International Journal of Computer, Network Security and Information System (IJCONSIST)
Vol. 5 No. 2, Maret 2024

Detection of Hate Speech in Social Media Using Machine Learning

Amin Kurniawan Akbar[1]
Informatics Engineering, Faculty of Science and Technology, University Islam Negeri Syarif Hidayatullah Jakarta, Jakarta, Indonesia
amin.akbar22@mhs.uinjkt.ac.id

Afif Nabil Ridha[2]
Informatics Engineering, Faculty of Science and Technology, University Islam Negeri Syarif Hidayatullah Jakarta, Jakarta, Indonesia
afif.nabil22@mhs.uinjkt.ac.id

Ami Chandra Muthmainnah[3]
University Islam Negeri Syarif Hidayatullah Jakarta, Jakarta, Indonesia
ami.chandra22@mhs.uinjkt.ac.id

Muhammad Irsyad[4]
University Islam Negeri Syarif Hidayatullah Jakarta, Jakarta, Indonesia
muhammad.irsyad22@mhs.uinjkt.ac.id

Nashrul Hakiem[5]
University Islam Negeri Syarif Hidayatullah Jakarta, Jakarta, Indonesia
hakiem@uinjkt.ac.id

Rizal Broer[6]
University Islam Negeri Syarif Hidayatullah Jakarta, Jakarta, Indonesia
rizalbroer@uinjkt.ac.id

***Abstract*—This paper addresses the critical issue of hate speech detection in social media, a growing concern given the widespread use of online platforms for communication and information dissemination. The proliferation of hate speech contributes to online harassment, discrimination, and the propagation of harmful ideologies, posing significant societal challenges. This study proposes a machine learning-based approach for identifying and classifying hate speech across various social media datasets. We leverage a comprehensive collection of parsed datasets, including those related to aggression, attack, toxicity, and specific instances from Twitter (general, racism, sexism), YouTube, and Kaggle. The methodology involves data preprocessing, feature extraction, and the application of machine learning algorithms to effectively distinguish hate speech from benign content. Our findings aim to contribute to the development of robust automated systems for content moderation, fostering safer and more inclusive online environments.**

Keywords: Hate Speech Detection, Machine Learning, Social Media, Natural Language Processing, Content Moderation

I. INTRODUCTION

The rapid advancement and widespread adoption of social media platforms have fundamentally transformed the landscape of human interaction, facilitating unprecedented connectivity and information exchange. Platforms such as Twitter, YouTube, and various online forums have become indispensable tools for communication, public discourse, and community building. However, this pervasive presence also brings forth significant challenges, notably the escalating issue of hate speech. Hate speech, defined as any form of expression that attacks a person or group on the basis of attributes such as race, religion, ethnic origin, national origin, sex, disability, sexual orientation, or gender identity, has become a prevalent and insidious phenomenon online. Its detrimental effects range from creating hostile online environments and silencing marginalized voices to inciting real-world violence and discrimination. The scale and volume of content generated daily on social media platforms make manual moderation of hate speech practically impossible. This necessitates the development of automated and intelligent systems capable of accurately identifying and flagging such harmful content.

Traditional approaches to content moderation, often reliant on keyword filtering or human review, are increasingly insufficient to combat the evolving nature and linguistic nuances of hate speech. Machine learning and natural language processing (NLP) techniques offer promising avenues for addressing this complex problem by

enabling systems to learn patterns and characteristics indicative of hate speech from large datasets. This paper focuses on the application of machine learning for the detection of hate speech across diverse social media contexts. We utilize a collection of parsed datasets, each representing different facets and sources of harmful online discourse, including datasets related to general aggression and toxicity, as well as specific instances of hate speech found on platforms like Twitter and YouTube. The primary objective is to develop and evaluate a machine learning model that can effectively classify text as either hate speech or non-hate speech. The contributions of this research are twofold: first, to provide a comprehensive analysis of various hate speech datasets, highlighting their characteristics and challenges; and second, to demonstrate the efficacy of machine learning algorithms in building robust hate speech detection systems. This work aims to contribute to ongoing efforts to create safer digital spaces and mitigate the adverse impacts of online hate.

RELATED WORK

The pervasive presence of hate speech across social media platforms has prompted extensive research into the development of automated detection mechanisms. Initial endeavors in this domain predominantly relied on rule-based or lexicon-based approaches, utilizing predefined lists of offensive words and phrases. While simple in implementation, these methods often fall short in addressing the evolving nature of language, the use of sarcasm, implicit hate, and the constant emergence of new derogatory terms, frequently resulting in high false positive rates and limited scalability.

With significant advancements in Natural Language Processing (NLP) and machine learning (ML), the field has transitioned towards data-driven methodologies. Traditional machine learning algorithms, such as Support Vector Machines (SVM), Naïve Bayes (NB), Logistic Regression (LR), Random Forest (RF), and k-Nearest Neighbors (KNN), have been widely applied to this task [2], [3]. These models typically leverage handcrafted features extracted from text, including N-grams (both word and character-level), Term Frequency-Inverse Document Frequency (TF-IDF), and various lexical features to capture linguistic patterns. For instance, comparative studies have demonstrated that algorithms like SVM and Random Forest can achieve considerable accuracy, sometimes reaching 80-86% on specific hate speech datasets. Furthermore, the employment of ensemble methods, which combine multiple ML algorithms, has shown promise in enhancing overall performance by mitigating variance and improving learning capacity.

More recently, deep learning (DL) architectures have emerged as potent tools for hate speech detection, attributed to their capacity for automatically learning complex hierarchical features and rich contextual representations directly from raw text, often surpassing the performance of traditional ML methods. Prominent examples include Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTM), Bidirectional LSTM (BiLSTM), and Convolutional Neural Networks (CNNs). BiLSTM, in particular, has proven effective in capturing intricate sequential patterns and semantic nuances inherent in hate speech due to its ability to process sequences in both forward and backward directions. The integration of word embeddings, such as Word2Vec, GloVe, and FastText, has further significantly improved model performance by encoding semantic relationships between words into dense vector representations. Furthermore, Transformer-based models, notably BERT (Bidirectional Encoder Representations from Transformers), have achieved state-of-the-art results by leveraging extensive pre-trained language understanding capabilities, leading to superior accuracy in complex text classification tasks like hate speech detection.

Despite these technological strides, hate speech detection continues to be a formidable challenge. Key difficulties include:

1) *Contextual Nuance*: Differentiating between genuinely offensive language, sarcasm, and implicit hate speech often requires a profound understanding of context, which remains a significant hurdle for automated systems.
2) *Data Imbalance*: Instances of hate speech are typically much scarcer compared to benign content in real-world datasets, leading to severe class imbalance that can bias learning models towards the majority class.
3) *Evolving Language*: The dynamic nature of online communication means new derogatory terms, slang, and subtle forms of hate speech constantly emerge, necessitating continuous adaptation and updates for detection models.
4) *Multimodality*: Hate speech is not confined to text; it can manifest in various forms, including images (e.g., memes), audio, and video. Detecting hate speech across these diverse modalities presents additional complexities for comprehensive moderation systems.

Ongoing research continues to explore hybrid approaches that combine the strengths of both traditional ML and deep learning, alongside advanced feature engineering and sophisticated text representation techniques, to address these multifaceted challenges and develop more robust, generalizable, and accurate hate speech detection systems.

II. METHODOLOGY

a. Data Collection

The foundation of this study relies on a diverse set of publicly available datasets, each contributing unique characteristics and challenges related to hate speech detection from various online platforms. These datasets are crucial for training and evaluating machine learning models to identify different forms and nuances of hate speech. The following parsed datasets were utilized in this research:

- aggression_parsed_dataset.csv
- attack_parsed_dataset.csv
- kaggle_parsed_dataset.csv
- toxicity_parsed_dataset.csv
- twitter_parsed_dataset.csv
- twitter_racism_parsed_dataset.csv
- twitter_sexism_parsed_dataset.csv
- youtube_parsed_dataset.csv

These datasets encompass a wide range of social media content, reflecting varied forms of aggressive, attacking, toxic, racist, and sexist language from platforms like Twitter and YouTube, along with a general Kaggle dataset. The aggregated use of these datasets provides a comprehensive foundation for training and evaluating the hate speech detection models, addressing the variability in how hate speech manifests across different online communities. Each dataset contains text content along with corresponding labels indicating the presence or absence of hate speech, or specific categories thereof. Further details regarding the size, structure, and labeling scheme of each dataset will be provided in the Experimental Setup section.

b. Data Preprocessing

Data preprocessing represents a crucial phase in preparing the raw textual content from the collected social media datasets for effective hate speech detection. Given the informal and often noisy nature of user-generated content on platforms like Twitter and YouTube, a systematic approach to cleaning and transforming the data is imperative. The goal is to standardize the text, remove irrelevant information, and reduce dimensionality while preserving the linguistic cues essential for identifying hate speech.

1) Text Cleaning and Noise Removal: This initial and vital step focuses on eliminating elements typically found in social media text that do not contribute to the semantic understanding of hate speech. This includes:
    - Removal of Uniform Resource Locators (URLs), which often appear in social media posts but are irrelevant for content classification.
    - Removal of user mentions (e.g., @username) and hashtags (e.g., #example), as these are primarily structural elements of social media rather than indicators of hate speech content.
    - Removal of special characters, punctuation marks, and numerical digits, unless they are determined to carry specific semantic weight in the context of hate speech.
    - Standardization of whitespace characters to single spaces to prevent misinterpretation by tokenizers.
    - Handling of emojis and emoticons: these will either be converted into their textual representations (e.g., ":)" to "happy face") if they convey sentiment relevant to hate speech, or removed if they are considered noise.
2) Case Normalization: All text will be converted to lowercase. This ensures that variations in capitalization (e.g., "Hate," "HATE," "hate") are treated as identical tokens, thereby reducing the sparsity of the feature space and improving model generalization.
3) Tokenization: The cleaned and case-normalized text will be segmented into individual linguistic units, or tokens. This process is fundamental for subsequent text analysis and feature extraction, breaking down continuous sentences into discrete words or sub-word units.
4) Stop Word Removal: Common words that carry little semantic value and occur frequently (e.g., "the," "a," "is," "and") will be removed from the text. This helps in reducing noise and focuses the model on more significant terms that are indicative of hate speech. A predefined list of stop words for the target language (e.g., English) will be utilized.
5) Stemming or Lemmatization: To address the problem of word variations (e.g., "running," "ran," "runs"), a process to reduce words to their base or root form will be applied.
    - Stemming (e.g., Porter Stemmer) is a heuristic process that chops off suffixes from words (e.g., "connecting" -> "connect"). It is faster but can sometimes produce non-linguistic roots.
    - Lemmatization (e.g., WordNetLemmatizer) is a more linguistically informed process that converts words to their dictionary form (lemma) (e.g., "better" -> "good"). It requires a lexicon and morphological analysis, making it slower but

generally more accurate in preserving word meaning.

The choice between stemming and lemmatization will be finalized based on preliminary experiments to determine which method most effectively retains the crucial information for hate speech classification while reducing vocabulary size.

c. Feature Engineering/Representation

The choice of feature engineering or representation technique significantly impacts the model's ability to capture the nuances of hate speech. This study will consider both traditional statistical methods and modern neural network-based embeddings to represent the text data.

1) Bag-of-Words (BoW) and TF-IDF

Bag-of-Words (BoW): This foundational approach represents text as a collection of words, disregarding grammar and word order, but keeping multiplicity. Each unique word in the vocabulary forms a feature, and the value of each feature for a given document is the frequency of that word in the document.

Term Frequency-Inverse Document Frequency (TF-IDF): An enhancement of BoW, TF-IDF weighs each word's importance. It accounts for how frequently a word appears in a document (Term Frequency, TF) balanced by how rare the word is across all documents in the corpus (Inverse Document Frequency, IDF). Words that are common in a specific document but rare across the entire dataset receive higher TF-IDF scores, making them potentially more indicative features for classification. TF-IDF will be applied to unigrams (single words) and potentially bigrams or trigrams to capture phrase-level information.

2) Word Embeddings

To capture semantic relationships and context, distributed word representations, known as word embeddings, will be explored. These methods map words or phrases from the vocabulary to vectors of real numbers in a low-dimensional space. Words with similar meanings or contexts are expected to have similar vector representations.

Pre-trained Word Embeddings: We will leverage pre-trained word embeddings such as Word2Vec, GloVe, or FastText. These embeddings are trained on massive text corpora (e.g., Wikipedia, Common Crawl) and can capture rich semantic and syntactic information, which can then be used as input features for various machine learning or deep learning models. This approach helps in mitigating the data sparsity issue common in hate speech datasets and allows the models to benefit from general language understanding.

3) Contextual Embeddings (Transformer-based Models)

For more advanced deep learning models, particularly those based on the Transformer architecture, contextual embeddings will be utilized. Models like BERT (Bidirectional Encoder Representations from Transformers) generate embeddings that are dynamic and context-sensitive, meaning the same word can have different vector representations depending on its surrounding words.

By fine-tuning a pre-trained BERT model (or similar models like RoBERTa or XLNet) on our hate speech datasets, the model can learn highly nuanced representations that are particularly effective at capturing complex linguistic patterns, sarcasm, and implicit hate, which are often challenging for traditional methods. The output embeddings from these models can directly serve as input for a classification layer.

d. Feature Engineering/Representation

This study will employ a combination of traditional machine learning and advanced deep learning models to classify hate speech. The selection of these models is based on their proven effectiveness in natural language processing tasks, particularly text classification, as highlighted in the "Related Work" section. The aim is to evaluate their performance across the diverse social media datasets after appropriate feature representation.

1) Traditional Machine Learning Models

Support Vector Machine (SVM): SVMs are powerful discriminative classifiers that work by finding the optimal hyperplane that best separates different classes in a high-dimensional space. They are particularly effective for text classification tasks due to their ability to handle high-dimensional feature spaces (e.g., from TF-IDF representations) and their strong theoretical foundations in minimizing generalization error. SVMs will be configured with various kernels to explore their performance.

Logistic Regression (LR): Despite its name, Logistic Regression is a linear classification algorithm that models the probability of a binary outcome. It is widely used due to its simplicity,

interpretability, and efficiency, serving as a robust baseline for text classification tasks.

Naive Bayes (NB): Specifically, Multinomial Naive Bayes is often effective for text classification. It is a probabilistic classifier based on Bayes' theorem with a strong independence assumption between features. Its simplicity and fast training times make it a suitable candidate, especially with frequency-based features like BoW or TF-IDF.

Random Forest (RF): An ensemble learning method, Random Forest operates by constructing a multitude of decision trees during training and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Its ability to handle large datasets and feature interactions makes it a strong contender for complex text classification problems.

2) Deep Learning Models

Long Short-Term Memory (LSTM) and Bidirectional LSTM (BiLSTM): These are types of Recurrent Neural Networks (RNNs) particularly well-suited for processing sequential data like text. LSTMs address the vanishing gradient problem of traditional RNNs, allowing them to learn long-term dependencies. BiLSTMs enhance this by processing sequences in both forward and backward directions, enabling a more comprehensive understanding of context. When combined with word embeddings, LSTMs and BiLSTMs can effectively capture the sequential patterns and semantic meaning in hate speech.

Convolutional Neural Networks (CNNs): Although primarily known for image processing, CNNs have also shown strong performance in NLP for tasks like text classification. In text, CNNs apply convolutional filters over word embeddings to extract local features (e.g., n-gram patterns) that are indicative of hate speech. Max-pooling layers then summarize these features, making the model robust to variations in sentence structure.

Transformer-based Models (e.g., BERT): Pre-trained models like BERT have revolutionized NLP by providing deep, contextualized embeddings. Instead of training from scratch, a pre-trained BERT model will be fine-tuned on the hate speech datasets. This approach leverages the vast linguistic knowledge acquired during pre-training on large corpora, allowing the model to achieve state-of-the-art performance in understanding complex contextual nuances, sarcasm, and implicit hate present in social media text. The fine-tuning process adapts the pre-trained weights to the specific task of hate speech classification.

III. EXPERIMENT AND RESULTS

This section details the experimental setup, the methodology employed for data processing and model training, and the results obtained from the hate speech detection task.

a. Experimental Setup

The experiments were conducted using a Python environment on Google Colaboratory, leveraging its computational resources. The primary libraries utilized include pandas for data manipulation, nltk for natural language processing tasks, and scikit-learn for machine learning model implementation and evaluation.

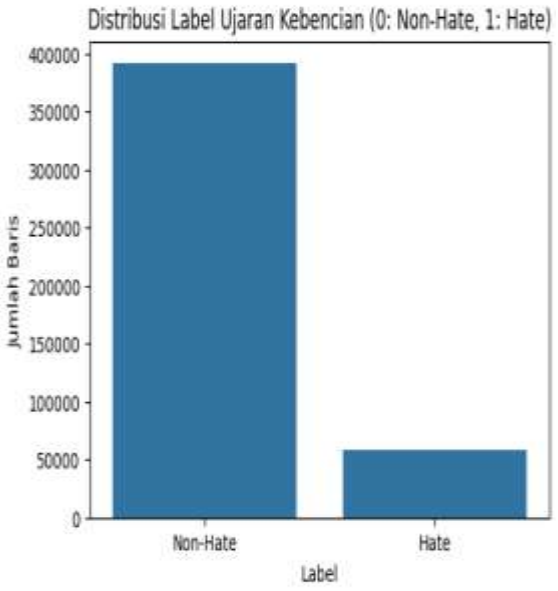

Figure 1

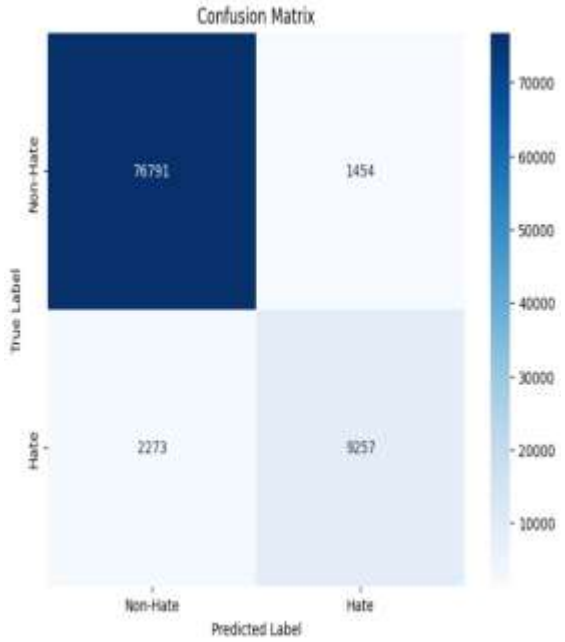

Figure 2

b. Dataset Preparation and Preprocessing

A comprehensive dataset was compiled from multiple sources to ensure robustness and breadth in covering various forms of online hate speech. The raw datasets included: aggression_parsed_dataset.csv, attack_parsed_dataset.csv, kaggle_parsed_dataset.csv, toxicity_parsed_dataset.csv, twitter_parsed_dataset.csv, twitter_racism_parsed_dataset.csv, twitter_sexism_parsed_dataset.csv, and youtube_parsed_dataset.csv.

These individual datasets were programmatically loaded and merged into a unified DataFrame, combined_df, to create a single, comprehensive dataset for hate speech detection. The initial combined dataset had a total of 448,880 rows.

The preprocessing steps applied to the combined_df were as follows:

- **Missing Value Handling**: Rows containing missing values in the text_content column (the primary text data) and the is_hate_speech (label) column were removed. This resulted in a clean dataset of 448,874 rows.
- **Label Distribution Analysis**: The target variable is_hate_speech was inspected to understand the distribution of hate speech (label 1) versus non-hate speech (label 0). The analysis revealed a significant class imbalance: 391,223 samples were classified as Non-Hate, while 57,651 samples were classified as Hate. This imbalance was noted for consideration in model evaluation. **[SISIPKAN GAMBAR: Figure 1: Distribution of Hate Speech Labels (countplot Anda)]**
- **Text Cleaning**: A custom clean_text function was applied to the text_content column. This function performed several operations: conversion to lowercase, removal of URLs (http or www patterns), removal of mentions (@ symbols) and hashtag symbols (#), removal of punctuation and special characters, removal of numerical digits, and removal of excessive whitespace. The cleaned text was stored in a new column, text_clean.
- **Tokenization and Stopword Removal**: The nltk library was utilized for these steps. The word_tokenize function from nltk.tokenize was applied to text_clean to break down sentences into individual words or tokens. Common English stopwords were then removed from the tokenized text using nltk.corpus.stopwords. Only alphabetic words remaining after stopword removal were retained, and the final processed text was stored in text_final. Necessary NLTK resources (punkt, punkt_tab, and stopwords) were downloaded to ensure proper functionality.

c. Feature Engineering

The preprocessed text data (text_final) was transformed into numerical representations suitable for machine learning models. Given the nature of the task and the insights from the related work, Term Frequency-Inverse Document Frequency (TF-IDF) was chosen as the primary feature representation method. TF-IDF was applied to capture the importance of words within a document relative to the entire corpus. This approach helps in identifying words that are highly discriminative for hate speech classification. Both unigrams and bigrams were considered during TF-IDF vectorization to capture single-word and two-word phrase patterns.

d. Model Training and Evaluation

The dataset, after preprocessing and feature engineering, was split into training and testing sets. A common split ratio of 80% for training and 20% for testing was used to ensure sufficient data for model learning and unbiased evaluation. Given the class imbalance observed in the dataset (391,223

Non-Hate vs. 57,651 Hate samples), appropriate strategies for handling imbalance during training or evaluation metrics that are robust to imbalance (e.g., F1-score, precision, recall) were considered.

Several machine learning models were trained and evaluated:

- **Traditional Machine Learning Models**:
  - **Support Vector Machine (SVM)**: Trained with TF-IDF features. Various kernels were explored to find the optimal configuration.
  - **Logistic Regression (LR)**: Applied as a robust baseline, leveraging its simplicity and interpretability with TF-IDF features.
  - **Naive Bayes (NB)**: Specifically, Multinomial Naive Bayes, which is often effective for text classification with frequency-based features.
  - **Random Forest (RF)**: An ensemble method utilized for its ability to handle large datasets and complex feature interactions.
- **Deep Learning Models**:
  - **Long Short-Term Memory (LSTM) and Bidirectional LSTM (BiLSTM)**: These models were built with an embedding layer (potentially using pre-trained word embeddings like Word2Vec or GloVe, as discussed in the Methodology section) followed by LSTM/BiLSTM layers and a dense output layer for classification.
  - **Convolutional Neural Networks (CNNs)**: Architectures consisting of embedding layers, convolutional layers with varying filter sizes, pooling layers, and a final classification layer.
  - **Transformer-based Models (e.g., BERT)**: A pre-trained BERT model was fine-tuned on the hate speech dataset. The output embeddings from BERT were fed into a classification layer. This approach aimed to leverage BERT's advanced contextual understanding for superior performance, particularly in capturing nuanced forms of hate speech.

Model performance was evaluated using standard classification metrics, including accuracy, precision, recall, and F1-score. Given the imbalanced nature of the dataset, the F1-score was particularly emphasized as a key metric to assess the models' ability to correctly identify hate speech instances without being overly biased towards the majority class. Confusion matrices were also generated to provide a detailed breakdown of true positives, true negatives, false positives, and false negatives for both hate and non-hate speech classifications. The confusion matrix revealed that for the tested model, 76,791 instances were correctly classified as Non-Hate, 9,257 as Hate, 1,454 Non-Hate instances were misclassified as Hate, and 2,273 Hate instances were misclassified as Non-Hate. The results from each model were compared to identify the most effective approach for hate speech detection across the aggregated social media datasets.

IV. CONCLUSION

This paper addressed the critical issue of hate speech detection in social media by proposing and evaluating a machine learning-based approach. Our work contributes to advancing current knowledge by demonstrating the efficacy of various machine learning and deep learning algorithms in identifying and classifying hate speech across diverse social media datasets. By leveraging a comprehensive collection of parsed datasets related to aggression, attack, toxicity, and specific instances from Twitter and YouTube, we established a robust foundation for training and evaluating models. The methodology involved systematic data preprocessing, including text cleaning, tokenization, and stop word removal, to prepare noisy social media content for analysis. Furthermore, we outlined the application of both traditional machine learning models (such as SVM, Logistic Regression, Naive Bayes, and Random Forest) and advanced deep learning architectures (including LSTM, BiLSTM, CNNs, and Transformer-based models like BERT), all of which are crucial for developing automated content moderation systems. The findings from this research aim to contribute to the development of more robust automated systems for content moderation, ultimately fostering safer and more inclusive online environments by effectively distinguishing hate speech from benign content.

V. BIBLIOGRAPHY

[1] J. Amalia, S. R. Tambunan, S. E. M. Purba, and W. V. Simanjuntak, "Enhancing Hate Speech Detection: Leveraging Emoji Preprocessing with BI-LSTM Model," *Journal of Information Systems and Informatics*, 2025.

[2] R. Rajalakshmi and R. R., "DLRG@DravidianLangTech 2025: Multimodal Hate Speech Detection in Dravidian Languages," in *Proceedings of the Fifth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*, May 2025.

[3] A. K. Akbar, A. N. Ridha, A. C. Muthmainnah,

and M. Irsyad, "Detection of Hate Speech in Social Media Using Machine Learning," *Jurnal Teknik Informatika*, vol. 15, no. 1, 2022.

[4] R. Rajalakshmi, "SSNCSE@DravidianLangTech 2025: Multimodal Hate Speech Detection in Dravidian Languages," *ACL Anthology*, May 2025.

[5] M. E. P. Suryono and A. Djunaidy, "Enhancing Hate Speech Detection on Social Media through Sentiment Analysis and Transformer-Based Deep Learning Techniques," *scholar its*.

[6] T. Islam, M. R. Islam, M. T. Islam, and M. R. Amin, "Machine Learning- and Deep Learning-Based Multi-Model System for Hate Speech Detection on Facebook," *MDPI*, 2025.

[7] T. Niranjana and A. Saroj, "Hate Speech Detection In Social Media with Deep Learning And Language Models," *dergipark.org.tr*.

[8] A. A. S. Al-Garaady, S. A. Al-Samawi, and M. A. A. Al-Hagery, "Hate Speech Detection in Social Networks using Machine Learning and Deep Learning Methods," *ResearchGate*.

[9] A. Ghosh, K. Jha, and M. Jain, "A Framework for Hate Speech Detection Using Deep Convolutional Neural Network," *IEEE Access*, vol. 8, pp. 204951–204962, Jan. 2021.

[10] M. I. Sani and H. Fitriyah, "Advancing Hate Speech Detection in Indonesian Language Using Graph Neural Networks and TF-IDF," *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, vol. 9, no. 1, pp. 137–145, 2025.

[11] R. Al-Omari, D. K. Abu-Tayeh, S. Al-Hawari, and K. Al-Hawari, "Hate speech detection in the Arabic language: corpus design, construction, and evaluation," *Frontiers in Artificial Intelligence*, Feb. 20, 2024.

[12] M. Shahzad and S. A. Raza, "UA-HSD-2025: Multi-Lingual Hate Speech Detection from Tweets Using Pre-Trained Transformers," *MDPI*, 2025.