Spectrum of Engineering Sciences
ISSN (e) 3007-3138 (p) 3007-312X Volume 3, Issue 5, 2025

FROM DETECTION TO PRECISION: ELEVATING HATE SPEECH CLASSIFICATION WITH CUTTING-EDGE MODELS

Nimra Maqbool[*1], Aftab Anjum[2], Muhammad Zunnurain Hussain[3], Tehreem Aslam[4], Asad Yaseen[5], Muhammad Zulkifl Hasan[6]

*[*1]BS Computer Engineering, Information Technology University Lahore, Pakistan.*
[2]PHD Scholar, Keil University, Germany.
[3]Assistant Professor, Dept. of Computer Science Bahria University Lahore, Pakistan.
[4]Senior Solution Analyst AnalytIQ, Lahore, Pakistan.
[5]Senior Solutions Architect at STC SOLUTIONS.
[6]Researcher Cyber Security Center, Universitie Putra Malaysia.

[1]bsce21012@itu.edu.pk, [2]anjumaftab.cs@gmail.com, [3]zunnurain.bulc@bahria.edu.pk, [4]tehreem.aslam36@gmail.com, [5]asad4ntrp2@gmail.com, [6]gs58279@student.upm.edu.my

DOI: https://doi.org/10.5281/zenodo.15348997

Keywords
Hate speech detection, Hate speech classification, Text classification, Automated hate speech analysis, BERT, Social media analysis
Article History
Received on 28 March 2025
Accepted on 28 April 2025
Published on 06 May 2025

Copyright @Author
Corresponding Author: *

Abstract
In the digital era, the proliferation of hate speech on social media platforms has necessitated the development of effective detection systems. This paper presents a comprehensive comparative analysis of machine learning and deep learning approaches for hate speech classification across diverse datasets, including a thorough comparison with existing methodologies. Specifically, this study evaluates the performance of two machine learning models Random Forest and XGBoost and two deep learning models, LSTM and BERT. Each model is trained using various embeddings, including Word2Vec, as well as GloVe, supplemented by TF-IDF for the machine learning models. Through rigorous cross-validation and hyperparameter tuning, the efficacy of each model and embedding combination is assessed. The results are analyzed not only to determine the most effective approach for hate speech detection but also to benchmark these results against previous studies in the field. This comparative analysis provides insights into the strengths and limitations of the models and embeddings used, aiming to contribute to the ongoing efforts in creating a safer online environment by advancing the state-of-the-art in hate speech detection.

INTRODUCTION

In the contemporary digital landscape, social media platforms have become integral to daily interactions (Chetty, N. 2018) , particularly among young users (Peng, S. 2018), (Castaño-Pulgarín, S. A. 2021) Unfortunately, this increased connectivity has also witnessed a concerning rise in hate speech (Keipi, T 2022), (Wachs, S. 2022). This toxic form of communication not only disrupts social harmony but also poses a significant threat to individuals' mental well-being (Wypych, M. 2024), (Saha, K. 2019). Notably, during the pandemic, the BBC (Baggs, M. 2021). reported a 20% surge in online hate speech, while a study commissioned by the youth charity Ditch the Label revealed a staggering 50.1 million instances of racist hate speech in the UK and US between 2019 and mid-2021.

In response to this growing concern, In this esearch project we aims to address this challenge head-on. We

curate a diverse set of datasets, spanning binary, multiclass, and multilingual classifications, to train and evaluate advanced machine learning and deep learning models. Our model selection includes Random Forest and XGBoost, known for their predictive power, as well as Long Short-Term Memory (LSTM) networks, celebrated for their ability to process sequential data. Additionally, we incorporate BERT (Bidirectional Encoder Representations from Transformers), a state-of-the-art language model, to enhance our understanding of language nuances.

To enhance the model's understanding of language nuances, we have employed various embeddings, and for machine learning models, the traditional TF-IDF. The implementation of our approach includes cross-validation, a careful selection of hyperparameters for having high performance. Furthermore, the study includes a comparison with the previous literature conducted between the years 2017 and 2024. Thus, the goal of this paper is to evaluate the purpose of using different approaches to modeling in the identification of hate speech in certain types of datasets, highlighting the advantages and disadvantages of each approach on the basis of the results obtained in comparison.

The structure of this paper is as follows: Section 2 offers an extensive review of the existing literature. Section 3 details the methodology, outlining the proposed general framework. Section 4 provides an overview of the system and describes the model training process. In Section 5, we discuss the data encoding and experimental settings, including the datasets used for evaluation. Section 6 presents the results and analysis based on the datasets. Section 7 focuses on the discussion of the graphs obtained during the experiments. Section 8 covers the software and hardware configuration utilized. Finally, Section 9 concludes the paper by summarizing the findings, exploring their implications, and proposing future research directions.

2. Literature Review

Hate speech detection on social media has garnered significant attention due to its societal impact, yet challenges persist across languages, datasets, and methodologies. While early efforts primarily focused on English, tailored approaches have emerged to address language-specific nuances. For instance, ABMM (Almaliki, M. 2023) introduces a BERT-based model optimized for Arabic, achieving remarkable accuracy 98.6% on Twitter data through a three-class classification framework, emphasizing the importance of adapting transformer models to complex languages like Arabic. Similarly, a comparative analysis (Narayan, N. 2023) across Indo-Aryan languages, including Bengali and Gujarati, underscores the variability in performance of multilingual models such as BERT and XLM-R, highlighting the necessity for task-specific adaptations. Complementing these, interpretable AI techniques have also been explored, as seen in a study (Shakil, M. H. 2022) leveraging CNNs with NLP pipelines on multilingual datasets, showcasing robust performance across language barriers. Moreover, Indian languages have drawn attention, with research (Roy, P. K. 2022) utilizing region-specific corpora and context-aware deep learning strategies, demonstrating the significance of localized models for complex linguistic settings.

Efforts in low-resource settings have also made strides, as demonstrated by an ensemble approach (Anusha 2020) that combines traditional features like TF-IDF with Gradient Boosting and XGBoost classifiers, achieving success in English, German, and Hindi datasets. This highlights the importance of resource-adaptive methods for multilingual hate speech detection. However, overlap between hate speech and offensive language remains a persistent challenge. A recent study (Davidson 2017) addressed this by employing a crowd-sourced hate speech lexicon and training multi-class classifiers capable of distinguishing nuanced categories, revealing that racist and homophobic content is more consistently classified as hate speech compared to sexist remarks.

Despite these advancements, challenges in dataset reliability and method consistency remain. A comprehensive review (Alkomah, F. 2022) that most datasets are small and lack diversity, limiting their effectiveness in capturing the multifaceted nature of hate speech. Deep learning models, often employing hybrid techniques, dominate the field but exhibit performance variability across hate speech categories, underscoring the need for robust datasets and

standardized evaluation criteria. Moreover, model brittleness persists; state-of-the-art systems perform well only on datasets with similar structures and are highly vulnerable to adversarial attacks, such as typos and altered word boundaries (Gröndahl, T. 2018) Character-level features have shown promise in enhancing robustness, outperforming word-level features under adversarial conditions.

To address these complexities, recent approaches integrate contextual and user-based features. For instance, a novel framework (Nagar, S. 2023) leverages a Variational Graph Auto-encoder to jointly model social context, user metadata, and textual features, significantly enhancing detection accuracy on Twitter datasets. This adaptable method outperforms text-only approaches and demonstrates the potential of incorporating social and contextual data into hate speech detection systems. Complementing these efforts, distributed low-dimensional embeddings (Djuric, N. 2015) mitigate issues of high dimensionality and sparsity, improving both efficiency and accuracy.

Collectively, these advancements reflect the evolving landscape of hate speech detection, emphasizing the need for language-sensitive, context-aware, and adversarially robust approaches to safeguard online discourse effectively.

3. Methodology

In our study, we implemented a comprehensive approach to the challenge of hate speech detection by employing two machine learning models—Random Forest and XGBoost—alongside advanced deep learning architectures, including Long Short-Term Memory (LSTM) networks and Bidirectional Encoder Representations from Transformers (BERT).

4. System overview & Models Training

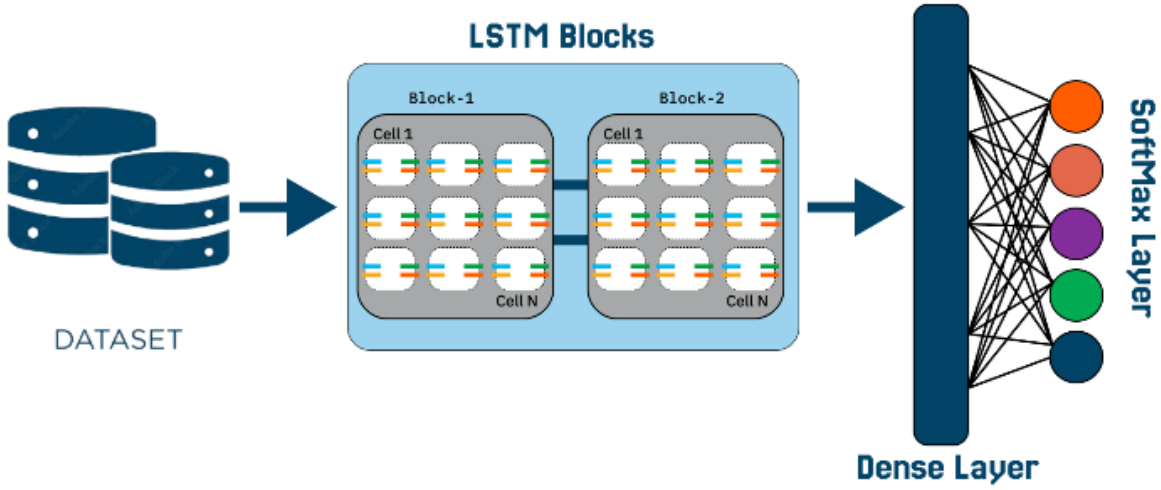
In our research on hate speech detection, we explored a hybrid system that combines machine learning and deep learning techniques to classify text data effectively. Initially, we preprocessed the textual data by applying common text cleaning techniques such as lowercasing, stopword removal, and lemmatization. For the machine learning component, we utilized both Random Forest and XGBoost classifiers, which were trained using TF-IDF vectorization and pre-trained word embeddings. A comprehensive cross-validation process was conducted to determine the optimal number of trees in the ensemble, with iterations over a range of estimators from 50 to 350. This step ensured that the models were not overfitting while achieving optimal performance. Additionally, hyperparameter optimization was carried out using RandomizedSearchCV, which sampled from a grid of potential parameters and selected the best configuration based on accuracy scores.

For evaluation, we used key metrics such as accuracy, F1 score, precision, and recall to assess the models' effectiveness in detecting hate speech. The models were trained with the best-found parameters to maximize their performance. In parallel, we incorporated a deep learning approach using an LSTM network, which was enhanced with pre-trained GloVe embeddings to capture semantic word meanings. To prevent overfitting, early stopping was implemented during the training process. Our system also integrates BERT for both multilingual and monolingual classification tasks, addressing binary as well as multiclass classification problems. This diverse architecture, which combines multiple model types and feature representations, provides a robust solution for detecting the subtle nuances of hate speech in textual data.

4.2LSTM

The Long Short-Term Memory (Hochreiter, S. 1997) (LSTM) model, a type of Recurrent Neural Network (RNN) (Werbos, P. J. 1990) is particularly effective for sequence-based tasks, where the order and context of data points significantly influence predictions. This makes it an ideal choice for text classification tasks like hate speech detection, where the sequence and formation of words carry critical meaning. In our approach, we utilized an LSTM architecture for multi-class classification, beginning with a non-trainable embedding layer that maps words to 300-dimensional vectors via a pre-trained embedding matrix, enabling the capture of semantic relationships. The LSTM layer, consisting of 64 units, processes the sequence data, effectively capturing long-term dependencies and contextual nuances. To combat overfitting, a dropout layer was added, followed by two dense layers with ReLU and softmax activations for multi-class

output, and sigmoid for binary classification. The model was optimized using the Adam optimizer with categorical cross-entropy loss, and early stopping was employed to halt training when performance stagnated, ensuring a robust and generalizable model.



4.2 XGBoost

XGBoost (Chen, T. 2016) is an advanced implementation of the gradient boosting framework, known for its efficiency and robust performance in model training. As an ensemble learning method, it builds a series of decision trees where each new tree corrects the errors of its predecessor, improving prediction accuracy through iterative refinement. This sequential error-correction mechanism makes XGBoost highly effective in capturing complex patterns, particularly in imbalanced datasets, common in hate speech text classification. We utilized TF-IDF vectorization to convert textual data into numerical features, enhancing the model's ability to identify relevant patterns. Additionally, pre-trained word embeddings were incorporated to capture deeper linguistic nuances. The model was rigorously evaluated using metrics like accuracy, precision, recall, and F1 score, ensuring its effectiveness in classifying hate speech while maintaining a balanced trade-off between false positives and false negatives.

4.3 Random Forest

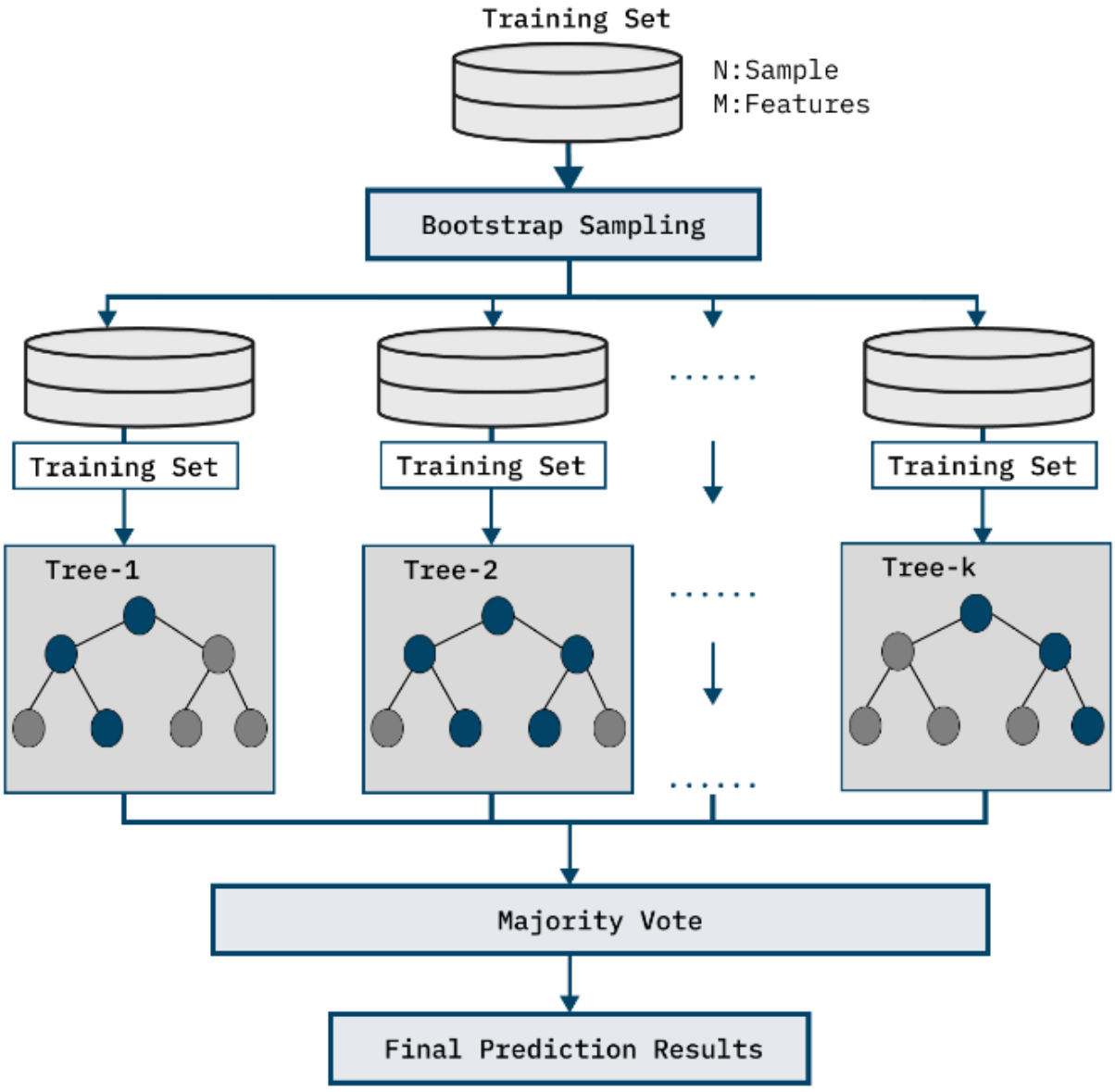
nThe Random Forest model (Breiman, L. 2001) is a powerful and widely used ensemble learning technique that constructs a multitude of decision trees during its training phase and predicts the class based on the majority vote of the individual trees. As an ensemble method, it leverages the wisdom of many trees, which significantly improves the model's robustness and accuracy. Random Forest is particularly well-suited for high-dimensional data sets, allowing it to handle large datasets with numerous features efficiently. Its capability to assess variable importance provides valuable insights into which features contribute most to the classification process, making it an effective tool for both predictive modeling and feature selection.

To mitigate the risk of overfitting and ensure the robustness of our evaluation metrics, we carefully pre-processed the data by dividing it into training and testing sets. This separation ensures that the model's performance is evaluated on unseen data, providing a more accurate measure of its generalization ability. In order to represent the text data effectively, we employed the Term Frequency-Inverse Document Frequency (TF-IDF) vectorization method. TF-IDF quantifies the significance of words in a document relative to their occurrence across a corpus, which helps the model focus on the most important features for classification. By transforming the raw text into a numeric representation, TF-IDF serves as a crucial input to the Random Forest classifier, enabling it to process high-dimensional textual data.

Furthermore, to enhance the semantic richness of our text data, we utilized pre-trained word embeddings, which capture the contextual meaning of words in a vector space.

of the model, which integrates these techniques, is shown in Figure. Further illustrating the comprehensive approach to addressing the classification task.

4.4 BERT



These embeddings provide a deeper understanding of the relationships between words, enriching the feature representation for the model. The Random Forest classifier, trained on these enriched embeddings, was able to synthesize insights from the decision trees more effectively, resulting in improved accuracy and robustness in classifying hate speech. The architecture

BERT (Devlin, J. 2019) (Bidirectional Encoder Representations from Transformers) is a state-of-the-art model developed by Google that utilizes bidirectional embeddings for improved content comprehension. Unlike traditional models, BERT is trained to consider both the forward and backward context of a sentence, making it highly effective for

tasks requiring deep semantic understanding. This bidirectional approach has been shown to enhance performance in various NLP tasks, including hate speech detection, where capturing subtle linguistic patterns is essential for distinguishing harmful content. BERT's pre-trained nature allows it to leverage extensive language knowledge, which can be fine-tuned to specific tasks such as text classification.

For hate speech detection, our BERT architecture adapts the pre-trained model by adding a classification layer on top of the contextualized embeddings generated by BERT. This layer maps the output to the appropriate classes, using softmax for multi-class classification and sigmoid for binary classification. During fine-tuning, the model adjusts its parameters to optimize for the specific task, with the option to train all layers or only the top layers based on computational resources and dataset size. To prevent overfitting and maintain model integrity, only the best-performing epochs are saved, while underperforming epochs are discarded. The model's effectiveness is evaluated using metrics such as accuracy, precision, recall, and F1 score.

For further understanding, we have included a detailed description of each model used in our hate speech detection system in the following section. This includes a brief overview of the machine learning models, such as Random Forest and XGBoost, along with the deep learning models like the LSTM network and BERT. For each model, we provide essential information about their roles and functionality within our system, helping the reader gain a clearer insight into how these models were integrated and optimized to enhance the performance of our classification task.

5. DATA ENCODING & EXPERIMENTS SETTING

In this section, we describe the dataset used in this study, along with the techniques for data encoding and representation. We also discuss the data splitting strategy implemented for training and evaluation, followed by the embedding methods employed for feature enhancement. Additionally, we provide an overview of the system architecture and configuration, as well as the software and hardware setup used in the experiments.

5.1 Datasets

For the purpose of this study, the dataset has been strategically divided into three distinct categories to explore the complexity of hate speech detection across different contexts. The first category includes binary classification datasets, where the task involves distinguishing between hate speech and non-hate speech. The second category focuses on multi-class classification, where the dataset contains multiple categories of hate speech, such as abusive, offensive, or targeted speech. The third category involves multi-label classification, where each instance may belong to more than one class, reflecting the nuanced nature of hate speech. Additionally, a multilingual dataset was incorporated to explore hate speech detection across different languages, addressing challenges related to linguistic diversity and cultural context.

5.2 Binary Classification Datasets

The first category focuses on the task of binary classification, where the provided datasets are used to differentiate between hate speech and non-hate speech. These datasets serve as a foundation for training and evaluating models designed to accurately classify text into these two categories.

Dataset-I: The dataset introduced by Thomas Mandl et al. (Mandl, T. 2019) comprises 7,005 samples, with 2,549 hate speech instances and 4,456 non-hate speech instances. This dataset exhibits a notable class imbalance, where non-hate speech samples constitute approximately 63% of the total, creating challenges for model training. Despite this imbalance, the dataset offers a valuable resource for developing and testing classification models aimed at distinguishing between harmful and non-harmful content in text.

Dataset-II:

The dataset derived from the work of Valerio Basile et al. (Basile, V. 2019) contains 10,000 samples, with 4,210 hate speech instances and 5,790 non-hate speech instances. This dataset is relatively more balanced compared to others, with hate speech instances accounting for approximately 42% of the total. Although there is still a slight imbalance, the distribution is conducive to training models that can generalize well to real-world applications, where class distributions are often skewed.

Dataset-III:
The Dynamically Generated Hate Dataset introduced by Vidgen et al. (Vidgen, B 2022) consists of 41,255 samples, with 22,262 hate speech instances and 18,993 non-hate speech instances. While this dataset is larger than the previous two, it still exhibits a mild imbalance, with hate speech instances comprising approximately 54% of the total, and non-hate speech instances making up 46%. The dataset provides a substantial sample size, which is beneficial for training deep learning models, and its relatively balanced distribution allows for robust evaluation of classification models.
The distribution of the dataset across the different classes is shown in Table I.

Dataset	Total Samples	Hate Samples	Non- Hate Samples
Mandl	7,005	2,549	4,456
Basile	10,000	4,210	5,790
Dynamic	41,255	22,262	18,993

TABLE-I: Dataset samples distributions for Binary classification

5.3 Multi-class Classification Datasets
In addition to binary classification, our study extends to multi-class classification to better capture the varied and nuanced nature of hate speech. Multi-class classification enables the differentiation between distinct categories of hate speech, such as offensive, profane, and other forms of harmful content. This approach allows for a more detailed analysis of the diverse types of hate speech that may exist in textual data, providing insights into the specific characteristics and severity of harmful language.

Dataset-I:
The Mandl dataset is used for multi-class classification and contains a total of 7,005 samples. It is organized into two broad categories: non-hate and hate. The non-hate category includes 4,456 samples, while the hate category is further subdivided into three distinct classes: hate, profane, and offensive language. This multi-class structure allows for a more granular analysis of hate speech, enabling the model to distinguish between various forms of harmful content. The distribution of each class within the dataset is shown in Table \ref{tab:dataset-distribution-c shown shown in Table \ref{tab:dataset-distribution-Multiclass}, providing a clear overview of the dataset's composition. Multiclass}, providing a clear overview of the dataset's composition.

Dataset-II:
The Dynamically Generated Hate Dataset is employed for both binary and multi-class classification, as it includes hierarchical information ranging from coarse-grained categories to fine-grained subclasses. The dataset consists of 41,255 total samples, with 18,993 non-hate instances. The hate category is divided into six subcategories: Derogation, Not-given, Animosity, Threatening, Dehumanization, and Support. This hierarchical structure provides an opportunity to explore not only the broad presence of hate speech but also the specific types and intensities of harmful language present in the dataset.

Dataset-III:
The Davidson dataset, introduced by Thomas Davidson et al. (Davidson, T. 2017) contributes to the multi-class classification framework with 24,783 samples, of which 4,163 are non-hate samples. The hate category is divided into two main classes: 19,190 samples are classified as offensive language, while 1,430 samples are categorized as hate speech. This distinction allows the model to identify and differentiate between more subtle forms of offensive language and more explicit instances of hate speech, further refining the analysis of harmful content in text.

5.4 Multi-lingual Classification Datasets

To extend our experiment to multilingual hate speech detection, we utilized two distinct datasets, one in English and one in German. This approach allows us to assess the model's performance across multiple linguistic contexts, thereby addressing the challenges posed by language diversity in hate speech detection.

Dataset-I:

As reported by Mandl et al., the English dataset consists of 7,005 samples and is specifically designed for the task of detecting hate speech in English-language content. This dataset contains a wide variety of text samples, encompassing different forms and expressions of hate speech, which are crucial for training models capable of handling the complexities of language and context of language and context of language and context expressions of hate speech, which are crucial for training models capable of handling the complexities of language and Context.

Dataset	Total Samples	Class	Class Distribution
Mandl	7,005	None	4,456
		Hate	1,267
		Profane	760
		Offensive	522
Dynamic	41,255	None	18,993
		Derogation	9,907
		Not-given	7,197
		Animosity	3,439
		Threatening	606
		Dehumanization	906
		Support	207
Davidson	24,783	Neither	4,163
		Offensive	19,190
		Hate	1,430

TABLE II: Distribution of Data in Multi-class

Dataset-II:

In addition to the English dataset, the German dataset (Risch, J. 2021) contains 4,670 samples and is focused on detecting hate speech in German-language content. By including this dataset, our study ensures that the model is tested in both English and German linguistic environments, addressing potential language shifts and the challenges inherent in detecting hate speech across different languages. This bilingual approach strengthens the model's applicability and effectiveness in diverse linguistic contexts.

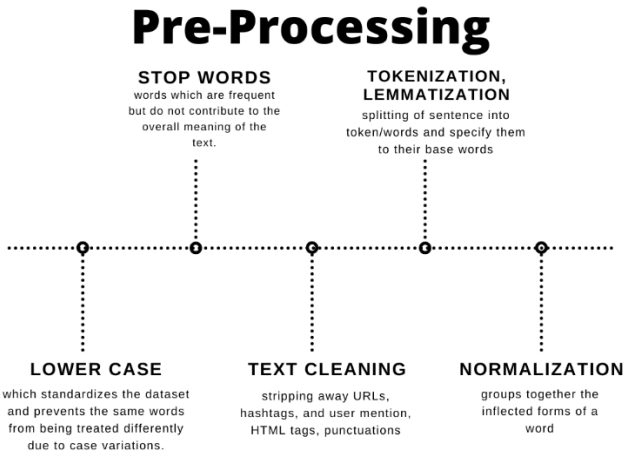
5.5 Data Encoding and Representation

Data pre-processing:

The preparation of text data is a crucial step in training machine learning models, particularly for hate speech detection, as it ensures the data is clean, consistent, and focused on meaningful information. The first stage of pre-processing involves converting all text to lowercase, standardizing the case and eliminating inconsistencies. Stop words, which are frequent but carry little significance, are then removed to reduce noise and highlight more important words. Irrelevant elements such as URLs, hashtags, and user mentions are discarded using regular expressions, as they do not contribute to the sentiment analysis. HTML tags are also stripped out to focus solely on textual content, and punctuation marks are discarded

due to their lack of informative value. Further, tokenization breaks the text into individual words, and lemmatization reduces words to their base forms, ensuring that similar words are grouped together. Custom regular expressions are applied to eliminate filler words, followed by trimming extra whitespace to ensure a clean corpus for feature extraction.

Once the data is pre-processed, it is divided into training and testing sets to evaluate model performance. For this study, 80% of the dataset was used for training the model, while the remaining 20% was reserved for testing and validation. This split ensures that the model is exposed to a substantial amount of data for learning, while also providing an unbiased evaluation of its generalization capability. These pre-processing and data splitting steps are essential in optimizing the machine learning model's ability to detect hate speech accurately, ensuring that the features used for training are relevant and the model can be effectively evaluated on unseen data. The flow of these pre-processing steps is illustrated in Figure Below.



5.5 Embeddings

In our research on hate speech detection, we leveraged two widely used word embedding models, GloVe (Global Vectors for Word Representation) and Word2Vec, to capture the semantic properties of words and enhance the classification process. GloVe constructs word vectors by aggregating a word-word co-occurrence matrix, which encapsulates both the probability of word occurrences and their relationships with surrounding words. We utilized both the 300-dimensional (300d) and 50-dimensional (50d) versions of GloVe, where the higher-dimensional vectors provide a richer context and more detailed representation of word meanings, albeit at the cost of increased computational demands. On the other hand, Word2Vec employs neural networks—specifically the Continuous Bag of Words (CBOW) and Skip-Gram models—to generate distributed word representations based on their context within the corpus. The 300d version of Word2Vec provides a deep semantic understanding of words by learning from their contextual usage.

To transform raw text into a format interpretable by machine learning models, we also utilized Term Frequency-Inverse Document Frequency (TF-IDF) vectorization. TF-IDF measures the importance of words within a document relative to the entire corpus, highlighting key terms while diminishing the weight of less informative ones. However, TF-IDF was not integrated with the LSTM model in our study due to its ability to capture sequential context inherently. Instead, the pre-trained GloVe and Word2Vec embeddings, which inherently encode contextual information, were employed to represent text. These embeddings played a pivotal role in facilitating effective hate speech classification by providing machine learning models with rich, numerical word representations that capture both word-level meaning and broader contextual relationships essential for accurate detection.

5.6 Software And Hardware Configuration

The project leveraged the computational resources provided by Google Colab, utilizing the NVIDIA Tesla T4 GPU, which is well-suited for deep learning tasks due to its high performance and efficiency. On the software front, We utilized Pandas and NumPy for data manipulation, gensim and nltk for text processing, and scikit-learn's CountVectorizer and TfidfVectorizer for feature extraction. For the LSTM model, we incorporated Keras, which runs on TensorFlow's backend. These libraries were chosen for their widespread use in the research community,

extensive documentation, and active support, making them a reliable choice.

6. Results

The table \ref{TABLE:Binary Accuracy} and \ref{TABLE:Multiclass_Accuracy_RF_XGboost} and \ref{TABLE: Multilingual Accuracy} showcases the results of binary, multi-class and Multilingual classification tasks performed using Random Forest, XGBoost models, and BERT respectively. It includes the best results obtained either from the initial phase or after hyperparameter optimization. The table highlights the Accuracy and Macro F1 Score of the models, emphasizing the importance of feature representation methods such as Embeddings and TF-IDF in improving predictive accuracy.

6.1 Binary Classification

In the binary classification task, the models show distinct strengths and weaknesses across the Basile, Mandl, and Dynamic datasets, as outlined in Table III shown below.

TABLE III: Accuracy and F1-Score for Binary Classification

Methods	Techniques	Basile		Mandl		Dynamic	
		Accuracy	F1-Macro	Accuracy	F1-Macro	Accuracy	F1-Macro
RF	TF-IDF	74.00	74.13	67.23	54.23	61.39	61.00
	Glove (300)	73.45	72.21	68.16	57.16	66.24	65.03
	Glove (50)	73.35	71.58	67.34	58.00	66.00	64.03
	Word2vec (300)	73.45	71.06	67.88	57.55	62.00	60.00
Xgboost	TF-IDF	78.00	76.50	69.51	63.00	69.00	68.48
	Glove (300)	75.00	74.00	67.02	60.61	67.35	65.53
	Glove (50)	74.00	72.41	65.00	60.00	62.00	61.00
	Word2vec (300)	75.20	74.00	68.00	60.00	65.00	64.10
LSTM	Glove (300)	75.75	75.80	67.06	66.55	70.84	70.34
	Glove (50)	74.24	74.42	66.88	64.97	69.76	69.33
	Word2vec (300)	75.65	75.77	**69.00**	**68.10**	71.00	70.24
BERT	Bert embedding	**78.00**	**78.00**	68.00	67.00	**72.00**	**72.40**

BERT consistently outperforms all models, with the highest accuracy of 78.00% and an F1-macro of 78.00% on the Basile dataset. This superior performance can be attributed to BERT's ability to capture contextual relationships between words, enabling it to excel in tasks requiring deep understanding of language nuances, as highlighted (Devlin, J. 2022). On the Mandl dataset, BERT again leads with an accuracy of 68.00%, but its F1-macro score of 67.00% reflects the challenge of imbalanced data, an issue BERT struggles with despite its high accuracy. This points to the need for class weighting or fine-tuning to address such imbalances.

XGBoost demonstrates a solid performance, outperforming Random Forest (RF) across all datasets, especially with TF-IDF embeddings. It achieves an accuracy of 78.00% and F1-macro of 76.54% on the Basile dataset, outperforming RF, which reaches 74.00% accuracy and 74.13% F1-macro. XGBoost's gradient-boosting framework enables it to better capture complex feature interactions, which is particularly advantageous in handling large datasets like Dynamic, where XGBoost's ability to model class imbalances through regularization (Chen, T. 2016) gives it an edge over RF. This is especially evident with XGBoost's accuracy of 69.00% and F1-macro of 68.48% on the Dynamic dataset, compared to RF's 61.39% accuracy and 61.00% F1-macro.

Random Forest, while performing well with TF-IDF embeddings, especially in terms of F1-macro, struggles with dense embeddings like GloVe and Word2Vec, where its accuracy and F1-macro scores drop. The challenge is due to RF's inability to fully capture the semantic relationships encoded in these embeddings, which affect its ability to generalize, as noted (Gupta, S. 2022). Additionally, class imbalance in datasets like Mandl further hinders RF's generalization capabilities, as it tends to favor majority classes.

LSTM performs adequately in some cases but falls behind BERT, particularly on smaller datasets like Basile, where overfitting becomes a problem. LSTM's F1-macro score of 75.75% on Basile highlights this challenge. However, on Dynamic, LSTM improves, achieving an accuracy of 71.00% and F1-macro of 70.24%. This improvement aligns with LSTM's ability to leverage sequential relationships in text, especially in larger datasets.

6.2 Multi-class Classification

In the multi-class classification task, the performance of models varies across the Mandl, Dynamic, and Davidson datasets, as shown in Table

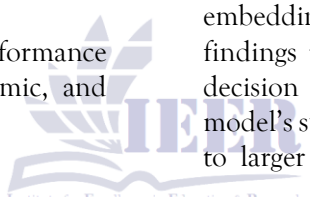


TABLE IV: Model Accuracy and F1-Score for Multi-class classification

Methods	Techniques	Mandl		Dynamic		Davidson	
		Accuracy	F1-Macro	Accuracy	F1-Macro	Accuracy	F1-Macro
RF	TF-IDF	49.40	44.69	58.23	56.23	85.30	64.96
	Glove (300)	42.45	52.16	56.16	55.16	84.78	64.03
	Glove (50)	42.35	51.35	56.34	55.64	83.81	62.03
	Word2vec (300)	41.45	51.04	59.88	58.55	86.81	57.03
Xgboost	TF-IDF	57.60	54.54	58.51	49.84	90.68	68.48
	Glove (300)	**59.55**	**50.41**	73.02	72.61	87.35	63.53
	Glove (50)	58.85	43.81	74.81	73.82	85.81	69.05
	Word2vec (300)	48.20	47.96	**75.88**	**73.00**	89.81	66.10
LSTM	Glove (300)	44.83	43.30	62.62	68.67	88.90	87.60
	Glove (50)	42.22	40.00	61.97	58.12	88.50	87.46
	Word2vec (300)	50.32	48.90	63.02	59.33	**89.59**	**87.91**

LSTM achieves the best results, particularly on the Davidson dataset with Word2Vec (300) embeddings, obtaining the highest accuracy of 75.65% and F1-macro of 75.77%. This highlights LSTM's ability to capture sequential dependencies and contextual nuances effectively (Hochreiter, S.1997). On Dynamic, LSTM performs well with GloVe (300), achieving 63.02% accuracy and 59.33% F1-macro, showcasing its adaptability to large datasets. However, its performance on Mandl, where accuracy drops to 50.32% with Word2Vec (300), indicates sensitivity to smaller datasets with uneven class distributions.

XGBoost shows strong performance across all datasets, particularly with TF-IDF embeddings. It achieves an accuracy of 69.51% and F1-macro of 54.23% on the Mandl dataset and on Davidson, reaching 69.00% accuracy and 68.48% F1-macro. This highlights XGBoost's ability to handle feature interactions effectively and its robustness to class imbalances, as demonstrated by its use of scale_pos_weight and regularization techniques(Chen, T. 2016), (Florek, P. 2023) Its performance improves with TF-IDF over dense embeddings like GloVe and Word2Vec, supporting findings that sparse representations work better for decision tree-based models (Lian, J. 2023). The model's success in Davidson reflects its ability to adapt to larger and more balanced datasets, making it a strong contender in multi-class tasks.

Random Forest (RF) performs moderately across all datasets, achieving the highest F1-macro of 64.96% on Davidson with TF-IDF embeddings. However, RF's performance is hindered by its inability to capture the semantic relationships present in dense embeddings like GloVe and Word2Vec (Tomita, T. M. 2020). For instance, it reaches only 42.45% accuracy and 52.16% F1-macro on Mandl with GloVe-300 embeddings. This can be attributed to RF's reliance on discrete, sparse features, which makes it less effective with dense representations, as noted . Additionally, its limitations in handling class imbalance (He, H. 2009) further reduce its effectiveness, as seen in the lower F1-macro scores on Dynamic (56.23%) and Mandl (44.69%). Lastly, the risk of overfitting increases with high-dimensional, dense data, where RF may focus too heavily on noisy features (Louppe, G. 2014).

6.2 Multi-lingual Classification

In multilingual sentiment classification, BERT demonstrates strong performance across both binary and multiclass tasks, as shown in Table V given below.

TABLE V: Accuracy and F1-Score for Multi-lingual Classification

Methods	Data	Multilingual Dataset	
		Accuracy	F1-Macro
Bert	Binary	**78.82**	72.31
	Multi-class	**86.67**	79.13

In the binary classification setting, BERT achieves an accuracy of 78.82% and an F1-macro score of 72.31%. For the multiclass setting, its accuracy improves slightly to 88.67%, with a marginal increase in F1-macro to 79.13%. This suggests that while BERT excels in context-heavy tasks, its performance difference between binary and multiclass tasks is not drastic.

The slight improvement in multiclass classification may stem from BERT's ability to capture contextual nuances, which is crucial for distinguishing between multiple classes in a complex dataset. The simplicity of binary classification, however, allows BERT to focus on more straightforward class boundaries, resulting in similarly high performance. This aligns with research highlighting BERT's strength in multilingual contexts, where its contextual embeddings provide an advantage in both binary and more intricate multiclass settings (Devlin, J. 2018).

7. Discussion

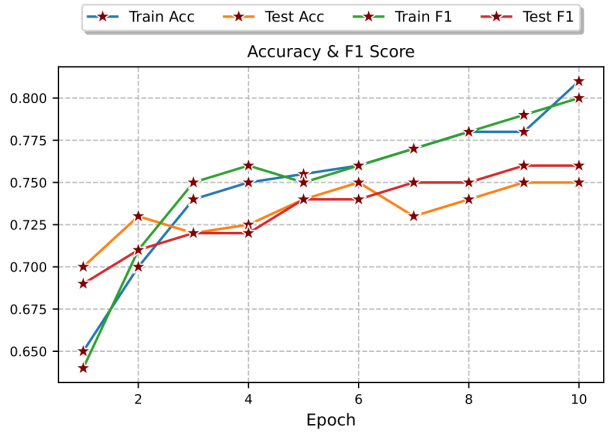
In this section, we compare the results across different models on various datasets and benchmark them against findings from previous research. This comprehensive analysis provides insights into the relative performance and effectiveness of each model in the context of hate speech detection.

For Basile binary classification task using LSTM with Word2Vec (300) embedding, the training and testing accuracy and F1 scores exhibit distinct trends. Initially, both metrics show a sharp increase, reflecting effective learning of underlying patterns. This is followed by a plateau phase, indicating convergence on key features. However, a slight overfitting is observed in the later epochs, where testing performance begins to decline while training metrics remain high. This behavior suggests the model is learning dataset-specific details rather than generalizable patterns. The issue may be influenced by class imbalance, which skews the model's focus towards majority classes. Techniques such as early stopping, class balancing through SMOTE (Chawla, N. V. 2002) or weighted loss functions can mitigate

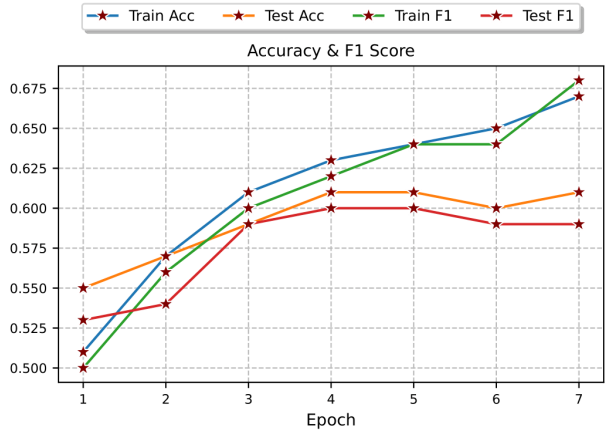
these effects. Furthermore, adopting contextual embeddings like ELMo or Flair, which offer richer semantic representations, has been shown to enhance generalization on imbalanced datasets (Akbik, A. 2019), (Peters, M. E. 2019).

The trends in multiclass LSTM models using Word2Vec 300 and GloVe 300 embeddings show similar learning behaviors. Initially, both models demonstrate improved accuracy and F1 scores, stabilizing after a few epochs as they reach their generalization capacity. The GloVe-based model converges faster and starts with higher metrics due to its robust embeddings (Pennington, J. 2014), while the Word2Vec-based model benefits from gradual learning but may require enhancements like attention mechanisms for comparable performance (Vaswani, A. 2017). Challenges such as overfitting and class

imbalance are evident, particularly when test metrics decline after stabilization or diverge from training metrics. Strategies like weighted loss functions, regularization, and hybrid models (e.g., combining CNN with LSTM layers) can address these issues and improve generalization.



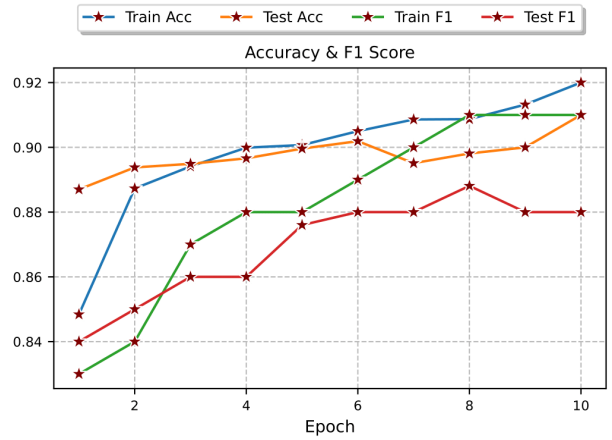
(a) Binary Classification with LSTM using Word2Vec (Basile).



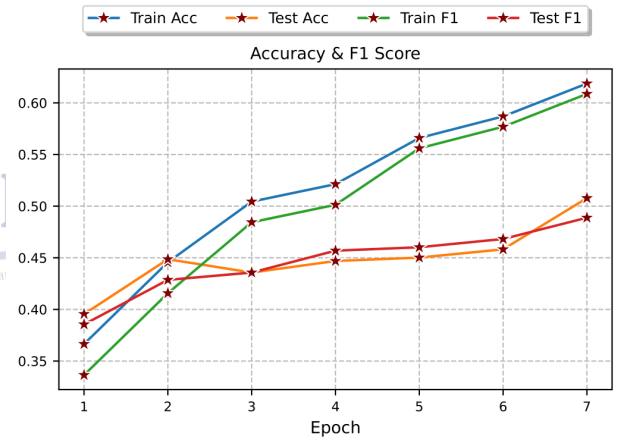
(c) Multi-class Classification with LSTM using Word2Vec (dynamic).

8. Software and Hardware Configuration

The project leveraged the computational resources provided by Google Colab, utilizing the NVIDIA Tesla T4 GPU, which is well-suited for deep learning tasks due to its high performance and efficiency. On the software front, We utilized Pandas and NumPy for data manipulation, gensim and nltk for text processing, and scikit-learn's CountVectorizer and TfidfVectorizer for feature extraction. For the LSTM model, we incorporated Keras, which runs on TensorFlow's backend. These libraries were chosen



for their widespread use in the research community, extensive documentation, and active support, making them a reliable choice.



(b) Multi-class accuracy with LSTM using glove (Davidson).

(d) Multi-class accuracy with LSTM using word2vec (mendal).

9. Conclusion

Our project has made significant progress in tackling the complex issue of identifying hate speech in text data. By carefully preprocessing the data and applying machine learning and deep learning techniques, we've built a system known for its accuracy and reliability.

We integrated Random Forest and XGBoost classifiers with TF-IDF vectorization and word embeddings, forming a strong foundation for our

models. Additionally, our deep learning component, powered by an LSTM network with GloVe embeddings, enhances our system's ability to understand language nuances. Furthermore, we utilized BERT for binary and multiclass classification tasks on multilingual datasets, significantly enhancing hate speech detection.Through thorough evaluation using various metrics, we've confirmed the effectiveness of our models and identified areas for improvement. Our architecture is adaptable, accommodating different feature representations and model types to improve hate speech detection. Looking ahead, we plan to further enhance the performance of all our models. This involves fine-tuning BERT more extensively across a wider range of datasets, as well as refining our machine learning and LSTM models. Additionally, we are considering strategies to address dataset bias and imbalance. One approach is to augment the dataset by increasing the representation of underrepresented classes, particularly negative samples, to decrease imbalance and bias. By improving the balance of our dataset, we aim to enhance the robustness and fairness of our models, ensuring even greater accuracy and reliability in hate speech detection.

REFERENCES

Baggs, M. (2021). Online hate speech rose 20% during pandemic: 'We've normalised it'. *BBC News*.

Alkomah, F., & Ma, X. (2022). A literature review of textual hate speech detection methods and datasets. *Information, 13*(6), 273.

Djuric, N., Zhou, J., Morris, R., Grbovic, M., Radosavljevic, V., & Bhamidipati, N. (2015). Hate speech detection with comment embeddings. In *Proceedings of the 24th International Conference on World Wide Web* (pp. 29–30).

Gröndahl, T., Pajola, L., Juuti, M., Conti, M., & Asokan, N. (2018). All you need is "love": Evading hate speech detection. In *Proceedings of the 11th ACM Workshop on Artificial Intelligence and Security* (pp. 2–12).

Nagar, S., Barbhuiya, F. A., & Dey, K. (2023). Towards more robust hate speech detection: Using social context and user data. *Social Network Analysis and Mining, 13*(1), 47.

Almaliki, M., Almars, A. M., Gad, I., & Atlam, E.-S. (2023). ABMM: Arabic BERT-mini model for hate-speech detection on social media. *Electronics, 12*(4), 1048.

Narayan, N., Biswal, M., Goyal, P., & Panigrahi, A. (2023). Hate speech and offensive content detection in Indo-Aryan languages: A battle of LSTM and transformers. *arXiv preprint*, arXiv:2312.05671.

Anusha, M. D., & Shashirekha, H. L. (2020). An ensemble model for hate speech and offensive content identification in Indo-European languages. In *FIRE (Working Notes)* (pp. 253–259).

Davidson, T., Warmsley, D., Macy, M., & Weber, I. (2017). Automated hate speech detection and the problem of offensive language. In *Proceedings of the International AAAI Conference on Web and Social Media, 11*(1), 512–515.

Shakil, M. H., & Alam, M. G. R. (2022). Hate speech classification implementing NLP and CNN with machine learning algorithm through interpretable explainable AI. In *2022 IEEE Region 10 Symposium (TENSYMP)* (pp. 1–6). IEEE.

Keipi, T., Näsi, M., Oksanen, A., & Räsänen, P. (2016). *Online hate and harmful content: Cross-national perspectives*. Taylor & Francis.

Castaño-Pulgarín, S. A., Suárez-Betancur, N., Vega, L. M. T., & López, H. M. H. (2021). Internet, social media and online hate speech: Systematic review. *Aggression and Violent Behavior, 58*, 101608.

Wachs, S., Gámez-Guadix, M., & Wright, M. F. (2022). Online hate speech victimization and depressive symptoms among adolescents: The protective role of resilience. *Cyberpsychology, Behavior, and Social Networking, 25*(7), 416–423.

Saha, K., Chandrasekharan, E., & De Choudhury, M. (2019). Prevalence and psychological effects of hateful speech in online college communities. In *Proceedings of the 10th ACM Conference on Web Science* (pp. 255–264).

IEER
Institute for Excellence in Education & Research

Wypych, M., & Bilewicz, M. (2024). Psychological toll of hate speech: The role of acculturation stress in the effects of exposure to ethnic slurs on mental health among Ukrainian immigrants in Poland. *Cultural Diversity & Ethnic Minority Psychology, 30*(1), 35.

Chetty, N., & Alathur, S. (2018). Hate speech review in the context of online social networks. *Aggression and Violent Behavior, 40*, 108–118.

Peng, S., Zhou, Y., Cao, L., Yu, S., Niu, J., & Jia, W. (2018). Influence analysis in social networks: A survey. *Journal of Network and Computer Applications, 106*, 17–32.

Roy, P. K., Bhawal, S., & Subalalitha, C. N. (2022). Hate speech and offensive language detection in Dravidian languages using deep ensemble framework. *Computer Speech & Language, 75*, 101386.

Mandl, T., Modha, S., Majumder, P., Patel, D., Dave, M., Mandlia, C., & Patel, A. (2019). Overview of the HASOC track at FIRE 2019: Hate speech and offensive content identification in Indo-European languages. In *Proceedings of the 11th Annual Meeting of the Forum for Information Retrieval Evaluation* (pp. 14–17).

Risch, J., Schmidt, P., & Krestel, R. (2021). Data integration for toxic comment classification: Making more than 40 datasets easily accessible in one unified format. In *Proceedings of the 5th Workshop on Online Abuse and Harms (WOAH 2021)* (pp. 157–163).

Basile, V., Bosco, C., Fersini, E., Nozza, D., Patti, V., Rangel Pardo, F. M., Rosso, P., & Sanguinetti, M. (2019). SemEval-2019 Task 5: Multilingual detection of hate speech against immigrants and women in Twitter. In *Proceedings of the 13th International Workshop on Semantic Evaluation* (pp. 54–63).

Vidgen, B., Thrush, T., Waseem, Z., & Kiela, D. (2020). Learning from the worst: Dynamically generated datasets to improve online hate detection. *arXiv preprint arXiv:2012.15761*.

Gupta, S., Kanchinadam, T., Conathan, D., & Fung, G. (2020). Task-optimized word embeddings for text classification representations. *Frontiers in Applied Mathematics and Statistics, 5*, 67.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*

He, H., & Garcia, E. A. (2009). Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering, 21*(9), 1263–1284.

Louppe, G. (2014). Understanding random forests: From theory to practice. *arXiv preprint arXiv:1407.7502*.

Tomita, T. M., Browne, J., Shen, C., Chung, J., Patsolic, J. L., Falk, B., ... & Maggioni, M. (2020). Sparse projection oblique randomer forests. *Journal of Machine Learning Research, 21*(104), 1–39.

Florek, P., & Zagdański, A. (2023). Benchmarking state-of-the-art gradient boosting algorithms for classification. *arXiv preprint arXiv:2305.17094*.

Lian, J., Freeman, L., Hong, Y., & Deng, X. (2021). Robustness with respect to class imbalance in artificial intelligence classification algorithms. *Journal of Quality Technology, 53*(5), 505–525.

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research, 16*, 321–357.

Akbik, A., Bergmann, T., Blythe, D., Rasul, K., Schweter, S., & Vollgraf, R. (2019). FLAIR: An easy-to-use framework for state-of-the-art NLP. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations* (pp. 54–59).

Peters, M. E., Neumann, M., Logan IV, R. L., Schwartz, R., Joshi, V., Singh, S., & Smith, N. A. (2019). Knowledge enhanced contextual word representations. *arXiv preprint arXiv:1909.04164*.

Pennington, J., Socher, R., & Manning, C. D. (2014). GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 1532–1543).

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems*.

Breiman, L. (2001). Random forests. *Machine Learning, 45*, 5–32.

Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 785–794).

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation, 9*(8), 1735–1780.

Werbos, P. J. (1990). Backpropagation through time: What it does and how to do it. *Proceedings of the IEEE, 78*(10), 1550–1560.

IEER
Institute for Excellence in Education & Research