'epoch'
weight_decay	0.01
per_device_train_batch_size	32
logging_steps	1
optim	'adamw_torch'

fine-tuning requires training on extensive data, aimed at specialized tasks but is more resource-intensive. We employed FLAN-T5 and mBART for their suitability in prompt-based interactions, using prompts such as "Please classify the following sentence: Hateful or Non-hateful" to evaluate both techniques.

Algorithm 3 describes a method for preparing prompts for fine-tuning with multilingual Transformer-based models. It begins by loading a specific tokenizer for the chosen model. For each entry in the dataset, it concatenates a predefined natural language prompt with the text data and then tokenizes this combined prompt along with the corresponding label. The process ensures both inputs and labels are prepared in a format suitable for the Transformer model, including padding and truncation to a maximum length of 512 tokens and formatting the tensors for the model. Finally, the dataset dictionary is updated with these tokenized inputs and labels, making it ready for the fine-tuning process. This approach leverages the power of natural language prompts to guide the model's learning, enhancing its ability to understand and perform the specified task.

Table 6 shows the training arguments used to prepare our model for prompt-based fine-tuning.

IV. RESULTS AND DISCUSSION

For the evaluation of the results, standard metrics of accuracy, precision, recall, and f1-score were utilized to quantify the model's classification performance across all experiments.

Algorithm 3 Preparing Prompt for Fine-Tuning With Multilingual Transformer-Based Models

Require: Dataset dictionary containing text data and labels for fine-tuning

Ensure: Updated dataset dictionary with tokenized inputs and labels for training

Load the tokenizer specific to the transformer model;
for each *dataset_dict* in Dataset **do**
  $prompt \leftarrow$ "natural language prompt" + $dataset\_dict['text']$;
  $end\_prompt \leftarrow dataset\_dict['label']$ corresponding to the task;
  $input\_ids \leftarrow tokenizer(prompt, padding = True, truncation = True, max\_length = 512, return\_tensors = "pt")$;
  $labels \leftarrow tokenizer(end\_prompt, padding = True, truncation = True, max\_length = 512, return\_tensors = "pt")$;
  Update *dataset_dict* with *input_ids* and *labels*;
end for
return the updated Dataset containing tokenized inputs and labels;

A. ANALYSIS OF RESULTS: UNSUPERVISED FASTTEXT WITH ML

In our initial assessment of the approach, we conducted experiments employing Unsupervised FastText combined with ML-based classifiers, incorporating hyperparameter tuning and quantization techniques as mentioned in 3. The dataset was divided into a training and testing split of 80% and 20% respectively. Table 7 represents the evaluation scores for ML-based classifiers using unsupervised FastText.

In table 7, for languages like English and Indonesian, the SVM model shows a higher level of precision and recall, both at 0.91 and 0.86 respectively, indicating its capability to correctly identify positive instances while minimizing false positives. The accuracy of these models is also in line with

TABLE 7. ML models: Evaluation scores with unsupervised FastText.

Language	Model	P	R	Acc	F
Arabic	DT	0.68	0.68	0.68	0.68
	RF	0.80	0.79	0.79	0.77
	LR	0.80	0.79	0.79	0.78
	SVM	**0.82**	**0.81**	**0.81**	**0.81**
English	DT	0.77	0.77	0.77	0.77
	RF	0.88	0.88	0.88	0.88
	LR	0.87	0.87	0.87	0.87
	SVM	**0.91**	**0.91**	**0.91**	**0.91**
Chinese	DT	0.56	0.55	0.55	0.56
	RF	0.60	0.66	0.66	0.56
	LR	**0.61**	0.67	0.67	0.55
	SVM	**0.60**	**0.67**	**0.67**	**0.56**
French	DT	0.72	0.72	0.72	0.72
	RF	0.83	0.82	0.82	0.81
	LR	0.83	0.82	0.82	0.82
	SVM	**0.84**	**0.82**	**0.82**	**0.82**
German	DT	0.54	0.54	0.54	0.54
	RF	0.62	0.62	0.62	0.62
	LR	0.64	0.64	0.64	0.64
	SVM	**0.65**	**0.65**	**0.65**	**0.65**
Russian	DT	0.76	0.76	0.76	0.76
	RF	0.83	0.82	0.82	0.82
	LR	0.84	0.83	0.83	0.83
	SVM	**0.86**	**0.86**	**0.86**	**0.86**
Turkish	DT	0.74	0.73	0.73	0.73
	RF	**0.85**	**0.83**	**0.83**	**0.76**
	LR	0.83	0.81	0.81	0.74
	SVM	0.83	0.81	0.81	0.74
Roman Urdu	DT	0.74	0.73	0.73	0.73
	RF	**0.85**	**0.83**	**0.83**	**0.76**
	LR	0.83	0.81	0.81	0.74
	SVM	0.83	0.81	0.81	0.74
Korean	DT	0.57	0.57	0.57	0.57
	RF	0.63	0.63	0.63	0.63
	LR	**0.69**	**0.69**	**0.69**	**0.68**
	SVM	0.68	0.68	0.68	0.68
Italian	DT	0.67	0.67	0.67	0.67
	RF	0.78	0.77	0.77	0.76
	LR	0.78	0.78	0.78	0.76
	SVM	**0.79**	**0.79**	**0.79**	**0.79**
Spanish	DT	0.64	0.63	0.63	0.64
	RF	0.71	0.72	0.72	0.67
	LR	0.73	0.74	0.74	0.71
	SVM	**0.76**	**0.76**	**0.76**	**0.73**
Portugese	DT	0.62	0.67	0.67	0.62
	RF	0.70	0.72	0.72	0.68
	LR	0.74	0.75	0.75	0.72
	SVM	**0.75**	**0.76**	**0.76**	**0.74**
Indonesian	DT	0.69	0.68	0.68	0.69
	RF	0.81	0.81	0.81	0.81
	LR	0.83	0.83	0.83	0.83
	SVM	**0.86**	**0.86**	**0.86**	**0.86**

precision and recall, which illustrates the model's overall reliability. The performance in languages such as Russian and Spanish is noteworthy as well, with the SVM achieving the highest scores among the classifiers, particularly in precision and recall score of 0.86 for Russian and 0.76 for Spanish demonstrating its ability to generalize well across different linguistic contexts. In the case of Chinese and Korean, the scores reflect a relatively lower performance, with precision and recall for the SVM model at 0.60 and 0.67 for Chinese, and 0.68 for both metrics in Korean. These figures suggest the models face more difficulty in classifying text in these languages, which could stem from the inherent complexity of the languages, the composition of the datasets, or both. For Roman Urdu and Turkish, the RF model exhibits high precision and recall, both at 0.85 for Turkish and slightly lower for Roman Urdu at 0.83, yet the F1-scores for these languages are 0.76 and 0.74, respectively. This disparity indicates that while the model is adept at identifying true positives, it may not be as effective when considering both precision and recall together, as reflected by the F1-score.

The German language results show a moderate performance with the highest precision and recall achieved by the SVM model at 0.65. This suggests a fair level of classification capability, which could be improved with further model tuning. In Spanish and Portuguese, the SVM model's precision and recall are reasonably good, at 0.76 for Spanish and 0.75 for Portuguese. This indicates a reliable performance in these languages, with the models being able to correctly identify instances of HS with a lower likelihood of false positives. Indonesian stands out with the SVM model showing high precision and recall at 0.86, matched by an accuracy of the same value. This high level of performance indicates that the SVM model is particularly effective for the Indonesian language within this study. From the results, it is clear that the SVM model has shown consistent and best performance across different languages and LR has provided the maximum results across Chinese are Korean languages as compared to other models.

B. ML MODELS: EVALUATION SCORES WITH UNSUPERVISED FASTTEXT AND LANGUAGE SPECIFIC TRANSFORMERS

In this evaluation, we conducted experiments after the integration of language-specific unsupervised FastText embeddings alongside language-specific transformer-based models for each language present in the dataset. The table 8 represents the results of language-specific transformers with unsupervised FastText.

For Arabic, the models paired with *AraBERT* and *AraELECTRA* notably achieve high precision and recall, both peaking at 0.88, which indicates a strong alignment between the models' predictions and the actual data. In the English context, the combination of RF+BERT stands out with a precision and recall at 0.94, suggesting an exceptional ability to correctly identify HS instances while maintaining a low rate of false positives. In Chinese, the models integrated with *ERNIE* and *MacBERT* show improved metrics, with *SVM* + *MacBERT* reaching precision and recall of 0.81, reflecting a commendable proficiency in understanding and classifying Chinese text. Turning to French, we observe that the *SVM* + *CAMEMBERT* model achieves a precision, accuracy, recall, and recall of 0.85, underscoring its effectiveness in the nuanced task of HS detection in French. German language models, such as *SVM* + *DeepestBERT*, demonstrate a performance with an accuracy and F1-score of 0.67, indicating a moderate score accuracy in the classification process.

Russian language classifiers coupled with *BERT* and *XLM* − *RoBERTa* exhibit precision and recall rates at 0.88, highlighting the robustness of these models in dealing with the Russian language. Turkish classifiers, particularly when combined with *TurkishBERT*, achieve a solid score of 0.94 in terms of accuracy and F1-score, reflecting an outstanding capability to discern HS content in Turkish. For the less commonly represented Roman Urdu, the *SVM* + *RomanBERT* model also achieves high precision and recall, both at 0.84, demonstrating the efficacy of this language-model pairing. The Korean classifiers, despite being challenged by the complexity of the language, show improved performance as compared to simple unsupervised FastText in table 7. It has evaluation scores of 0.74 across all four matrices when combined with *KRBERT*, this significant enhancement is the result of our designed architecture. In Italian, the *SVM* + *dehateBERT* combination exhibits the highest precision and recall of 0.87, suggesting it's highly suited for the Italian HS detection task. Spanish classifiers paired with *BETO* and *XLM* −*RoBERTa* −*Spanish* models show consistent precision and recall rates, with *SVM* + *BETO* reaching 0.80, indicating a strong predictive performance. Portuguese classifiers, especially *SVM* + *dehateBERTPortuguese*, present precision and recall rates of 0.77, pointing to a reliable classification capability. Lastly, Indonesian classifiers coupled with *IndoBERT* and *IndoLEM* consistently hit the peak with precision and recall at 0.93, signifying the models' high competence in this language.

These findings demonstrate that while some language-model combinations provide robust detection capabilities, others suggest room for further optimization. The consistently high scores in certain models, such as *SVM* + *dehateBERT* for Italian and *SVM* +*TurkishBERT* for Turkish, indicate particular effectiveness in those language contexts. Conversely, the relatively lower scores in languages like Korean necessitate further investigation into model suitability and potential refinements. Overall, these results contribute valuable knowledge to the field, enhancing our understanding of model performance in multilingual HS detection tasks. For DT, the performance was notably consistent across all languages, marking an improvement over prior outcomes when it was solely paired with unsupervised FastText. Following this analysis, it is evident that integrating unsupervised FastText with language-specific transformer-based models significantly improves performance metrics across the board. This enhancement is particularly notable for languages

TABLE 8. ML Models: Evaluation Scores with Unsupervised FastText and Language specific transformers.

Language	Model	P	R	Acc	F1
Arabic	DT+AraBERT	0.86	0.86	0.86	0.86
	RF+AraBERT	0.88	0.88	0.88	0.88
	LR+AraBERT	0.88	0.88	0.88	0.88
	SVM+AraBERT	**0.88**	**0.88**	**0.88**	**0.88**
	DT+AraELECTRA	0.87	0.87	0.87	0.87
	RF+AraELECTRA	0.88	0.88	0.88	0.88
	LR+AraELECTRA	0.88	0.88	0.88	0.88
	SVM+AraELECTRA	**0.88**	**0.88**	**0.88**	**0.88**
	DT+AraELECTRA	0.80	0.79	0.79	0.80
	RF+AraELECTRA	0.82	0.82	0.82	0.82
	LR+AraELECTRA	0.82	0.82	0.82	0.82
	SVM+AraELECTRA	0.83	0.83	0.83	0.83
English	DT+BERT	0.90	0.90	0.90	0.90
	RF+BERT	0.94	0.94	0.94	0.94
	LR+BERT	0.93	0.93	0.93	0.93
	SVM+BERT	**0.93**	**0.93**	**0.93**	**0.93**
	DT+XLNet	0.90	0.90	0.90	0.90
	RF+XLNet	0.91	0.91	0.91	0.91
	LR+XLNet	0.91	0.91	0.91	0.91
	SVM+XLNet	0.91	0.91	0.91	0.91
Chinese	DT+Chinese BERT	0.65	0.65	0.65	0.65
	RF+Chinese BERT	0.76	0.76	0.76	0.74
	LR+Chinese BERT	0.79	0.79	0.79	0.79
	SVM+Chinese BERT	0.79	0.79	0.79	0.79
	DT+ERNIE	0.79	0.79	0.79	0.79
	RF+ERNIE	0.81	0.81	0.81	0.81
	LR+ERNIE	0.80	0.80	0.80	0.80
	SVM+ERNIE	**0.81**	**0.81**	**0.81**	**0.81**
	DT+Mac BERT	0.80	0.80	0.80	0.80
	RF+Mac BERT	0.81	0.81	0.81	0.81
	LR+Mac BERT	0.80	0.80	0.80	0.80
	SVM+Mac BERT	**0.81**	**0.81**	**0.81**	**0.81**
French	DT+CAMEM BERT	0.82	0.82	0.82	0.82
	RF+CAMEM BERT	0.84	0.84	0.84	0.84
	LR+CAMEM BERT	0.84	0.84	0.84	0.84
	SVM+CAMEM BERT	**0.85**	**0.85**	**0.85**	**0.85**
	DT+FlauBERT	0.78	0.78	0.78	0.78
	RF+FlauBERT	0.83	0.83	0.83	0.82
	LR+FlauBERT	**0.83**	**0.83**	**0.83**	**0.83**
	SVM+FlauBERT	0.83	0.83	0.83	0.82
German	DT+German BERT	0.64	0.64	0.64	0.64
	RF+German BERT	0.64	0.64	0.64	0.64
	LR+German BERT	0.64	0.64	0.64	0.64
	SVM+German BERT	0.65	0.65	0.65	0.65
	DT+Deepest BERT	0.66	0.66	0.66	0.66

TABLE 8. *(Continued.)* ML Models: Evaluation Scores with Unsupervised FastText and Language specific transformers.

Language	Model	P	R	Acc	F1
	RF+Deepest BERT	0.66	0.66	0.66	0.66
	LR+Deepest BERT	0.67	0.67	0.67	0.67
	SVM+Deepest BERT	**0.67**	**0.67**	**0.67**	**0.67**
Russian	DT+Russian BERT	0.88	0.88	0.88	0.88
	RF+Russian BERT	0.88	0.88	0.88	0.88
	LR+Russian BERT	0.88	0.89	0.88	0.89
	SVM+Russian BERT	**0.88**	**0.88**	**0.88**	**0.89**
	DT+XLM-RoBERTa Russian	0.87	0.87	0.87	0.87
	RF+XLM-RoBERTa Russian	0.88	0.88	0.88	0.88
	LR+XLM-RoBERTa Russian	0.88	0.88	0.88	0.88
	SVM+XLM-RoBERTa Russian	**0.88**	**0.88**	**0.88**	**0.88**
Turkish	DT+Turkish BERT	0.94	0.94	0.94	0.94
	RF+Turkish BERT	0.94	0.94	0.94	0.94
	LR+Turkish BERT	0.94	0.94	0.94	0.94
	SVM+Turkish BERT	**0.94**	**0.94**	**0.94**	**0.94**
	DT+Distil Turkish BERT	0.93	0.93	0.93	0.93
	RF+Distil Turkish BERT	0.93	0.93	0.93	0.93
	LR+Distil Turkish BERT	0.88	0.89	0.88	0.89
	SVM+DistilcTurkish BERT	0.93	0.94	0.94	0.94
Roman Urdu	DT+Roman BERT	0.82	0.82	0.82	0.82
	RF+Roman BERT	0.84	0.84	0.84	0.84
	LR+Roman BERT	0.84	0.84	0.84	0.84
	SVM+Roman BERT	**0.84**	**0.84**	**0.84**	**0.84**
Korean	DT+KoBERT	0.57	0.57	0.57	0.57
	RF+KoBERT BERT	0.63	0.63	0.63	0.63
	LR+KoBERT BERT	0.70	0.69	0.69	0.68
	SVM+KoBERT BERT	0.73	0.56	0.56	0.41
	DT+KRBERT	0.73	0.73	0.73	0.73
	RF+KRBERT	0.74	0.74	0.74	0.74
	LR+KRBERT	0.74	0.74	0.74	0.74
	SVM+KRBERT	**0.74**	**0.74**	**0.74**	**0.74**
Italian	DT+UmBERTO	0.82	0.82	0.82	0.82
	RF+UmBERTO	0.84	0.84	0.84	0.84
	LR+UmBERTO	0.84	0.84	0.84	0.84
	SVM+UmBERTO	0.84	0.84	0.84	0.84
	DT+dehateBERT	0.84	0.84	0.84	0.84
	RF+dehateBERT	0.86	0.86	0.86	0.86
	LR+dehateBERT	0.87	0.87	0.87	0.87
	SVM+dehateBERT	**0.87**	**0.87**	**0.87**	**0.87**

TABLE 8. *(Continued.)* ML Models: Evaluation Scores with Unsupervised FastText and Language specific transformers.

Language	Model	P	R	Acc	F1
Spanish	DT+BETO	0.80	0.79	0.79	0.79
	RF+BETO	0.80	0.80	0.80	0.80
	LR+BETO	0.80	0.80	0.80	0.80
	SVM+BETO	**0.80**	**0.80**	**0.80**	**0.80**
	DT+XLM-RoBERTa Spanish	0.77	0.77	0.77	0.77
	RF+XLM-RoBERTa Spanish	0.78	0.77	0.78	0.77
	LR+XLM-RoBERTa Spanish	0.78	0.78	0.78	0.78
	SVM+XLM-RoBERTa Spanish	0.78	0.77	0.78	0.77
	DT+Albert Spanish	0.73	0.72	0.72	0.72
	RF+Albert Spanish	0.75	0.76	0.76	0.75
	LR+Albert Spanish	0.75	0.75	0.75	0.75
	SVM+Albert Spanish	0.76	0.76	0.76	0.76
Portugese	DT+BERTImbau	0.75	0.76	0.76	0.75
	RF+BERTImbau	0.75	0.76	0.76	0.76
	LR+BERTImbau	0.80	0.75	0.75	0.75
	SVM+BERTImbau	0.77	0.77	0.77	0.77
	DT+dehateBERT Portugese	0.72	0.72	0.72	0.72
	RF+dehateBERT Portugese	0.77	0.77	0.77	0.77
	LR+dehateBERT Portugese	0.77	0.77	0.77	0.77
	SVM+dehateBERT Portugese	**0.77**	**0.77**	**0.77**	**0.77**
Indonesian	DT+IndoBERT	0.93	0.93	0.93	0.93
	RF+IndoBERT	0.93	0.93	0.93	0.93
	LR+IndoBERT	0.93	0.92	0.92	0.92
	SVM+IndoBERT	0.93	0.93	0.93	0.93
	DT+IndoLEM	0.93	0.93	0.93	0.93
	RF+IndoBERT	0.93	0.93	0.93	0.93
	LR+IndoBERT	0.93	0.93	0.93	0.93
	SVM+IndoBERT	**0.93**	**0.93**	**0.93**	**0.93**

such as German, Chinese, and Korean, where unsupervised FastText alone struggled to demonstrate satisfactory results. This synergy suggests a promising direction for advancing the NLP capabilities in linguistically diverse settings.

C. ANALYSIS OF THE RESULTS: SUPERVISED FASTTEXT EMBEDDINGS

In this evaluation, we conducted three distinct experiments. The details are mentioned below.

1) CROSS-LINGUISTIC MODEL EVALUATION VIA N-1 LANGUAGE ANALYSIS

Initially, we combined quantized supervised FastText embeddings with ML classifiers, creating a hybrid model. This model was trained on data from 12 languages and subsequently tested on a 13th language, employing a transfer learning approach using the $N-1$ strategy. This strategy serves the purpose of cross-language generalizability where our model trained on multiple languages can effectively generalize to a new, unseen language. After the training in 12 languages, our model will work as a multilingual model which will be further used for the test data for the unseen language. The results can be seen in table 9. For the Arabic language, precision and recall are both at 0.54 for DT and RF, with LR slightly higher at 0.51 for precision and 0.62 for recall, which may suggest a better capture of true positives but also an increased likelihood of false positives. The SVM's balanced precision and recall at 0.53 indicate

similar discernment as DT and RF. English and Chinese test sets show an even performance across DT and RF, with precision and recall at 0.50 and 0.54 respectively. The uniformity of these scores may reflect the models' limitations in generalizing from the training to the test set, which is an intrinsic challenge of transfer learning.

Upon analyzing the results from the French and German test sets, it is observed that the performance metrics for the LR and SVM models are closely matched with those of DT and RF, with all models exhibiting precision and recall scores in the range of 0.51 to 0.57 for French and 0.51 to 0.52 for German. These scores indicate a moderate level of model precision and ability to recall relevant instances, suggesting a potential need for enhanced feature representation to capture the full complexity of these languages. The F1-scores and accuracy metrics for the French test set present a consistent pattern, with all models achieving a score of 0.57. These poor results need to be further improved for a better understanding of the learning classifier. Comparatively, the Russian test set shows a divergence between models, with LR and SVM attaining a higher recall of 0.59, and corresponding F1-scores of 0.51, indicating a better grasp of the true positives. However, the relatively lower precision suggests that these models may also be including more false positives in their predictions. Turkish results are comparatively better, with LR showing a precision of 0.69 and recall of 0.66, suggesting a higher adaptability of the transfer learning approach for the Turkish language. In Korean, the recall is relatively higher for DT and RF at 0.55 compared to precision, which may indicate the model's propensity to over-classify instances as HS. Italian and Spanish show a modest improvement with SVM achieving a precision of 0.61 and 0.57 respectively. These languages demonstrate the SVM's capability to maintain performance despite the shift to a new test language. Portuguese and Indonesian test sets show a moderate performance, with precision and recall metrics indicative of the models' general struggle with accurate classification in these language contexts.

The N-1 strategy's results highlight the necessity for robust feature representation that can handle the linguistic variability inherent in transfer learning scenarios. They also highlight the potential need for additional model fine-tuning or the incorporation of language adaptation techniques to improve performance, particularly for languages with distinct linguistic features that may not be well-represented in the training data. In the next section, we will perform incremental learning with test data augmentation where

From this approach, we noticed that our classifiers excel in certain languages but display suboptimal performance in others. Therefore, we decided to implement incremental learning to further refine our classifiers.

2) INCREMENTAL LEARNING WITH TEST DATA AUGMENTATION

In the second experiment, the model was trained on the same 12 languages, with the addition of a 20% sample from the test language, to observe the impact of including test language data on performance which serves the purpose of incremental learning. This technique was employed to enhance the model's robustness with test data inclusion. Table 10 represents the results using incremental learning. After analyzing results from incremental learning we can observe that the performance of classifiers increased across all languages especially for Chinese and German test sets, the Chinese test set displays modest outcomes, with LR achieving an F1-score and accuracy of 0.57, the highest among the models for this language. The relatively lower scores in Chinese suggest challenges in adapting the learned embeddings to the complex syntax and semantics of Chinese. Similarly, the German test set shows moderate improvements, with F1-scores and accuracy mostly around 0.56, indicating the incremental learning strategy's limited impact on languages with close linguistic similarities to the training set but still posing unique challenges. The Indonesian test set stands out with the highest F1-scores and accuracy, with all models, especially the RF and LR, achieving an F1-score and accuracy of 0.84. This indicates a significant improvement in model performance, highlighting the effectiveness of including test data samples during training for languages with distinct linguistic features. In the English test set, we observe a substantial performance with F1-scores and accuracy consistently at 0.88 across DT, RF, and SVM models, with LR slightly higher at 0.88. This uniformity suggests a robust transferability of learned features across languages with substantial resource availability. The Turkish and Italian test sets also show impressive results, with F1-scores and accuracy reaching up to 0.81 and 0.71, respectively. These scores reflect the models' enhanced ability to generalize from the training to the unseen test language, benefiting from the incremental learning approach. These scores have been significantly increased after performing incremental testing.

3) UNIFIED MULTILINGUAL HS DETECTION FRAMEWORK

For this final experiment, we trained our models on a combined dataset comprising 13 languages, aiming to assess their capability to recognize patterns across this diverse linguistic dataset. This approach seeks to evaluate the models' effectiveness in a comprehensive multilingual context. In table 11 we evaluate how a single model performs when trained on a diverse, multilingual dataset. It reflects a more real-world scenario where platforms receive content in multiple languages and need a unified model. The performance achieved by SVM after regularization and optimization outperformed all other ML and DL-based classifiers by providing 0.99 scores across all four evaluation matrices which highlights that SVM is a very suitable approach to addressing multilingual HS classification tasks when integrated with quantized FastText word embeddings. In terms of DL-based models, CNN-LSTM showed better performance as compared to LSTM and its variant BiLSTM. These results show that supervised FastText is the best technique to address classification problems even if you

TABLE 9. Analysis of the Results: Cross-linguistic N-1 strategy using supervised FastText and ML.

Test Language	Model	P	R	Acc	F
Arabic Test Set	DT	0.54	0.39	0.39	0.39
	RF	0.54	0.40	0.40	0.40
	LR	0.51	0.62	0.62	0.48
	RF	0.54	0.40	0.40	0.40
English Test Set	DT	0.50	0.49	0.49	0.50
	RF	0.50	0.49	0.49	0.49
	LR	0.50	0.49	0.49	0.49
	SVM	0.50	0.49	0.49	0.49
Chinese Test Set	DT	0.54	0.35	0.35	0.31
	RF	0.54	0.35	0.35	0.31
	LR	0.54	0.65	0.65	0.52
	SVM	0.55	0.35	0.35	0.31
French Test Set	DT	0.57	0.57	0.57	0.57
	RF	0.57	0.57	0.57	0.57
	LR	0.57	0.57	0.57	0.57
	SVM	0.57	0.57	0.57	0.57
German Test Set	DT	0.52	0.51	0.51	0.51
	RF	0.52	0.51	0.51	0.51
	LR	0.51	0.51	0.51	0.50
	SVM	0.52	0.51	0.51	0.50
Russian Test Set	DT	0.69	0.42	0.42	0.36
	RF	0.68	0.41	0.41	0.33
	LR	0.48	0.59	0.59	0.51
	SVM	0.48	0.59	0.59	0.51
Turkish Test Set	DT	0.68	0.61	0.61	0.64
	RF	0.68	0.63	0.63	0.65
	LR	0.69	0.66	0.66	0.67
	SVM	0.68	0.64	0.64	0.66
Roman Urdu Test Set	DT	0.53	0.52	0.52	0.52
	RF	0.53	0.52	0.52	0.52
	LR	0.54	0.53	0.53	0.53
	SVM	0.53	0.52	0.52	0.52
Korean Test Set	DT	0.44	0.55	0.55	0.40
	RF	0.45	0.55	0.55	0.41
	LR	0.58	0.45	0.45	0.28
	SVM	0.44	0.55	0.55	0.41
Italian Test Set	DT	0.61	0.61	0.61	0.61
	RF	0.60	0.61	0.61	0.61
	LR	0.60	0.61	0.61	0.60
	SVM	0.61	0.62	0.62	0.61
Spanish Test Set	DT	0.56	0.55	0.55	0.55
	RF	0.57	0.55	0.55	0.56
	LR	0.56	0.55	0.55	0.55
	SVM	0.57	0.55	0.55	0.56
Portugese Test Set	DT	0.59	0.60	0.60	0.60
	RF	0.59	0.61	0.61	0.60
	LR	0.59	0.61	0.61	0.59
	SVM	0.59	0.61	0.61	0.59
Indonesian Test Set	DT	0.52	0.50	0.50	0.51
	RF	0.52	0.50	0.50	0.50
	LR	0.52	0.50	0.50	0.51
	SVM	0.51	0.49	0.49	0.50

are working with highly imbalanced datasets with different languages.

The following figure 3 represents the confusion matrix for the CNN-BiLSTM architecture.

TABLE 10. Analysis of the results: Incremental learning with test data augmentation using supervised FastText and ML.

Test Language	Model	P	R	Acc	F
Arabic Test Set	DT	0.72	0.72	0.72	0.72
	RF	0.73	0.73	0.73	0.73
	LR	0.74	0.74	0.74	0.74
	SVM	0.65	0.63	0.78	0.64
English Test Set	DT	0.88	0.88	0.88	0.88
	RF	0.88	0.88	0.88	0.88
	LR	0.89	0.88	0.88	0.88
	SVM	0.89	0.88	0.88	0.88
Chinese Test Set	DT	0.56	0.38	0.37	0.39
	RF	0.57	0.37	0.37	0.39
	LR	0.60	0.65	0.65	0.57
	SVM	0.55	0.37	0.37	0.38
French Test Set	DT	0.72	0.72	0.72	0.72
	RF	0.72	0.72	0.72	0.72
	LR	0.73	0.73	0.73	0.72
	SVM	0.73	0.72	0.72	0.72
German Test Set	DT	0.56	0.56	0.56	0.56
	RF	0.56	0.57	0.55	0.57
	LR	0.56	0.56	0.56	0.56
	SVM	0.56	0.56	0.56	0.56
Russian Test Set	DT	0.76	0.77	0.77	0.76
	RF	0.77	0.77	0.77	0.77
	LR	0.77	0.77	0.77	0.77
	SVM	0.77	0.77	0.77	0.77
Turkish Test Set	DT	0.81	0.81	0.81	0.81
	RF	0.81	0.82	0.82	0.81
	LR	0.81	0.82	0.82	0.81
	SVM	0.81	0.82	0.82	0.81
Roman Urdu Test Set	DT	0.71	0.71	0.71	0.71
	RF	0.71	0.71	0.71	0.71
	LR	0.72	0.71	0.71	0.71
	SVM	0.71	0.71	0.71	0.71
Korean Test Set	DT	0.61	0.61	0.61	0.61
	RF	0.61	0.61	0.61	0.60
	LR	0.61	0.60	0.60	0.61
	SVM	0.60	0.60	0.60	0.60
Italian Test Set	DT	0.70	0.70	0.70	0.70
	RF	0.71	0.71	0.71	0.71
	LR	0.71	0.71	0.71	0.71
	SVM	0.71	0.71	0.71	0.71
Spanish Test Set	DT	0.69	0.68	0.68	0.69
	RF	0.69	0.68	0.68	0.69
	LR	0.69	0.68	0.68	0.69
	SVM	0.69	0.68	0.68	0.69
Portugese Test Set	DT	0.67	0.66	0.66	0.67
	RF	0.67	0.66	0.66	0.67
	LR	0.67	0.66	0.66	0.67
	SVM	0.67	0.66	0.66	0.67
Indonesian Test Set	DT	0.83	0.83	0.83	0.83
	RF	0.84	0.83	0.83	0.84
	LR	0.84	0.83	0.83	0.84
	SVM	0.84	0.83	0.83	0.83

The following figures 4 and 5 represent the training validation loss and accuracy curves for CNN-BiLSTM architecture. These curves show a consistent convergence on the multilingual HS dataset, with the validation metrics

TABLE 11. Unified multilingual evaluation with FastText.

Model	P	R	Acc	F
DT	0.98	0.98	0.98	0.98
RF	0.98	0.98	0.98	0.98
LR	0.98	0.98	0.99	0.99
SVM	0.99	0.99	0.99	0.99
LSTM	0.90	0.90	0.90	0.90
BiLSTM	0.90	0.90	0.91	0.91
CNN-BiLSTM	0.91	0.91	0.91	0.91

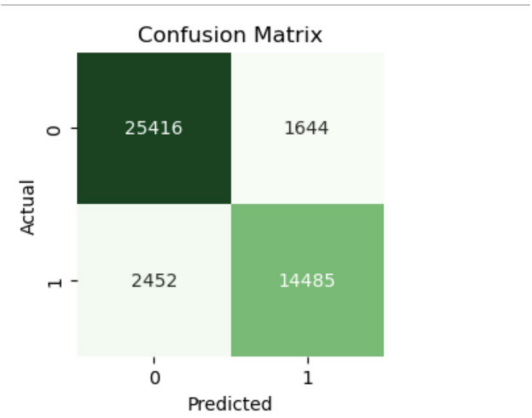

FIGURE 3. CNN-BiLSTM confusion matrix.

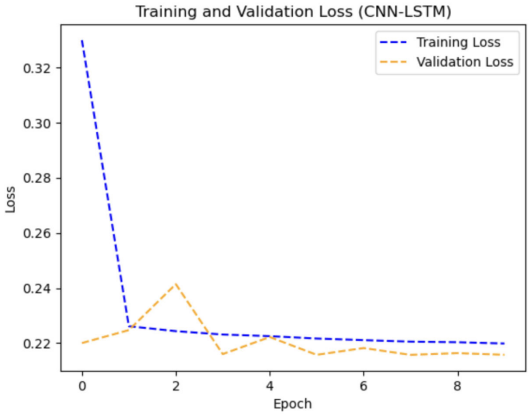

FIGURE 4. CNN-BiLSTM validation loss curve.

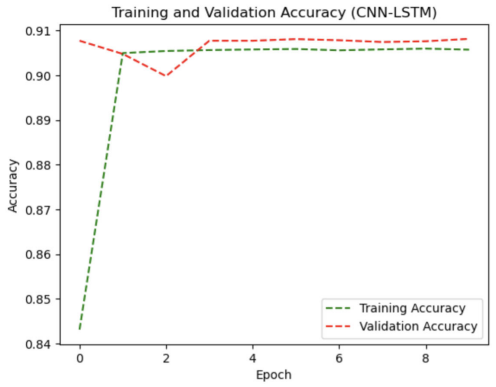

FIGURE 5. CNN-BiLSTM validation accuracy curve.

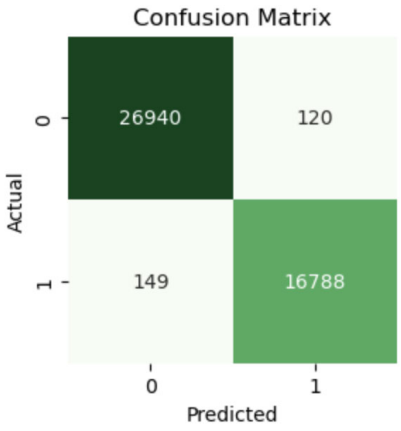

FIGURE 6. SVM confusion matrix.

closely tracking the training metrics during the training period. The close match between training and validation accuracy, along with a steady reduction in loss for both training and validation phases, indicates that the model is successfully learning and does not show evidence of overfitting to the training dataset.

Figure 6 represents the confusion matrix for our best-performing algorithm linear SVM. The confusion matrix shows the performance of a classification model. There are 26,940 true negatives where the model correctly predicted the negative class (0), and 16,788 true positives where the model correctly predicted the positive class (1). However, there are 120 false positives and 149 false negatives, indicating instances where the model incorrectly predicted the positive class and negative class, respectively.

D. UNIFIED MULTILINGUAL HS DETECTION WITH MULTILINGUAL TRANSFORMER BASED MODELS

In this final evaluation, we performed multilingual HS detection using multilingual transformer-based models using hyperparameter and generative configuration. The following table represents the evaluation scores using mBERT and ELECTRA. In Table 12, *mBERT* surpassed both *ELECTRA* and unsupervised FastText embeddings in performance. This was observed not only when using unsupervised FastText

TABLE 12. Unified multilingual evaluation with transformer based models.

Model	P	R	Acc	F
mBERT	**0.93**	**0.92**	**0.92**	**0.92**
ELECTRA	0.83	0.83	0.83	0.83

TABLE 13. Generative configuration for few shot and full fint-tuning.

Model	Top-k	Top-p	Temperature
FLAN-T5	5	0.5	0.3
mBART	7	0.5	0.3

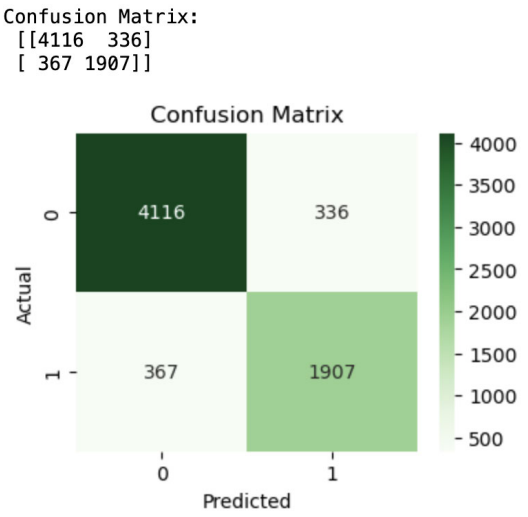

FIGURE 7. GermEval 2018 confusion matrix.

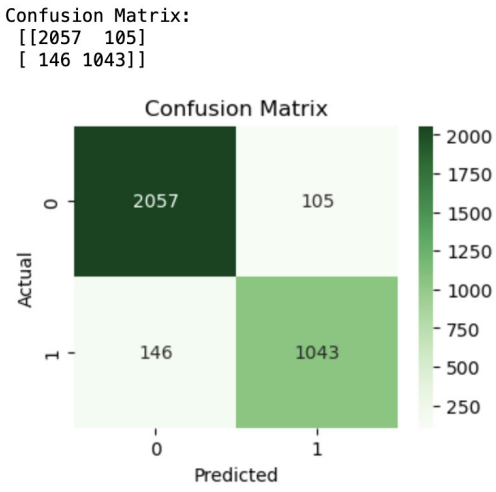

FIGURE 8. GermEval 2021 confusion matrix.

all alone but also when it was combined with language-specific transformer-based models. Furthermore, *mBERT* exceeded the performance metrics of our supervised FastText evaluations conducted using the $N-1$ strategy as well as in incremental learning contexts. The superior performance of *mBERT* can be attributed to its design as a pre-trained model, which has undergone training across 104 languages. It benefits from built-in capabilities for bidirectional text analysis, allowing it to effectively process data across multiple languages simultaneously. This, coupled with its robust training framework, contributes to its outstanding performance. Additionally, *mBERT*'s extensive pretraining on a diverse linguistic dataset provides it with a comprehensive understanding of language nuances, further enhancing its applicability across a broad range of multilingual tasks. The high performance of *mBERT* over *ELECTRA* because it has more number of parameters as compared to *ELECTRA*.

The tables 14 and 15 represent the evaluation scores using few-shot and full fine-tuning respectively. It can be seen that the results obtained from full fine-tuning are slightly better than few-shot fine-tuning, the table 13 represents the generative configuration details used for few-shot and full fine-tuning. Following full fine-tuning, a noticeable enhancement was observed with FLAN-T5, which outperformed mBART across all evaluation matrices

V. COMPARISON WITH THE STATE OF THE ART METHODS

Following the completion of our experiments, we evaluated our hybrid SVM classifier on several benchmark public datasets. This evaluation was part of our broader study to explore the efficacy of incremental learning coupled with test data augmentation, an approach we previously applied to our multilingual dataset. For this phase of assessment, we allocated only 20% of each dataset for training purposes, with the remaining portion utilized for testing, allowing us to closely examine the classifier's performance under these conditions. Table 16 shows the comparison of our work with state-of-the-art on different datasets.

The following figures 7, 8, 9, 10 and 11 represent the confusion matrix on different HS datasets.

VI. INTERPRETABILITY MODELING

Interpretability modeling holds significant importance in HS detection, as it facilitates the development of models or

TABLE 14. Unified multilingual evaluation with few shot fine tuning.

Model	P	R	Acc	F
FLAN-T5	0.85	0.85	0.84	0.84
mBART	**0.86**	**0.86**	**0.86**	**0.86**

TABLE 15. Unified multilingual evaluation with full fine-tuning.

Model	P	R	Acc	F
FLAN-T5	**0.90**	**0.90**	**0.90**	**0.90**
mBART	0.89	0.88	0.88	0.89

TABLE 16. Comparison of the results with the state of the art.

Ref.	Dataset	Baseline Score	Proposed Work
Seemann et al. [78]	German (GermEval-2018)	F1-Score: 0.72	**F1-Score: 0.90**
Khan et al. [79]	Roman Urdu HS	F1-score: 0.91	**F1-Score: 0.95**
Seemann et al. [78]	German (GermEval-2021)	F1-Score: 0.58	**F1-Score: 0.93**
Fersini et al. [56]	Italian and English (AMI 2018)[34]	Acc: 0.59	**Acc: 0.96**
Fan et al. [80],	HateEval 2019	Acc: 0.87	**Acc: 0.95**
Fillies et al. [81]	HateEval 2019	Acc: 0.59	**Acc: 0.95**

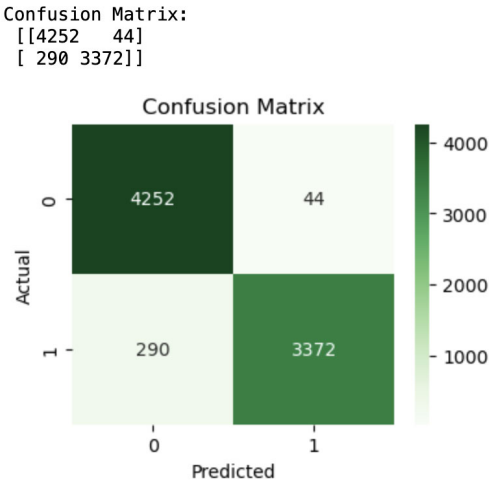

FIGURE 9. AMI 2018 confusion matrix.

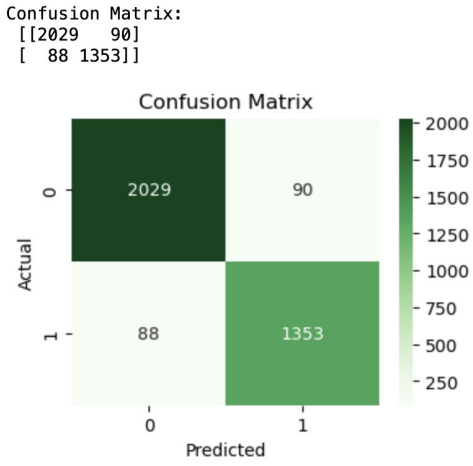

FIGURE 10. HateEval confusion matrix.

techniques that enhance clarity and transparency in complex learning algorithms [66].

A. LIME

LIME is a technique developed to provide a clear understanding and evaluation of the predictions generated by various learning algorithms. It is designed to shed light on the rationale behind a model's decisions, making it especially useful in scenarios where it's crucial to comprehend how a model arrives at its conclusions, beyond just its prediction accuracy [82]. LIME focuses on constructing an interpretable model, denoted as $\hat{g}$, within a specific model class G. This model strives to minimize the discrepancy, or loss $\mathcal{L}$, between its predictions and those of the original, more complex model f. This is done while taking into account a locality kernel π_x that emphasizes the importance of closeness in the data space, and $\Omega(g)$, which measures the complexity of the interpretable model g. The aim is to favor simpler models for their ease of interpretation, ensuring that the explanations remain straightforward and accessible [83].

$$\hat{g} = \arg\min_{g \in G} \mathcal{L}(f, g, \pi_x) + \Omega(g) \tag{11}$$

In this research, we apply LIME to the supervised FastText SVM, which demonstrated the best performance in multilingual HS detection in terms of accuracy, precision, recall, and F1-score. To enhance clarity and provide insight

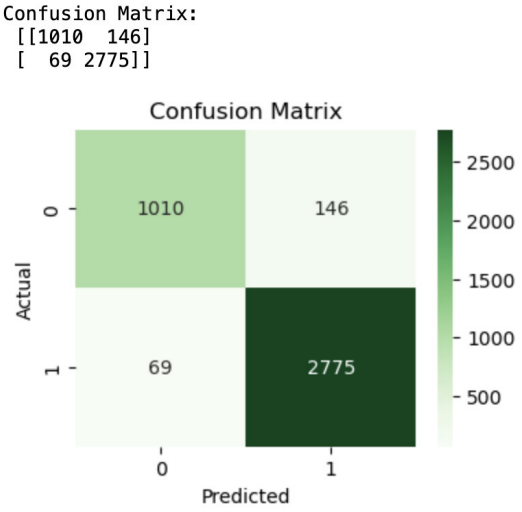

FIGURE 11. Roman Urdu confusion matrix.

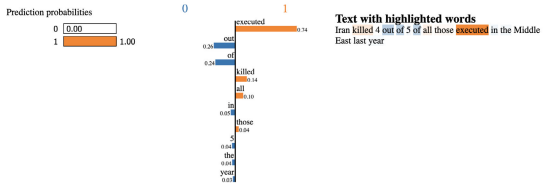

FIGURE 12. English Example 1: Hateful instance visualization with LIME.

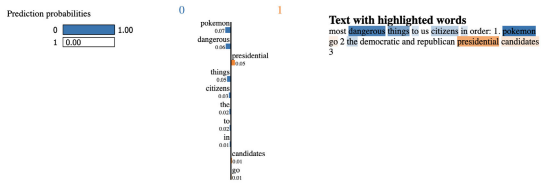

FIGURE 13. English Example 2: Non-hateful instance visualization with LIME.

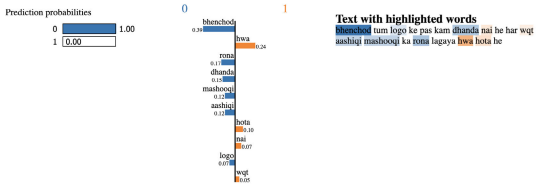

FIGURE 14. Roman Urdu Example 1: Non hateful instance visualization with LIME.

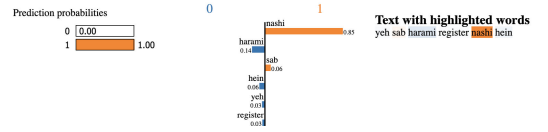

FIGURE 15. Roman Urdu Example 2: Hateful instance visualization with LIME.

into the decision-making process, we included two examples for each language to illustrate our approach and rationale.

In figure 12 the text has been classified as 'hateful', with a probability of 1.00, is primarily influenced by the terms 'executed' and 'killed', which the model has assigned high weights in its analysis. This association with 'hateful' utterances within the model's training data appears to be strong. The inclusion of '4 out of 5' seems to have a less pronounced but still noticeable impact on the classification, pointing to the model's capacity to interpret numerical expressions in conjunction with action words to inform its decision-making process. Figure 13 depicts the text as non-hate speech. The term 'presidential' is given a moderate positive weight, indicating its significant influence in steering the prediction away from HS. Interestingly, 'dangerous', a term that could suggest negativity is not enough to sway the model towards an HS classification, possibly due to its association with the term 'pokemon', which is likely perceived as non-threatening. This combination suggests the model's ability to contextualize potential risk indicators within a broader, non-hostile narrative.

The LIME visualization in 14 shows that the classifier has predicted Urdu text as non-HS with a probability of 1.00 for class ''0''. Despite the presence of the strong phrase, such as ''bhenchod'' (a derogatory term), which typically would contribute to an HS classification, the model has interpreted the overall context of the highlighted words in such a way that it does not deem the text as hateful. This could be due to the model not being sensitive to certain nuances of the language or the specific contextual use of the words. Secondly, this instance has been annotated as neutral as well in the dataset, which might have influenced the model's learning during the training process. In figure 15, the term ''harami'' (which can be translated to 'bastard' or used as an insult in Urdu) has been given significant weight, likely because it is recognized by the model as a strong indicator of HS. The highlighted word ''nashi'' (which could mean intoxication or a state of being high in Urdu, depending on context) is also weighted but seems to contribute less to the classification. This could be due to the model associating it with negative behavior in the context it has been trained on. Overall, the classification suggests the model is sensitive to certain derogatory terms within the dataset it was trained on.

The LIME visualization for figure 16 indicates a prediction leaning towards the 'non-hateful' category with a probability of 1 for class '0'. Given that no individual words are marked with significant weights in the figure, it seems the model's classification is influenced more by the overall context or the combination of words rather than individual terms with strong indicative power. A similar scenario can be observed in figure 17, where the model correctly classified it as hateful but did not assign any weight. In situations where LIME fails to attribute significant weight to any features (words) in its visualization, yet the instance remains classified, the model may be relying on nuanced interactions between features that LIME overlooks. As a local explanation method, LIME may

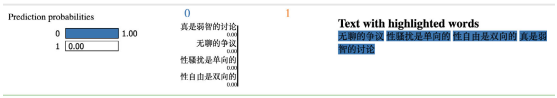

FIGURE 16. Chinese Example 1: Non-hateful instance visualization with LIME.

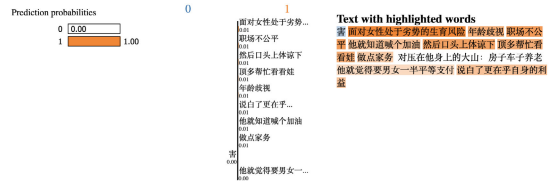

FIGURE 17. Chinese Example 2: Hateful instance visualization with LIME.

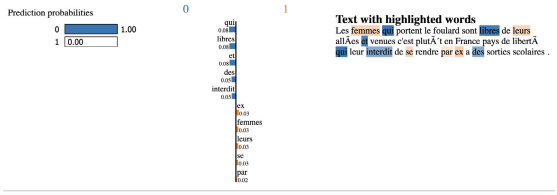

FIGURE 18. French Example 1: Non-hateful instance visualization with LIME.

FIGURE 19. French Example 2: Hateful instance visualization with LIME.

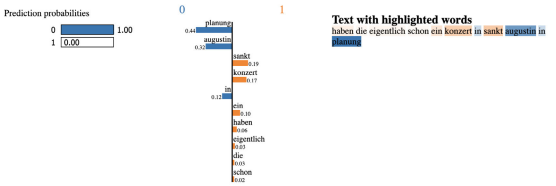

FIGURE 20. German Example 1: Non-hateful instance visualization with LIME.

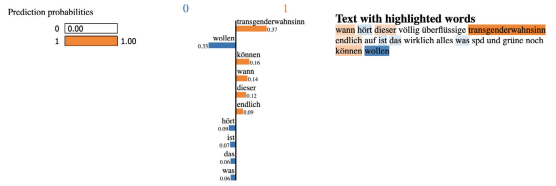

FIGURE 21. German Example 2: Hateful instance visualization with LIME.

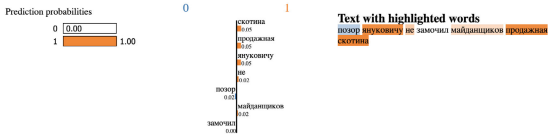

FIGURE 22. Russian Example 1: Hateful instance visualization with LIME.

not consistently capture the global behavior of the model, particularly when the decision boundary is intricate.

In figure 18, LIME visualization shows that our learning algorithm classifies the text as non-hateful. The highlighted terms in the text, 'libres' (free) and 'interdit' (forbidden), are given weights that do not tip the balance toward HS. This suggests that the model interprets the context in which these words are used as non-hateful. The model seems to consider the statement as an expression of a situation rather than as a message carrying hate, as indicated by the high probability of non-hate classification. In figure 19, the model assigns a definite probability of 1.00 to the HS class ''1''. The terms 'mental', 'verratti', and 'attardé', each carrying substantial weight, contribute significantly to this classification. The presence of 'attardé' (a derogatory term) along with 'mental' seems to be central to the model's prediction, suggesting a pejorative context that the model has learned to associate with HS.

The LIME visualization in figure 20 for the German text indicates a classification with a certainty of 1.00 for the non-hate category, class ''0''. The model appears to assign the highest weight to the word 'planung' (planning), followed by 'augustin', and 'sankt' (Saint), suggesting that the context involving these words is associated with non-hateful content. The term 'konzert' (concert) also contributes to this classification but to a lesser extent. The highlighted terms do not inherently carry hateful sentiment, leading the model to predict the text as neutral. In figure 21,the model's prediction for this German instance is categorically in the HS class ''1'' with a probability of 1.00. The term ''transgenderwahnsinn,'' which can be translated as ''transgender madness,'' carries a strong weight and is likely considered derogatory, contributing significantly to the HS classification. Other terms such as ''wollen'' (want) and ''können'' (can) also contribute to this classification, possibly due to their context within the sentence.

In figure 22 the text has been classified as hateful. The highlighted words carry a derogatory statement, and a reference to Yanukovych, possibly the former president of Ukraine carry strong negative connotations and are critical in informing this decision.

In figure 23, model has unequivocally classified the Turkish text as HS, assigning a probability of 1.00 to class ''1''. Key terms contributing to this decision include ''salak'' (stupid or idiot), ''müslüman'' (Muslim), and ''çocuk'' (child), alongside other words that, in this context, appear to be used derogatorily. The combination of these terms, especially with the negative connotations associated with ''salak'' and the sensitive context involving ''çocuk'' and ''müslüman'', leads the model to identify the text as carrying a hateful sentiment. The model has classified the Turkish text in figure 24 as non-hate speech with a confidence of 1.00 for class ''0''. The highlighted words ''yalan'' (lie), ''kalp'' (heart), and ''atar'' (beats/palpitates), are potentially associated with the themes of love and deceit, but within

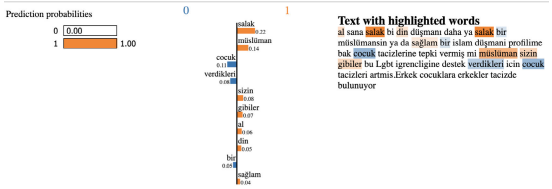

FIGURE 23. Turkish Example 1: Hateful instance visualization with LIME.

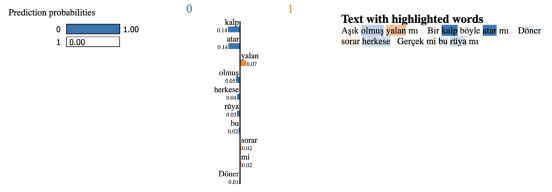

FIGURE 24. Turkish Example 2: Non-hateful instance visualization with LIME.

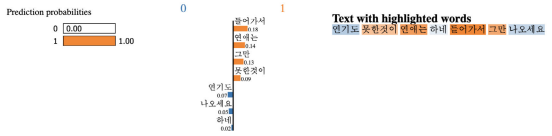

FIGURE 25. Korean Example 1: Hateful instance visualization with LIME.

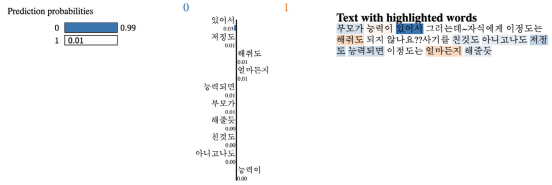

FIGURE 26. Korean Example 2: Non-hateful instance visualization with LIME.

this context, they are not indicative of HS. This suggests that the model is interpreting the text as either metaphorical or pertaining to a different context, such as romance or personal reflection, rather than as an expression of hate.

In the first LIME visualization, Figure 25, the model identifies the Korean text as HS, assigning a probability of 1.00 to this classification. On the other hand, the second LIME visualization, Figure 26, shows a high probability of 0.99 for the text being non-hate speech. In this case, the model does not assign significant weight to any specific term that would suggest an HS classification. The absence of highlighted derogatory terms or phrases leads to the inference that the text is free from language typically associated with HS. This demonstrates the model's contextual sensitivity in distinguishing between texts that contain potentially offensive language and those that do not.

Figure 27 shows a classification of the Italian text as HS. Contributing factors include pejorative terms such as ''pedate'' (kicks) and ''culo'' (ass), which are highlighted as influential in this classification. Their presence, particularly in the given sequence and context, suggests to the model a strong negative or aggressive sentiment. In contrast, Figure 28 presents a more nuanced classification. The model assigns a probability of 0.68 to class ''0'' (non-hate speech), indicating a more moderate stance. Here, the highlighted words include ''vogliono'' (want), ''soluzione'' (solution), and ''diventare''

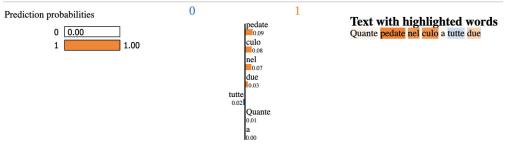

FIGURE 27. Italian Example 1: Hateful instance visualization with LIME.

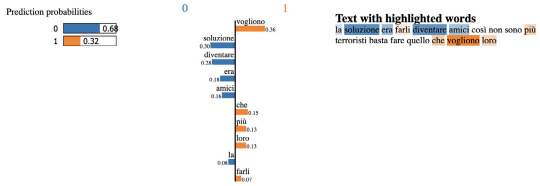

FIGURE 28. Italian Example 2: Non-hateful instance visualization with LIME.

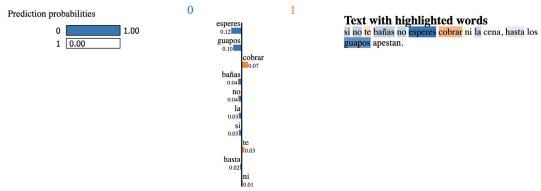

FIGURE 29. Spanish Example 1: Non-hateful instance visualization with LIME.

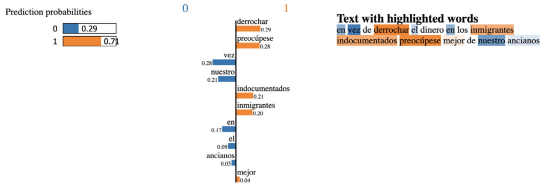

FIGURE 30. Spanish Example 2: Hateful instance visualization with LIME.

(become). The context provided by these terms does not align with HS, leading the model to a non-hateful classification.

For Figure 29, the LIME visualization for Italian text assigns a probability of 1.00 to class ''0'', indicating the text is classified as non-hate speech. The terms highlighted, such as ''esperes'' (you wait) and ''guapos'' (handsome), even with the inclusion of ''culo'' (ass), are interpreted in a context that does not align with hateful utterance, according to the model's analysis. In Figure 30, the model's probability of 0.71 for class ''1'' suggests a classification of HS. The highlighted terms like ''derrochar'' (to waste) and ''preocúpese'' (worry about), in conjunction with ''inmigrantes'' (immigrants), contribute to this classification, indicating the model identifies a negative sentiment within the text, perceiving it as HS.

In Figure 31, the LIME visualization for Portugese instance assigns a probability of 0.59 to class ''0'', suggesting the text is likely non-hate speech. The highlighted words ''feministas'' (feminists) and ''vocês'' (you) carry weights, but not enough to tip the balance toward a hateful utterance. Conversely, Figure 32 shows a LIME visualization with a probability of 0.71 for class ''1'', indicating a tendency towards HS. The terms ''preocúpese'' (worry about), ''nuestro'' (our), and ''ancianos'' (elderly) are highlighted alongside ''inmigrantes'' (immigrants), and the model interprets the overall sentiment as negative, possibly due to the context

FIGURE 31. Portugese Example 1: Hateful instance visualization with LIME.

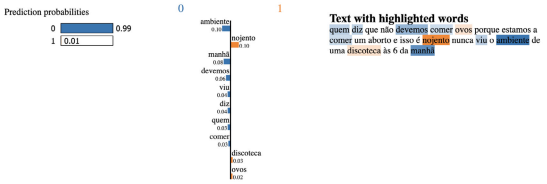

FIGURE 32. Portugese Example 2: Non-hateful instance visualization with LIME.

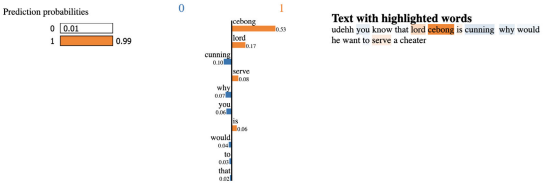

FIGURE 33. Indo Example 1: Hateful instance visualization with LIME.

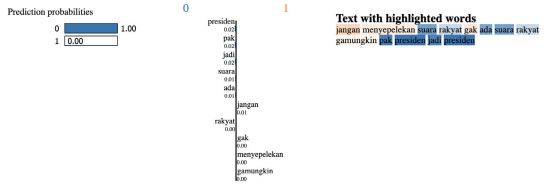

FIGURE 34. Indo Example 2: Non-hateful instance visualization with LIME.

suggesting a dismissive or exclusionary attitude towards immigrants. v

In the LIME visualization for the Indonesian text in figure 33, the model assigns a high probability of 0.99 to class ''1'', which is indicative of HS. The term ''cebong'' is slang, often used derogatorily in political contexts within Indonesia. Combined with ''lord'' and ''cunning'', the model likely interprets this as derogatory or insulting language. The phrase ''he want to serve a cheater'' further contributes to this classification, suggesting malicious intent. In Figure 34, the classification is clear-cut as non-hateful with a probability of 1.00 for class ''0''. The terms ''presiden'' (president) and ''pak'' (mister) are neutral and combined with ''jangan'' (don't) and ''rakyat'' (people) in the context, it's likely discussing political opinion or advice, which the model doesn't identify as HS.

After conducting interpretability modeling with LIME, we discovered that languages with complex structures, such as Chinese and Korean, heavily rely on context and the specific characters within words. Although LIME successfully visualized instances that our learning model correctly classified as hateful or not hateful, it struggled to provide concrete reasons for these classifications. This limitation might stem from LIME's focus on local rather than global representations, which can hinder its ability to fully explain decisions in languages that depend significantly on nuanced contextual and intra-word relationships.

VII. CONCLUSION AND FUTURE WORK

This research advances the field of HS binary classification using a series of different methods. Unsupervised FastText embeddings exhibit slightly better performance as compared to unsupervised FastText all alone. Supervised FastText can be the best selection for classification tasks as it outperformed all ML, DL, and transformer-based methods during the assessment of a unified multilingual HS detection scenario. The mBERT model proved to be particularly suited for multilingual HS detection tasks, demonstrating robust across-the-board metrics and outperforming other multilingual transformer-based models. Transformers with a larger number of parameters can perform better than those with fewer parameters because of their ability to capture more contextual information. Full fine-tuning can exhibit better performance as compared to few-shot fine-tuning. The interpretability modeling with LIME revealed its limitations in providing clear explanations for classifications in languages characterized by complex structures, such as German, Chinese, and Korean. This is attributed to LIME's local perspective, which may not fully capture the nuanced contextual dependencies integral to these languages. Our findings demonstrate a marked enhancement in performance across diverse datasets. Notably, the proposed model achieved F1-scores of 0.90 in German (GermEval-2018) compared to the baseline of 0.72, 0.93 in German (GermEval-2021) up from 0.58, and 0.95 in Roman Urdu HS, improving upon the previous high of 0.91. Additionally, for mixed-language datasets like Italian and English (AMI 2018), accuracy was significantly increased to 0.96 from a baseline of 0.59. These results underscore our model's robustness and adaptability, setting a new benchmark for HS detection systems across varied linguistic contexts. In the future, we intend to enhance our research to encompass multiclass and multilabel text classification for resource-scarce languages, utilizing advanced LLMs like Llama, GPT, and mT5 transformers. Our objective is to address the challenges of data scarcity and enhance model versatility across linguistic varieties, with a dedicated focus on elevating HS detection capabilities for a broader array of underrepresented languages. The future work will also involve the implementation and discussion of other XAI algorithms such as SHAP. This direction aims to navigate the intricacies of multilingual text analysis, ensuring a more inclusive and effective approach to mitigate online HS.

DECLARATIONS

1) COMPETING INTERESTS
   - The authors declare that they have no competing interests.
2) AUTHORS' CONTRIBUTION
   - **Ehtesham Hashmi:** Conceptualization, data analysis, formal analysis, research execution, design

of methods, resources, software, writing original draft, investigation.

- **Sule Yildirim Yayilgan:** Visualization, supervision, project management, funding acquisition, validation.
- **Ibrahim A. Hameed:** Visualization, supervision, validation, formal analysis.
- **Muhammad Mudassar Yamin:** Visualization, research conduct, validation.
- **Mohib Ullah:** Visualization, validation, formal analysis.
- **Mohamed Abomhara:** Supervision, project management, funding acquisition, validation.

3) ETHICAL AND INFORMED CONSENT FOR DATA USED

- Not Applicable.

4) DATA AVAILABILITY AND ACCESS

- The datasets employed in this research are publicly accessible, and their respective URLs are provided in the ''Dataset'' subsection under the ''Methodology'' section of this article.

REFERENCES

[1] K. Thapliyal, M. Thapliyal, and D. Thapliyal, ''Social media and health communication: A review of advantages, challenges, and best practices,'' *Emerg. Technol. Health Literacy Med. Pract.*, vol. 1, pp. 364–384, Jul. 2024.

[2] E. Hashmi, M. M. Yamin, and S. Y. Yayilgan, ''Securing tomorrow: A comprehensive survey on the synergy of artificial intelligence and information security,'' *AI Ethics*, vol. 1, pp. 1–19, Jul. 2024.

[3] F. Mehmood, H. Ghafoor, M. N. Asim, M. U. Ghani, W. Mahmood, and A. Dengel, ''Passion-net: A robust precise and explainable predictor for hate speech detection in Roman Urdu text,'' *Neural Comput. Appl.*, vol. 36, no. 6, pp. 3077–3100, Feb. 2024.

[4] F. Poletto, V. Basile, M. Sanguinetti, C. Bosco, and V. Patti, ''Resources and benchmark corpora for hate speech detection: A systematic review,'' *Lang. Resour. Eval.*, vol. 55, no. 2, pp. 477–523, Jun. 2021.

[5] I. Gagliardone, D. Gal, T. Alves, and G. Martinez, *Countering Online Hate Speech*. de Fontenoy, Paris: Unesco Publishing, 2015.

[6] K. Ghosh, A. Senapati, M. Narzary, and M. Brahma, ''Hate speech detection in low-resource bodo and assamese texts with ML-DL and BERT models,'' *Scalable Computing, Pract. Exper.*, vol. 24, no. 4, pp. 941–955, Nov. 2023.

[7] J. Risch, *Toxicity*, vol. 12, 2023, pp. 219–230.

[8] J. Papcunová, M. Martoncik, D. Fedáková, M. Kentos, M. Bozogánová, I. Srba, R. Moro, M. Pikuliak, M. Šimko, and M. Adamkovic, ''Hate speech operationalization: A preliminary examination of hate speech indicators and their structure,'' *Complex Intell. Syst.*, vol. 9, no. 3, pp. 2827–2842, Jun. 2023.

[9] J. Groshek and C. Cutino, ''Meaner on mobile: Incivility and impoliteness in communicating online,'' in *Proc. 7th Int. Conf. Social Media Soc.*, 2016, pp. 1–7.

[10] 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.*, 2021, pp. 457–468.

[11] T. U. Haque, N. N. Saber, and F. M. Shah, ''Sentiment analysis on large scale Amazon product reviews,'' in *Proc. IEEE Int. Conf. Innov. Res. Develop. (ICIRD)*, May 2018, pp. 1–6.

[12] S. A. Aljuhani and N. Saleh, ''A comparison of sentiment analysis methods on Amazon reviews of mobile phones,'' *Int. J. Adv. Comput. Sci. Appl.*, vol. 10, no. 6, pp. 1–66, 2019.

[13] C. Meske and E. Bunde, ''Design principles for user interfaces in AI-based decision support systems: The case of explainable hate speech detection,'' *Inf. Syst. Frontiers*, vol. 1, pp. 743–773, Mar. 2022.

[14] P. Saha, K. Garimella, N. K. Kalyan, S. K. Pandey, P. M. Meher, B. Mathew, and A. Mukherjee, ''On the rise of fear speech in online social media,'' *Proc. Nat. Acad. Sci. USA*, vol. 120, no. 11, Mar. 2023, Art. no. 2212270120.

[15] M. R. Awal, R. K. Lee, E. Tanwar, T. Garg, and T. Chakraborty, ''Model-agnostic meta-learning for multilingual hate speech detection,'' *IEEE Trans. Computat. Social Syst.*, vol. 1, no. 1, pp. 1–10, May 2023.

[16] S. Akuma, T. Lubem, and I. T. Adom, ''Comparing bag of words and TF-IDF with different models for hate speech detection from live tweets,'' *Int. J. Inf. Technol.*, vol. 14, no. 7, pp. 3629–3635, Dec. 2022.

[17] H. EL-Zayady, M. S. Mohamed, K. Badran, and G. Salama, ''A hybrid approach based on personality traits for hate speech detection in Arabic social media,'' *Int. J. Electr. Comput. Eng.*, vol. 13, no. 2, p. 1979, Apr. 2023.

[18] D. Mittal and H. Singh, ''Enhancing hate speech detection through explainable AI,'' in *Proc. 3rd Int. Conf. Smart Data Intell. (ICSMDI)*, Mar. 2023, pp. 118–123.

[19] S. Agarwal, A. Sonawane, and C. R. Chowdary, ''Accelerating automatic hate speech detection using parallelized ensemble learning models,'' *Expert Syst. Appl.*, vol. 230, Nov. 2023, Art. no. 120564.

[20] A. Toktarova, D. Syrlybay, B. Myrzakhmetova, G. Anuarbekova, G. Rakhimbayeva, B. Zhylanbaeva, N. Suieuova, and M. Kerimbekov, ''Hate speech detection in social networks using machine learning and deep learning methods,'' *Int. J. Adv. Comput. Sci. Appl.*, vol. 14, no. 5, pp. 1–26, 2023.

[21] H. Saleh, A. Alhothali, and K. Moria, ''Detection of hate speech using BERT and hate speech word embedding with deep model,'' *Appl. Artif. Intell.*, vol. 37, no. 1, Dec. 2023, Art. no. 2166719.

[22] 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.

[23] Z. Waseem, ''Are you a racist or am I seeing things? Annotator influence on hate speech detection on Twitter,'' in *Proc. 1st Workshop NLP Comput. Social Sci.*, 2016, pp. 138–142.

[24] 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.

[25] S. J. Johnson, M. R. Murty, and I. Navakanth, ''A detailed review on word embedding techniques with emphasis on word2vec,'' *Multimedia Tools Appl.*, vol. 83, no. 13, pp. 37979–38007, Oct. 2023.

[26] D. S. Asudani, N. K. Nagwani, and P. Singh, ''Impact of word embedding models on text analytics in deep learning environment: A review,'' *Artif. Intell. Rev.*, vol. 56, no. 9, pp. 10345–10425, Sep. 2023.

[27] F. Sufi, ''Generative pre-trained transformer (GPT) in research: A systematic review on data augmentation,'' *Information*, vol. 15, no. 2, p. 99, Feb. 2024.

[28] G. Yenduri, M. Ramalingam, G. C. Selvi, Y. Supriya, G. Srivastava, P. K. R. Maddikunta, G. D. Raj, R. H. Jhaveri, B. Prabadevi, W. Wang, A. V. Vasilakos, and T. R. Gadekallu, ''GPT (generative pre-trained transformer)—A comprehensive review on enabling technologies, potential applications, emerging challenges, and future directions,'' *IEEE Access*, vol. 12, pp. 54608–54649, 2024.

[29] A. B. De Oliveira, C. de Souza Baptista, A. A. Firmino, and A. C. De Paiva, ''A large language model approach to detect hate speech in political discourse using multiple language corpora,'' *Proc. 39th ACM/SIGAPP Symp. Appl. Comput.*, 2024, pp. 1461–1468.

[30] A. A. Firmino, C. de Souza Baptista, and A. C. de Paiva, ''Improving hate speech detection using cross-lingual learning,'' *Expert Syst. Appl.*, vol. 235, Jan. 2024, Art. no. 121115.

[31] M. S. Khan, M. S. I. Malik, and A. Nadeem, ''Detection of violence incitation expressions in Urdu tweets using convolutional neural network,'' *Expert Syst. Appl.*, vol. 245, Jul. 2024, Art. no. 123174.

[32] A. Svetasheva and K. Lee, ''Harnessing large language models for effective and efficient hate speech detection,'' Tech. Rep., 2024.

[33] J. A. García-Díaz, S. M. Jiménez-Zafra, M. A. García-Cumbreras, and R. Valencia-García, ''Evaluating feature combination strategies for hate-speech detection in Spanish using linguistic features and transformers,'' *Complex Intell. Syst.*, vol. 9, no. 3, pp. 2893–2914, Jun. 2023.

[34] J. de la Rosa, E. G. Ponferrada, P. Villegas, P. Gonzalez de Prado Salas, M. Romero, and M. Grandury, ''BERTIN: Efficient pre-training of a Spanish language model using perplexity sampling,'' 2022, *arXiv:2207.06814*.

[35] R. Pan, J. A. García-Díaz, F. Garcia-Sanchez, and R. Valencia-García, "Evaluation of transformer models for financial targeted sentiment analysis in Spanish," *PeerJ Comput. Sci.*, vol. 9, p. e1377, May 2023.

[36] S. Vassileva, G. Grazhdanski, S. Boytcheva, and I. Koychev, "Fusion@BioASQ MedProcNER: Transformer-based approach for procedure recognition and linking in Spanish clinical text," in *Proc. Work. Notes CLEF*, 2023, pp. 190–205.

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

[38] S. Nagar, F. A. Barbhuiya, and K. Dey, "Towards more robust hate speech detection: Using social context and user data," *Social Netw. Anal. Mining*, vol. 13, no. 1, p. 47, Mar. 2023.

[39] 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*, vol. 12, 2018, pp. 1–26.

[40] A. M. Founta, D. Chatzakou, N. Kourtellis, J. Blackburn, A. Vakali, and I. Leontiadis, "A unified deep learning architecture for abuse detection," in *Proc. 10th ACM Conf. Web Sci.*, Jun. 2019, pp. 105–114.

[41] J. Pérez, M. Castro, E. Awad, and G. López, "Generation of probabilistic synthetic data for serious games: A case study on cyberbullying," *Knowl.-Based Syst.*, vol. 286, Feb. 2024, Art. no. 111440.

[42] C. Ziems, W. Held, O. Shaikh, J. Chen, Z. Zhang, and D. Yang, "Can large language models transform computational social science?" 2023, *arXiv:2305.03514*.

[43] L. Khan, A. Amjad, K. M. Afaq, and H.-T. Chang, "Deep sentiment analysis using CNN-LSTM architecture of English and Roman Urdu text shared in social media," *Appl. Sci.*, vol. 12, no. 5, p. 2694, Mar. 2022.

[44] A. A. Nagra, K. Alissa, T. M. Ghazal, S. Kukunuru, M. M. Asif, and M. Fawad, "Deep sentiments analysis for Roman Urdu dataset using faster recurrent convolutional neural network model," *Appl. Artif. Intell.*, vol. 36, no. 1, Dec. 2022, Art. no. 2123094.

[45] S. Chen, J. Wang, and K. He, "Chinese cyberbullying detection using XLNet and deep bi-LSTM hybrid model," *Information*, vol. 15, no. 2, p. 93, Feb. 2024.

[46] Y. Sun, S. Wang, Y. Li, S. Feng, X. Chen, H. Zhang, X. Tian, D. Zhu, H. Tian, and H. Wu, "ERNIE: Enhanced representation through knowledge integration," 2019, *arXiv:1904.09223*.

[47] E. Mahajan, H. Mahajan, and S. Kumar, "Ensmulhatecyb: Multilingual hate speech and cyberbully detection in online social media," *Expert Syst. Appl.*, vol. 236, Jun. 2024, Art. no. 121228.

[48] A. Kumar, S. Vignesh Murthy, S. Singh, and S. Ragupathy, "The ethics of interaction: Mitigating security threats in LLMs," 2024, *arXiv:2401.12273*.

[49] L. Yan, L. Sha, L. Zhao, Y. Li, R. Martinez-Maldonado, G. Chen, X. Li, Y. Jin, and D. Gasević, "Practical and ethical challenges of large language models in education: A systematic scoping review," *Brit. J. Educ. Technol.*, vol. 55, no. 1, pp. 90–112, Jan. 2024.

[50] I. de Zarza, J. de Curto, G. Roig, P. Manzoni, and C. T. Calafate, "Emergent cooperation and strategy adaptation in multi-agent systems: An extended coevolutionary theory with LLMs," *Electronics*, vol. 12, no. 12, p. 2722, Jun. 2023.

[51] E. Hashmi and S. Y. Yayilgan, "Multi-class hate speech detection in the Norwegian language using FAST-RNN and multilingual fine-tuned transformers," *Complex Intell. Syst.*, vol. 10, no. 3, pp. 4535–4556, Jun. 2024.

[52] I. Bigoulaeva, V. Hangya, I. Gurevych, and A. Fraser, "Label modification and bootstrapping for zero-shot cross-lingual hate speech detection," *Lang. Resour. Eval.*, vol. 57, no. 4, pp. 1515–1546, Dec. 2023.

[53] J. C. Pereira-Kohatsu, L. Quijano-Sánchez, F. Liberatore, and M. Camacho-Collados, "Detecting and monitoring hate speech in Twitter," *Sensors*, vol. 19, no. 21, p. 4654, Oct. 2019.

[54] F. E. Ayo, O. Folorunso, F. T. Ibharalu, I. A. Osinuga, and A. Abayomi-Alli, "A probabilistic clustering model for hate speech classification in Twitter," *Expert Syst. Appl.*, vol. 173, Jul. 2021, Art. no. 114762.

[55] J. A. García-Díaz, M. Cánovas-García, R. Colomo-Palacios, and R. Valencia-García, "Detecting misogyny in Spanish tweets. An approach based on linguistics features and word embeddings," *Future Gener. Comput. Syst.*, vol. 114, pp. 506–518, Jan. 2021.

[56] E. Fersini, D. Nozza, and P. Rosso, "Overview of the evalita 2018 task on automatic misogyny identification (AMI)," in *CEUR Workshop Proc.*, vol. 2263, 2018, pp. 1–9.

[57] G. D. Valle-Cano, L. Quijano-Sánchez, F. Liberatore, and J. Gómez, "SocialHaterBERT: A dichotomous approach for automatically detecting hate speech on Twitter through textual analysis and user profiles," *Expert Syst. Appl.*, vol. 216, Apr. 2023, Art. no. 119446.

[58] H. A. Batarfi, O. A. Alsaedi, A. M. Wali, and A. T. Jamal, "Impact of data augmentation on hate speech detection," in *Proc. Int. Conf. Innov. Community Services*, 2023, pp. 187–199.

[59] M. Ptaszynski, A. Pieciukiewicz, and P. Dybała, "Results of the poleval 2019 shared task 6: First dataset and open shared task for automatic cyberbullying detection in Polish Twitter," Tech. Rep., 2019.

[60] D. Trajano, R. H. Bordini, and R. Vieira, "OLID-BR: Offensive language identification dataset for Brazilian Portuguese," *Lang. Resour. Eval.*, vol. 1, pp. 1–27, May 2023.

[61] R. S. Satpute and A. Agrawal, "A critical study of pragmatic ambiguity detection in natural language requirements," *Int. J. Intell. Syst. Appl. Eng.*, vol. 11, no. 3s, pp. 249–259, 2023.

[62] Y. Chen, A. Huang, Z. Wang, I. Antonoglou, J. Schrittwieser, D. Silver, and N. de Freitas, "Bayesian optimization in Alphago," 2018, *arXiv:1812.06855*.

[63] Y. Yu, X. Si, C. Hu, and J. Zhang, "A review of recurrent neural networks: LSTM cells and network architectures," *Neural Comput.*, vol. 31, no. 7, pp. 1235–1270, Jul. 2019.

[64] B. Jang, M. Kim, G. Harerimana, S.-U. Kang, and J. W. Kim, "Bi-LSTM model to increase accuracy in text classification: Combining Word2vec CNN and attention mechanism," *Appl. Sci.*, vol. 10, no. 17, p. 5841, Aug. 2020.

[65] H. Ali, E. Hashmi, S. Yayilgan Yildirim, and S. Shaikh, "Analyzing Amazon products sentiment: A comparative study of machine and deep learning, and transformer-based techniques," *Electronics*, vol. 13, no. 7, p. 1305, Mar. 2024.

[66] E. Hashmi, S. Y. Yayilgan, M. M. Yamin, S. Ali, and M. Abomhara, "Advancing fake news detection: Hybrid deep learning with FastText and explainable AI," *IEEE Access*, vol. 12, pp. 44462–44480, 2024.

[67] B. Sabiri, B. El Asri, and M. Rhanoui, "Mechanism of overfitting avoidance techniques for training deep neural networks," in *Proc. 24th Int. Conf. Enterprise Inf. Syst.*, 2022, pp. 418–427.

[68] S. Kumar, H. Marklund, and B. Van Roy, "Maintaining plasticity in continual learning via regenerative regularization," 2023, *arXiv:2308.11958*.

[69] A. Vaswani, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2023, pp. 1–11.

[70] K. Clark, M.-T. Luong, Q. V. Le, and C. D. Manning, "Electra: Pre-training text encoders as discriminators rather than generators," 2020, *arXiv:2003.10555*.

[71] E. Hashmi, S. Y. Yayilgan, and S. Shaikh, "Augmenting sentiment prediction capabilities for code-mixed tweets with multilingual transformers," *Social Netw. Anal. Mining*, vol. 14, no. 1, p. 86, Apr. 2024.

[72] C. Raffel, "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.

[73] H. W. Chung, "Scaling instruction-finetuned language models," 2022, *arXiv:2210.11416*.

[74] Y. Liu, J. Gu, N. Goyal, X. Li, S. Edunov, M. Ghazvininejad, M. Lewis, and L. Zettlemoyer, "Multilingual denoising pre-training for neural machine translation," *Trans. Assoc. Comput. Linguistics*, vol. 8, pp. 726–742, Dec. 2020.

[75] M. Platt and D. Platt, "Effectiveness of generative artificial intelligence for scientific content analysis," in *Proc. IEEE 17th Int. Conf. Appl. Inf. Commun. Technol.*, Oct. 2023, pp. 1–26.

[76] M. M. Yamin, E. Hashmi, M. Ullah, and B. Katt, "Applications of LLMs for generating cyber security exercise scenarios," Tech. Rep., 2024.

[77] D. Chen, J. Hu, X. Wei, and E. Wu, "Real3D: The curious case of neural scene degeneration," *Proc. AAAI Conf. Artif. Intell.*, vol. 38, no. 2, pp. 1028–1036, Mar. 2024.

[78] N. Seemann, Y. S. Lee, J. Höllig, and M. Geierhos, "Generalizability of abusive language detection models on homogeneous German datasets," *Datenbank-Spektrum*, vol. 23, no. 1, pp. 15–25, Mar. 2023.

[79] 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, Jan. 2021.

[80] X. Fan, J. Liu, J. Liu, P. Tuerxun, W. Deng, and W. Li, "Identifying hate speech through syntax dependency graph convolution and sentiment knowledge transfer," *IEEE Access*, vol. 12, pp. 2730–2741, 2024.
[81] J. Fillies, M. P. Hoffmann, and A. Paschke, "Multilingual hate speech detection: Comparison of transfer learning methods to classify German, Italian, and Spanish posts," in *Proc. IEEE Int. Conf. Big Data*, Dec. 2023, pp. 5503–5511.
[82] P. Biecek and T. Burzykowski, "Local interpretable model-agnostic explanations (lime)," *Explanatory Model Anal. Explore, Explain Examine Predictive Models*, vol. 1, pp. 107–124, Jun. 2021.
[83] E. Hashmi, M. Mudassar Yamin, S. Imran, S. Y. Yayilgan, and M. Ullah, "Enhancing misogyny detection in bilingual texts using FastText and explainable AI," in *Proc. Int. Conf. Eng. Comput. Technol. (ICECT)*, May 2024, pp. 1–6.

EHTESHAM HASHMI received the B.S. degree in computer science from the University of Central Punjab, Lahore Campus, in 2020, and the M.S. degree in computer science from COMSATS University Islamabad, Lahore Campus, in 2022. Currently, he is pursuing the Ph.D. degree with the Department of Information Security and Communication Technology (IIK), Norwegian University of Science and Technology (NTNU). From 2022 to 2023, he was a Lecturer with the Department of Computer Science, University of Lahore. His research interests include multilingual natural language processing, computational linguistics, large language models, knowledge graphs, and data mining.

SULE YILDIRIM YAYILGAN received the M.Sc. degree in computer engineering, in 1995, and the Ph.D. degree in artificial intelligence and computer science, in 2002. She has been with the Department of Information Security and Communication Technology (IIK), NTNU, since 2009. She has worked for more than 25 years in academic teaching. She has been supervising students at different academic levels and has been publishing more than 100 journal and conference papers. Her research interests include image processing, information security, natural language processing, computational linguistics, large language models, and data mining.

IBRAHIM A. HAMEED (Senior Member, IEEE) received the B.Sc. and M.Sc. degrees in control engineering from Menofia University, Egypt, the Ph.D. degree in industrial systems and information engineering from Korea University, in 2010, and the Ph.D. degree in mechanical engineering from Aarhus University, in 2012. He has been the Department Deputy Head of Research and Innovation, from 2018 to 2022. Since 2018, he has led the international master's program in simulation and visualization and founded the Social Robots Laboratory, NTNU, Ålesund. He has been actively involved in promoting gender equality and diversity at his faculty, since 2021. He is currently a Full Professor. His work experience includes roles as an Associate Professor with Aalesund University College and Aalborg University, where he also conducted postdoctoral research. His teaching repertoire includes courses on control systems, intelligent systems, and advanced topics in artificial intelligence, with a current focus on machine learning and deep learning. His research interests include artificial intelligence, machine learning, control engineering, and social robots. He has authored more than 200 publications in these fields. He has been a member of ACM, since 2021. He was elected as the Chair of the IEEE Computational Intelligence Society Norway Chapter, in 2019. He is an Associate Editor of IEEE TRANSACTIONS ON ARTIFICIAL INTELLIGENCE.

MUHAMMAD MUDASSAR YAMIN is currently an Associate Professor with the Department of Information and Communication Technology, Norwegian University of Science and Technology (NTNU). He is a member with the System Security Research Group. Before joining NTNU, he was an Information Security Consultant and served multiple government and private clients. His research interests include system security, penetration testing, security assessment, and intrusion detection. He holds multiple cybersecurity certifications, such as OSCE, OSCP, LPT-MASTER, CEH, CHFI, CPTE, CISSO, and CBP. He is serving as a Reviewer for Nature.

MOHIB ULLAH is currently a Researcher with NTNU, where he is involved in different research, management, teaching, and industrial projects. His research interests include medical imaging, crowd analysis, object segmentation, behavior classification, and tracking. In these research areas, he has published several high-impact peer-reviewed journals, conferences, and workshop articles. He served as a Program Committee Member for the International Workshop on Computer Vision in Sports (CVsports). He also served as the Chair for the Technical Program of European Workshop on Visual Information Processing. He is a Reviewer of well-reputed conferences and journals (*Neurocomputing* (Elsevier), *Neural Computing and Applications* (Elsevier), *Multimedia Tools and Applications* (Spring), IEEE ACCESS, *Journal of Imaging*, IEEE CVPRw, IEEE ICIP, and IEEE AVSS).

MOHAMED ABOMHARA received the bachelor's degree in Libya, in 2006, the master's degree (M.Sc.) in computer science in Malaysia, in 2011, the master's degree (M.B.A.) in business administration, in 2013, and the Ph.D. degree in information technology from the University of Agder, Norway, in 2018. He is currently a Cybersecurity Researcher and a Data Protection Specialist with the Department of Information Security and Communication Technology, Norwegian University of Science and Technology (NTNU). His scholarly contributions extend to active participation in several prestigious European, Erasmus+, and Norwegian Research Council Projects, where he has assumed both technical and managerial roles and has published multiple journal and conference papers. His commitment to advancing technology while upholding ethical and privacy standards underscores his prominent role in academia and research. His primary research interests include the development of data-driven technologies that uphold critical principles, such as transparency, accountability, and privacy.