Complex & Intelligent Systems
https://doi.org/10.1007/s40747-024-01392-5

ORIGINAL ARTICLE

Multi-class hate speech detection in the Norwegian language using FAST-RNN and multilingual fine-tuned transformers

Ehtesham Hashmi[1] · Sule Yildirim Yayilgan[1]

Received: 18 December 2023 / Accepted: 12 February 2024

Abstract

The growth of social networks has provided a platform for individuals with prejudiced views, allowing them to spread hate speech and target others based on their gender, ethnicity, religion, or sexual orientation. While positive interactions within diverse communities can considerably enhance confidence, it is critical to recognize that negative comments can hurt people's reputations and well-being. This emergence emphasizes the need for more diligent monitoring and robust policies on these platforms to protect individuals from such discriminatory and harmful behavior. Hate speech is often characterized as an intentional act of aggression directed at a specific group, typically meant to harm or marginalize them based on certain aspects of their identity. Most of the research related to hate speech has been conducted in resource-aware languages like English, Spanish, and French. However, low-resource European languages, such as Irish, Norwegian, Portuguese, Polish, Slovak, and many South Asian, present challenges due to limited linguistic resources, making information extraction labor-intensive. In this study, we present deep neural networks with FastText word embeddings using regularization methods for multi-class hate speech detection in the Norwegian language, along with the implementation of multilingual transformer-based models with hyperparameter tuning and generative configuration. FastText outperformed other deep learning models when stacked with Bidirectional LSTM and GRU, resulting in the FAST-RNN model. In the concluding phase, we compare our results with the state-of-the-art and perform interpretability modeling using Local Interpretable Model-Agnostic Explanations to achieve a more comprehensive understanding of the model's decision-making mechanisms.

Keywords Hate speech · Norwegian language · Natural language processing · Deep Learning · Transformers · Interpretability modeling

Introduction

The complexity and challenges of hate speech in the digital era

As digital technology advances, the era of social computing has significantly enhanced the way individuals interact, especially noticeable in the use of social media platforms and chat forums [30]. The concept of Hate Speech (HS), often veiled in complexity, holds diverse interpretations across regions and cultures, presenting significant hurdles in its detection and control, particularly in our digital age. HS appears in several forms [12], including cyberbullying [71], flaming [47], profanity [25], abusive language [50], toxicity [58], and discrimination [72]. While there is no universally accepted definition of HS, Nobata et al. [48] presented the most widely accepted: "any form of communication that denigrates a specific group of individuals based on attributes such as race, color, ethnicity, gender, sexual orientation, nationality, religion, or other distinguishing characteristics". Several studies align on a similar depiction of HS; [16, 34, 63, 65], characterizing it as an intentional act of hostility towards a specific group, influenced by real or perceived characteristics that constitute the group's identity. The huge increase in disparaging remarks on Twitter and other cyber platforms is leading to physical violence in the real world. As a result, the research community considers the automated detection of hate-related content on Twitter as a significant challenge [76]. Online HS

✉ Ehtesham Hashmi
hashmi.ehtesham@ntnu.no

Sule Yildirim Yayilgan
sule.yildirim@ntnu.no

[1] Department of Information Security and Communication Technology (IIK), Norwegian University of Science and Technology (NTNU), Teknologivegen 22, 2815 Gjøvik, Innlandet, Norway

Published online: 21 March 2024

is at the junction of various societal disputes [26]. It demonstrates the revolutionary impact of technology by bringing both opportunities and difficulties. It is difficult to attain a balance between fundamental rights [27], such as freedom of expression [11], and the defense of human dignity.

Regulatory measures and the global response to online hate speech

Maintaining a safe and pleasant online environment may be extremely difficult because of how amplified such behavior can become when there is anonymity and disconnection from consequences in the real world [51]. Effective and accurate methods to identify and resolve these problems require immediate and keen attention because of their rapid growth and the nature of evolution. As the custodian of freedom of expression, UNESCO actively promotes mutual understanding through all forms of mass communication, including the Internet and social media [23, 40]. On the 31st of May 2016, a voluntary code to stop illegal HS online was introduced as a result of cooperation between the European Commission and Information Communications Technology (ICT) companies. This program mandates the removal of all content that aligns with the definition of HS as set forth by the European Union (E-U) [4]. With the outbreak of the COVID-19 pandemic, there has been a worldwide increase in HS and discrimination, prompting governments at all levels, from local to national, to emphasize the significance of community resilience. Furthermore, the impact of hatred and misinformation during the pandemic has been seen all around the world [19, 33]. The EU has established measures to control how external firms interact and combat the spread of hatred and its code of conduct has shown significant improvement in recent years. [1] These guidelines explicitly state that it is unlawful to participate in any activity that encourages or incites violence against a group or an individual, identifiable by characteristics, such as race, skin color, religion, ancestry, or cultural association [28]. The following figures; 1, 2 depict hate crime incidents in the US. From 2007 to 2020, an increased tendency was seen; however, stability was recorded from 2020 to 2021, indicating potential advances in handling hate.

This paper seeks to detect the HS for the Norwegian dataset by incorporating Deep Learning (DL) and multilingual transformer-based models with hyperparameter tuning. Next, the contributions of this research work are summarized, followed by how the rest of the paper is organized.

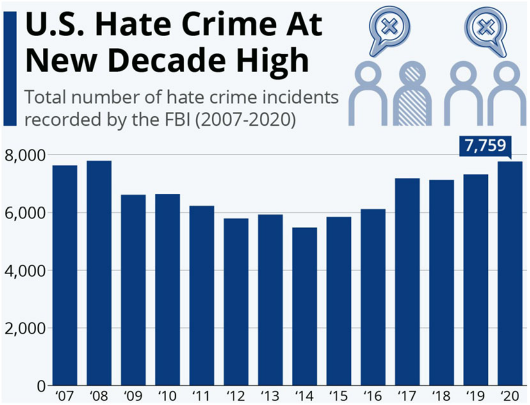


Fig. 1 Hate crime (2007–2020)

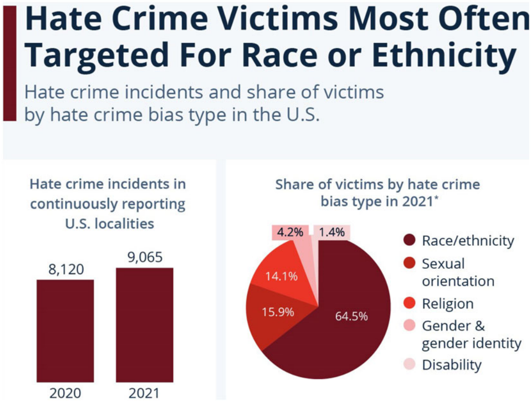


Fig. 2 Hate crime (2020–2021)

Work contributions

The contributions of this paper are as follows.

1. In this paper, our primary contribution is the refinement and application of established HS detection methodologies through the use of regularization methods, hyperparameter tuning, and generative configurations. This approach has been meticulously applied to a baseline dataset in the Norwegian dialect, which includes class categories like neutral, provocative, offensive, moderately hateful, and hateful. The aim is to significantly enhance HS detection capabilities specifically for the Norwegian language and within these distinct categories.
2. In addressing our classification problem, we strategically employed supervised FastText embeddings, offering distinct advantages over unsupervised FastText and other word embeddings. The supervised FastText embeddings are fine-tuned to the nuances of Norwegian HS data, capturing domain-specific context and enhancing the performance of sequential DL-based models, which include Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), and Convolutional Neural Network (CNN).

[1] https://european-union.europa.eu/principles-countries-history/principles-and-values/aims-and-values_en

3. We performed state-of-the-art multilingual transformer-based models, such as Multilingual Bidirectional Encoder Representations from Transformers (mBERT), ELECTRA, and FLAN-T5, along with Norwegian Language Models (LMs) like Nor-T5, Nor-BERT, Scandi-BERT, and nb-BERT. Notably, these Norwegian LMs, previously unexplored in the context of HS detection, were utilized with the implementation of hyperparameter tuning to optimize their performance for our task.
4. This work involves the implementation of prompt-based fine-tuning using two different techniques, including few-shot and full fine-tuning with generative configuration. This approach allows us to harness the power of transformer-based models and adapt them to our specific task.
5. Based on the best performance of Bidirectional LSTM and GRU (BiLSTM-GRU), we compared our results with the baseline study and performed the interpretability modeling with Local Interpretable Model-Agnostic Explanations (LIME) to achieve a more comprehensive understanding of the model's decision-making mechanisms.

Structure of the paper

The rest of the paper is structured as follows: Sect. 2 discusses the existing research work on HS. Section 3 explains the proposed work methodology. Section 4 focuses on the results and discussions. Section 5 is based on the comparison of the results with the baseline methods. Section 6 is related to the interpretability modeling with LIME. Section 7 presents the conclusion and future work.

Related work

Recent advancements in Artificial Intelligence (AI) and Natural Language Processing (NLP) have heightened the prominence of HS detection, leading to the development of various innovative methods in this field [43]. These techniques enhance the understanding of HS and its implications, including monitoring social media and analyzing public discourse. Moreover, the primary focus of these studies has been on well-resourced languages such as English. This emphasis on languages with abundant resources has created a disparity in hate speech research, especially for languages with fewer resources, such as Irish, Portuguese, Norwegian, and various South Asian languages [5].

Machine learning-based methods

H. Elzayady et al. [18] introduced a method for detecting HS in Arabic dialects using a combination of classical machine learning (ML) and DL-based approaches in two phases, incorporating personality traits. In the first phase, the AraPersonality dataset was used, applying correlation validation between personality traits and HS. In the second phase, the Term Frequency-Inverse Document Frequency (TF-IDF) was used for feature extraction. These features were then input into ML models, including Decision Tree (DT) [15], Random Forest (RF), Logistic Regression (LR), Support Vector Machine, and Extreme Gradient Boosting (XGBoost), alongside DL models, such as CNN, LSTM, BiLSTM, and GRU. In their research, Mittal et al. [45] focused on HS detection in the English language using an ML-based approach. They employed the XGBoost model with a Count Vectorizer (CV) for feature extraction. Additionally, they integrated the LIME model to interpret the predictions made by the machine learning algorithm. Their methodology achieved an F1-score of 0.94, demonstrating its effectiveness.

William et al. [75] addressed a tertiary classification problem in HS detection using ML-based methods. They employed both TF-IDF Word2Vec for feature extraction, finding that TF-IDF yielded better results compared to Word2Vec embeddings. The study implemented various models, including SVM, RF, AdaBoost, and KNN. Among these, SVM was found to outperform the other models in terms of accuracy. In their study, Akuma et al. [1] analyzed a dataset of HS and offensive language from kaggle[2] using four ML-based algorithms: KNN, DT, LR, and NB, along with two distinct word embeddings, BoW and TF-IDF. Their work showed that DT when integrated with TF-IDF achieved the best accuracy score of 0.92 when compared to the other models in their research. A. Khanday et al. [32] conducted HS detection on Twitter using COVID-19-related tweets, applying ML and ensemble learning techniques with TF-IDF and BoW. They collected 30,000 tweets during the pandemic, of which 11,000 were annotated as containing hate-related content. The Stochastic Gradient Boosting (SGB) classifier emerged as the most effective, achieving an accuracy and F1-score of 0.98.

Deep learning and transformer-based methods

Saleh et al. [61] conducted binary classification for HS detection using BiLSTM and the transformer-based model BERT. Their research included three publicly available datasets: [16, 73, 74]. They employed three different types of embeddings: domain-specific, Word2Vec, and Global Vectors for Word Representation (GloVe) Word embeddings. The BERT model achieved a 96% F1-score on a combined balanced dataset, outperforming other DL-based methods. S. Nagar et al. [46] introduced a novel approach for HS, utilizing two pub-

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

licly available datasets [21] and [22]. Their proposed model, named the Variational Graph Auto-Encoder (VGAC), leverages multi-modal data by combining two distinct features: the textual content of tweets and the social network structure of the users who posted them. Initially, the text from a tweet is encoded using a chosen text encoder, and then, it undergoes further processing with a Fully Forward Neural Network (FFNN). Concurrently, the user's features, which include the social network structure, language usage, and metadata, are encoded using a social network encoder. By integrating the encoded text and user features, their framework aims to comprehensively understand the context of a tweet. Khan et al. [31] presented a deep neural network architecture for sentiment categorization in code-mixed texts. CNN layers are utilized for feature selection, and LSTM layers are applied to capture long-term dependencies in textual input. They also used several word embedding techniques, such as Word2Vec Continuous Bag of Words (CBoW), GLOVE, and FastText. A similar approach was used by Nagra et al. [46] where they conducted SA at the sentence level for RU using Faster Recurrent CNN (FR-CNN) on the RUSA-19 dataset.

Awal et al. [5] proposed a multilingual Model-Agnostic Meta-Learning (MAML) [52] method for detecting HS, employing different publicly available datasets. The base models used in their study were mBERT and XLM-R, alongside datasets from Founta et al. [21], i Orts, [49], Mandl et al. [39], and Bosco et al. [10]. In this study, their proposed model, HATE-MAML, outperformed the baseline models by over 3% in accuracy. In their study, Mazari et al. [41] performed multi-label HS detection using ensemble learning methods. They employed two different word embeddings, FastText and GloVe, and also trained a BERT model combined with BiLSTM and BiGRU, utilizing a dataset from the Kaggle.[3] The multi-labels in their study included categories, such as 'identity hate', 'threat', 'insult', 'obscene', 'toxic', and 'severe toxic'. Ali et al. [2] performed a tertiary classification of HS detection on Twitter for the Urdu language. This classification was divided into three categories: hate speech, offensive, and neutral. They utilized deep learning-based models, such as LSTM and GRU, stacked with FastText embeddings, and also implemented a transformer-based BERT model using the Hugging Face tokenizer. Among these models, BERT emerged as the most accurate, achieving a notable accuracy score of 0.73. A similar approach was undertaken by Mehta et al. [42], where they applied traditional ML algorithms, SVM, MNB, RF, LR, and DL-based model LSTM and the transformer-based BERT model. Among these, LSTM emerged as the most effective, achieving an impressive accuracy score of 0.98. After reviewing the existing literature, we conclude that many studies addressed HS using traditional ML and DL-based methods for online data. We will do this analysis using DL, and multilingual transformers with hyperparameter tuning methods, instruction fine-tuning, and with generative configurations in a different way that will provide us with a deep understanding of these approaches.

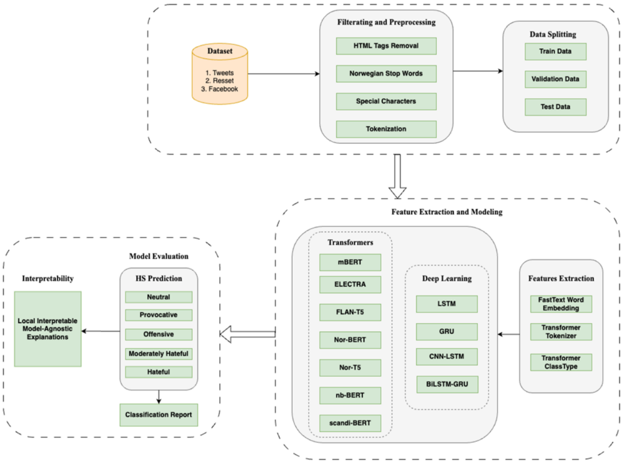


Fig. 3 Proposed work methodology

Table 1 represents the comparative analysis of the current SOTA studies. The prevailing existing methods in HS detection have shown a tendency to underutilize multilingual transformers and language-specific transformers, particularly those that leverage the increasingly popular prompt-based fine-tuning technique in generative AI. Additionally, these methods have primarily focused on word embedding techniques, often giving less attention to the crucial aspects of regularization and hyperparameter tuning that are essential for ensuring the robust performance of algorithms. In contrast, our work not only integrates these advanced transformer models and emphasizes the importance of regularization but also pioneers in applying prompt-based fine-tuning with generative configuration and explainable AI for multi-class HS detection in low-resource Norwegian language.

Table 2 highlights the prior research conducted on low-resource European languages.

Methodology

The proposed research methodology involves a systematic approach to achieving promising results, as shown in Fig. 3. Each of the steps from our research methodology is further elaborated in detail below.

Dataset

In our study, we addressed the multi-class classification problem using the same dataset as the one used in the baseline

[3] https://www.kaggle.com/competitions/jigsaw-toxic-comment-classification-challenge/data

Table 1 Comparative analysis of state-of-the-art methods

References	Dataset	Feature Set	Method	Results
Elzayady et al. [18]	AraPersonality, SemEval 2020 Arabic offensive	TF-IDF	DT, XGBoost, SVM, LR, RF, LSTM, BiLSTM, CNN, GRU	Accuracy: 0.82
Saleh et al. [61]	Davidson et al. [16], Waseem, [73], Waseem and Hovy, [74]	Word2Vec, GloVe	BiLSTM, BERT	F1-score: 0.96
Mazari et al. [41]	Wikipedia Comments	FastText, GloVe	BERT, BiLSTM, BiGRU	F1-score: 0.99
Mittal and Singh. [45]	Online Tweets	Count Vectorizer	XGBoost, LIME, SHAP	F1-score: 0.94
Awal et al. [5]	Founta et al. [21], i Orts, [49], Mandl et al. [39], Founta et al. [22]	Contextual Embeddings	mBERT, XLM-R, MAML	ROC-AUC: 0.79
William et al. [75]	Online Tweets	Word2Vec, TF-IDF	AdaBoost, RF, LR, SVM	Accuracy: 0.79
Ali et al. [2]	Online Tweets	FastText	LSTM, GRU, BERT	F1-score: 0.73
Nagar et al. [46]	Founta et al. [21], Founta et al. [22]	Contextual Embeddings	VGAC	Accuracy: 0.85
Mehta and Passi, [42]	Online Tweets	TF-IDF	LR, RF, SVM, MNB, LSTM, BERT	Accuracy: 0.98
Akuma et al. [1]	Online Tweets	TF-IDF, BoW	LR, DT, KNN, NB	Accuracy: 0.92
Khanday et al. [32]	Online Tweets	TF-IDF, BoW	LR, MNB, SVM, DT, Bagging, AdaBoost, RF, SGB	Accuracy: 0.98
Khan et al. [31]	RUSA-19, RUSA	N-gram	CNN, RNN	Accuracy: 0.92
Rizwan et al [59]	RUSA-19	ELMO, FastText, LASER	LSTM, BERT, BiLSTM, CNN, XLM-R	F1-Score: 0.89
Proposed Work Model	**Dataset**	**Feature Set**	**Method**	**Results**
FAST-RNN, Prompt-Based Fine-Tuning	Andreassen Svanes and Seim Gunstad, [3](Resset, Twitter, Facebook)	Supervised FastText, Regularization, Hyperparameter Tuning, Generative Configurations	LSTM, GRU, CNN-LSTM, BiLSTM-GRU, mBERT, ELECTRA, FLAN-T5, scandi-BERT, Nor-T5, Nor-BERT, nb-BERT, LIME	Precision: 0.98, Recall:0.98, F1-score: 0.98, Accuracy: 0.98

study [3]. This dataset is categorized into five distinct classes: ’1’ for neutral, ’2’ for provocative, ’3’ for offensive, ’4’ for moderately hateful, and ’5’ for hateful. It was compiled from three social media platforms: Facebook (FB), Twitter, and Resset.[4] Furthermore, the baseline study provides a comprehensive explanation of each class label’s definition, ensuring clarity in the categorization of the data. The dataset exhibits a significant imbalance, with a predominance of neutral instances totaling 34,085, while hateful instances number only 250. This stark disparity highlights the need for an effective approach to accurately identify the relatively rare hateful instances. In Table 3, there are some examples of Norwegian

[4] https://inyheter.no/

Table 2 Comparative analysis of HS detection in low-resource dialects

References	Dataset	Hateful	Non Hateful	Evaluation
Bigoulaeva et al. [9]	German (GermEval-2018)	39.5%	60.5%	F1-Score: 0.98
Pereira-Kohatsu et al. [53]	HaterNet	58.5%	42.5%	Acc: 0.83
Ayo et al. [6]	Davidson et al. [16]	20%	80%	AUC: 0.96
García-Díaz et al. [24]	Spanish MisoCorpus 2020	58%	42%	Acc: 0.85
Fersini et al. [20]	Italian and English (AMI 2018)	41%	59%	Acc: 0.83
del Valle-Cano et al. [69]	Italian	58.5%	42.5%	F1-Score: 0.998
Batarfi et al. [7]	HateEval 2019	41%	59%	F1-Score: 0.998
Ptaszynski et al. [55]	Polish (PolEval-2019)	89.71%	20.24%	Acc: 0.90
Trajano et al. [67]	Brazilian Portuguese	31.5%	68.5%	F1-Score: 0.78

Table 3 Examples of Norwegian HS text with English translation

Category	Original text	English translation
Neutral	Alle har det bedre under/med Trump (bortsett fra hans mediamotstandere)	Everyone is better off under/with Trump (except his media opponents)
Provocative	Jeg visste ikke at @USER likte orgier og holdt til i Florida	I didn't know @USER liked orgies and lived in Florida
Offensive	Søt tenåring elsker å ri på en stor pikk	Cute teen loves to ride a big dick
Moderately Hateful	Innen 2050 er vi nok bde islamisert og kolonisert	By 2050 we will probably be both Islamized and colonized
Hateful	Men har dm ikkje anna og gjera Send dm langt inni dalstroka innafor og la dei beite der	But if they have nothing else to do Send them far inland and let them graze there

Table 4 Dataset distribution with class labels

Category	*Countof comments*
Neutral	34, 085
Provocative	4737
Offensive	1563
Moderately hateful	510
Hateful	250

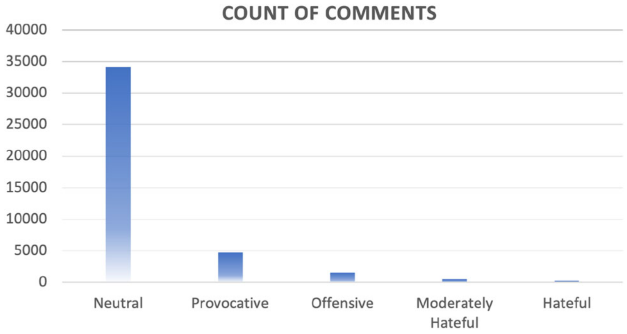


Fig. 4 Dataset distribution

HS instances along English translation. Table 4 and Fig. 4 illustrate the distribution of the dataset in terms of class labels and their count.

Data preprocessing

Data preprocessing is crucial in many ML and DL-based models for eliminating irrelevant text from the dataset, ensuring that the data are presented in a concise and appropriate format. In our study, we focused on two main columns: "text" containing all the comments, and "category" representing the five distinct classes. The preprocessing of "text" involved several key steps. First, we converted all uppercase letters to lowercase and removed non-essential characters, including ASCII symbols. The process also included tokenizing words and sentences and removing stop words.[5] To further refine our data, we used Python's RegEx library to filter and process elements like numbers, punctuation, and specific patterns, including email addresses, URLs, and phone numbers.

In the context of transformer-based models, our preprocessing approach was more specific, we conducted a limited set of preprocessing steps, deliberately excluding the removal of stop words, as it is not recommended under any circumstances. Another reason for limiting preprocessing for transformer-based models is to address the issue of syntactic ambiguity [64], which has been a significant drawback in previous DL-based techniques and models. Syntactic ambiguity occurs when words within a sentence might have several

[5] https://github.com/stopwords-iso/stopwords-no

interpretations depending on the context, making it a difficult problem to interpret.

Word embedding

Word embeddings offer numerical representations for textual inputs. FastText[6] embeddings provide several benefits compared to traditional word embeddings due to their ability to capture subword details and manage words not in the vocabulary more effectively. This feature makes FastText particularly advantageous for languages with complex morphology and diverse variations.

Equation 2 shows the mathematical formula to compute FastText word embeddings [44]

$$u_w + \frac{1}{|N|} \sum_{n \in N} x_n, \tag{1}$$

where

u_w: represents the vector for w in the embedding space.

$\frac{1}{|N|}$: is the fraction representing the average.

$\sum$: is used to sum over a set of vectors.

$n \in N$: specifies that we are summing over the set N.

x_n: represents the vector for the context words in the set.

FastText, a word representation tool developed by Facebook's research division, offers both unsupervised and supervised modes and features a comprehensive database of 2 million words from Common Crawl, each represented by 300 dimensions. Altogether, this library contains an impressive total of 600 billion word vectors. This word embedding method stands out with its distinctive methodology, which includes the use of manually crafted n-grams as features in addition to individual words [56].

FastText embeddings use morphological features, which enhances their effectiveness in vector representation and generalizability in a range of applications [68]. In this work, supervised FastText was used to focus on a categorical classification problem. It uses labeled training data to learn the associations between texts and labels, allowing for more accurate predictions on unseen data. This approach is advantageous in scenarios where the objective is to categorize text into predefined classes, as it provides context-based learning guided by the labeled examples. In contrast, unsupervised FastText focuses on learning word representations from a large corpus of unlabelled text, which is only useful for understanding word associations, and does not directly address the particular requirements of classification tasks. Moreover, unsupervised FastText cannot effectively identify the subtle and specific distinctions between different categories that are essential for accurate classification. In our experimentation, we trained the FastText model over 50 epochs, employing learning rates of 0.01.

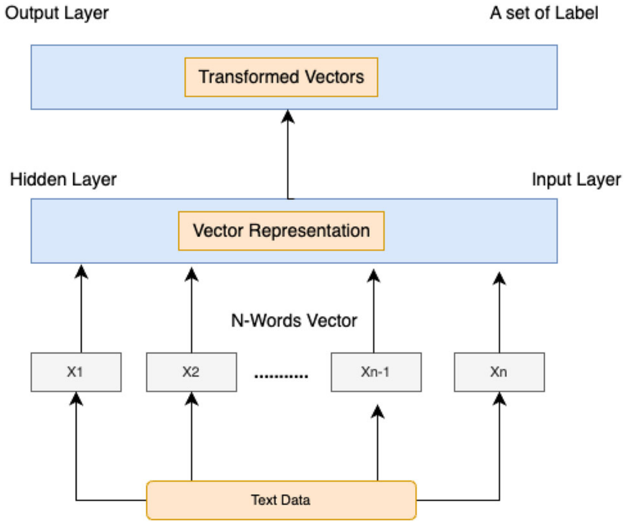


Fig. 5 FastText word embedding architecture

Modeling approaches

This section will detail the DL and transformer-based models utilized in this paper. It will provide an in-depth examination of each model's architecture and its application within our research framework.

DL-based models

In this paper, we implemented LSTM and its variant BiLSTM, along with GRU. These RNN-based models are known for their effectiveness in processing sequential data, with LSTM units being particularly adept at capturing long-term dependencies. BiLSTM enhances this capability by processing data in both forward and backward directions. GRU, similar to LSTM, efficiently manages sequence dependencies but with a simpler architectural design. Additionally, we explored a hybrid model, BiLSTM-GRU which is our proposed FAST-RNN, combining the strengths of both LSTM and GRU architectures with FastText embeddings. Furthermore, we performed CNN with the stacking of LSTM, leveraging CNN's ability to extract spatial features and LSTM's sequential data handling, offering a comprehensive approach to model complex patterns in data.

1. **FAST-RNN architecture:** The FAST-RNN architecture described in Fig. 6 is a sophisticated neural network

[6] https://fasttext.cc/

model proposed for categorical HS classification tasks. At its core, the model commences with an embedding layer that transforms text data into a dense sequence matrix of maximum sequence length ’m’. This matrix feeds into a BiLSTM segment comprising two model layers with 80 and 60 LSTM units, which processes the data bidirectionally to capture long-range dependencies in both forward and backward directions. Subsequently, the sequence is passed through a GRU segment with 60 GRU units, harnessing the model’s ability to focus on the most salient features of the input for classification while reducing computational complexity.

2. **Regularization:** Regularization is a technique in the learning algorithms that prevents overfitting, which occurs when a model performs well on training data but poorly on unseen test data [60]. The robust performance of the FAST-RNN model is considerably enhanced by the implementation of kernel L2 regularization, set at a lambda value of 0.01 for both the BiLSTM and GRU layers. L2 regularization is crucial for reducing the magnitude of the weights, which encourages the model to favor smaller weight values [35]. This approach serves a dual purpose: it reduces the likelihood of overfitting and strengthens the model’s ability to generalize, ensuring dependable performance on new, unseen datasets. The choice to utilize L2 instead of L1 regularization was intentional. L1 regularization tends to promote sparsity by driving some weights to zero [38], which, in our scenario, could lead to underfitting a limitation that became apparent during initial testing. Equations (2) and (3) are the mathematical formulas to calculate $L1$ and $L2$ regularization.
3. **Hyperparameter tuning:** In our thorough hyperparameter tuning process, we carefully fine-tuned the model’s parameters through a series of deliberate experiments. We trained the model for 10 epochs, a length of time chosen to ensure the model learned effectively without overfitting. This was finalized as the model’s loss stabilized over time. For the task of multi-class classification, we adopted the $cross - entropy$ loss function due to its well-established effectiveness. This loss function assesses the alignment between predicted probabilities and the actual class distribution, a critical metric for classification tasks of this nature. To optimize our model’s performance, we employed the $Adam\ optimizer$, known for its ability to dynamically adjust the $learning\ rate$. This adaptive learning rate mechanism enhances the model’s efficiency in exploring and converging toward optimal parameter values.

The name ’FAST-RNN’ highlights the model’s fast training and processing speed, along with its strong performance compared to other DL-based models in our study. We also tried training for 5 epochs and using L1 regularization, but the results were not as good. Five epochs did not give the model enough time to learn properly, and L1 regularization, which can reduce some weights to zero, was too obvious. Therefore, training for 10 epochs with L2 regularization was the best choice. It allowed the model to learn fully while still being able to perform well on new, unseen data, leading to the improved performance of the FAST-RNN. Table 5 illustrates the hyperparameters and configuration details of each DL-based model

$$L1(\mathbf{w}) = \lambda \sum_{i=1}^{n} |w_i|, \tag{2}$$

where

$\mathbf{w}$: is the weight vector of the model

λ: is the regularization coefficient

n: is the number of weights in the vector

w_i: is the ith weight in the weight vector.

$L1$ regularization adds the absolute value of the magnitude of the coefficients as a penalty term to the loss function. The absolute value makes this penalty term nonlinear in the weights, and thus, $L1$ regularization can lead to sparse solutions, with many coefficients being exactly zero

$$L2(\mathbf{w}) = \lambda \sum_{i=1}^{n} w_i^2; \tag{3}$$

$L2$ regularization adds the squared magnitude of the coefficients as a penalty term to the loss function. The squaring makes the penalty smoother and differentiable at $w_i = 0$. Unlike $L1$ regularization, $L2$ does not result in sparse models, as it typically does not force coefficients to be exactly zero (though they may be small).

Transformer-based models

The Transformer, introduced in 2017 by Vaswani et al. [70], is an NLP framework built for sequence-to-sequence tasks. It operates on the self-attention mechanism that efficiently handles long-range dependencies and consists of two primary components: an encoder and a decoder. The mechanism of self-attention within the Transformer can be mathematically formulated as follows:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK_i^\top}{\sqrt{d_k}}\right)V_i, \tag{4}$$

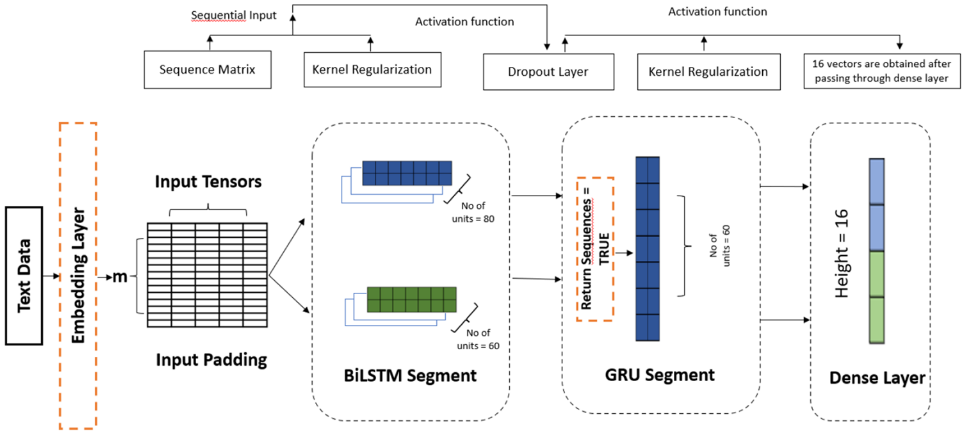


Fig. 6 Proposed FAST-RNN architecture

Table 5 Configuration details for DL models

Model	Model layer	Dense layer	Dropout layer	Pooling layer	Regularization	Epochs	Function	Loss
LSTM	3	2	2	–	L2	10	Softmax	Categorical entropy
GRU	3	2	2	–	L2	10	Softmax	Categorical entropy
CNN-LSTM	3	2	2	2	L2	10	Relu	Categorical entropy
BiLSTM-GRU	3	2	2	–	L2	10	Relu	Categorical entropy

where

Q: is the loss to minimize

K: is the key matrix

V: is the value matrix

d_k: is the dimension of the key vectors

N: is the length of the input sequence

i: is the index of the query vector.

This paper utilizes multilingual transformers, with a focus on optimizing their hyperparameters. Unlike previous language models such as RNNs, which faced limitations in computational and memory capacities for generative tasks, transformers represent a substantial improvement. In our study, we used the Norwegian HS text dataset for which multilingual text classification transformers like mBERT, ELECTRA, FLAN-T5, along with Norwegian LMs NorBERT, ScandiBERT, nbBERT, and Nor-T5, were utilized.

Multilingual transformers

mBERT

BERT, a transformer model, underwent self-training on an extensive, multilingual dataset. This implies it was trained solely using raw text, without any human-labeled data, leveraging publicly accessible data and an automated method for generating inputs and labels from the text. In contrast, mBERT, a specialized version of BERT, was pre-trained specifically on the largest Wikipedia articles across 104 languages. It employed a Masked Language Modeling (MLM) approach for its training [17].

ELECTRA

In BERT's MLM pretraining, input tokens are replaced with a **[MASK]** placeholder, and the model learns to predict the original tokens. Electra, however, introduces a more efficient method called replaced token detection. Unlike BERT, Electra replaces some tokens with plausible alternatives from a smaller generator network, and a discriminative model is trained to identify whether each token in the input has been replaced or not. The generator part in Electra assigns probabilities to the generation of specific tokens x_t using a softmax layer [14].

FLAN-T5

FLAN-T5,[7] an extension of the Text-to-Text Transfer Transformer (T5) model [57], represents a significant advancement in NLP. Developed for instruction fine-tuning, FLAN-T5 is trained across various tasks, enhancing its adaptability and efficiency in text-to-text operations [13]. Its proficiency in summarizing dialogs and classifying text makes it invaluable for any real-world applications. Additionally, FLAN-T5

[7] https://huggingface.co/docs/transformers/model_doc/flan-t5

Table 6 Configuration details for transformer-based models

Model	Class	Batches	Lr	Epoch
mBERT	BertTokenizer	32	2e–5	5
ELECTRA	ElectraTokenizer	32	2e–3	5
FLAN-T5	AutoTokenizer	32	2e–5	5

Table 7 Configuration details for Norwegian transformer-based models

Model	Class	Batch	Lr	Epoch
nb-BERT	SequenceClassification	32	2e–5	5
Nor-BERT	SequenceClassification	32	2e–6	5
Nor-T5	Seq2SeqLM	32	2e–3	5
scandi-BERT	SequenceClassification	32	2e–5	5

excels in text classification. It automates the categorization of text into predefined classes, such as Sentiment Analysis (SA), spam detection, or topic modeling. Table 6 presents the configuration and hyperparameters of the multilingual transformer-based models utilized in this study.

Norwegian LMs

Recently, significant advancements have been made in Norwegian LMs. A. Kutuzov et al. [37] introduced NorBERT, available in various sizes and trained on the Norwegian Academic Corpus (NAK) and Norwegian Wikipedia. NorBERT$_2$ uses data from the Norwegian section of mC4 and the NCC's public part. P.E. Kummervold et al. [36] developed NB-BERT models: NB-BERT$_{\text{base}}$, which builds upon mBERT, and NB-BERT$_{\text{large}}$, independently trained on the complete NCC corpus. Additionally, Scandinavian BERT (Scandi-BERT), covering Danish, Norwegian, Icelandic, Faroese, and Swedish texts, has over 60% Norwegian content from NCC. Recently, two novel Norwegian LMs, Nor-T5[8] and North-T5,[9] were proposed by Samuel et al. [62]. Nor-T5, and North-T5 transformer models are designed for Norwegian and Scandinavian sequence-to-sequence tasks. These models were evaluated against multilingual T5 models and a series of specialized North-T5 models, which are essentially mT5 models further fine-tuned specifically on Norwegian data. This comparison aims to assess their effectiveness in handling Norwegian language tasks. Table 7 presents the configuration and hyperparameters of the Norwegian transformer-based models utilized in this study.

Generative configuration

In the process of refining the proposed multilingual transformers and Norwegian LMs, we implemented substantial modifications to the hyperparameters, which resulted in noticeable improvements in our outcomes. These adjustments encompassed the exploration of diverse batch sizes, learning rates, and epochs. Additionally, we also employed generative configuration parameters, which are additional parameters that the model utilizes during training. These parameters are invoked during the inference phase, providing us with control over factors such as the maximum token count in the generated output and the level of creativity in the text. These techniques include random sampling methods like *top-k* and *top-p*, which impose constraints on randomness and increase the likelihood of producing creative and diverse outputs [54]

Top-k sampling involves choosing the k most likely words from the model's probability distribution for the next word. The process is defined by the following formula:

$$P(w) = \begin{cases} \frac{e^{(P(w))}}{\sum_{w'} e^{(P(w'))}} & \text{if } w \text{ is in the top-}k \\ 0 & \text{otherwise,} \end{cases} \tag{5}$$

where

w: is the word being sampled,

$P(w)$: is the probability of word,

V: is the vocabulary of possible words.

Top-p sampling selects the minimum number of words needed to have a cumulative probability exceeding a predefined threshold p. Following is the mathematical expression to calculate the top-p sampling [29]:

$$P(w) = \frac{1}{\sum_{w' \in V : P(w') \geq p} P(w')} \tag{6}$$

where

$\sum\limits_{w' \in V : P(w') \geq p} P(w')$: sum of probabilities.

Furthermore, we integrated an additional set of configuration parameters in our method, specifically the "temperature" parameter. This parameter significantly influences the model's calculated probability distribution used in predicting the subsequent token. Essentially, the temperature value serves as a scaling factor within the softmax layer of the transformer models. A higher temperature setting increases

[8] https://huggingface.co/ltg/nort5-large

[9] https://huggingface.co/north/t5_base_NCC

Table 8 Generative configuration details for transformer-based models

Model	Top-k	Top-p	Temperature
FLAN-T5$_{\text{small}}$	5	0.5	0.3
FLAN-T5$_{\text{base}}$	5	0.2	0.3
Nor-T5$_{\text{small}}$	7	0.5	0.3

the randomness in the generated output, while a lower temperature value reduces the range of possible words in the generated text [54]. Following is the mathematical expression for random sampling with temperature [29]:

$$P(w) = \frac{exp^{(P(w)/\tau)}}{\sum_{w'} exp^{(P(w')/\tau)}} \tag{7}$$

where

τ: parameter controlling distribution diversity

$\sum\limits_{w'} exp^{(P(w')/\tau)}$: normalization factor.

Table 8 provides an overview of the generative configuration employed during the fine-tuning of our models, including details on class type and the tokenizer used.

Transformer-based models vary in their capabilities with generative configurations, as exemplified by mBERT and ELECTRA, which have been designed for classification tasks rather than text generation. This specialization accounts for their inability to work with generative parameters. In comparison, FLAN-T5 and Nor-T5, both are variants of the T5 transformer and can be used for text generation, summarization, and translation tasks. This functionality is also influenced by their class type; both Nor-T5 and FLAN-T5 belong to the "Seq2SeqLM" class, a category not applicable to other transformer-based models like BERT and ELECTRA.

Prompt based fine-tuning

In traditional ML, models are trained on a large dataset to learn a task. However, in prompt-based learning, the transformer-based models are given a natural language prompt or a set of instructions that guides them to perform a specific task without extensive training. This approach utilizes the pre-trained knowledge of these transformers and adapts it to new tasks through carefully crafted prompts. In our study, we employed two different types of prompt-based fine-tuning: few-shot and full fine-tuning.

Few-shot fine-tuning

Few-shot fine-tuning is a process that entails training a model on a small (few examples), task-specific dataset, in contrast to traditional fine-tuning which typically requires a larger dataset. In this method, the model is given a limited number of examples with natural language prompt along with the desired outcome. These examples aid the model in adjusting its responses to better suit the particular task's requirements. Few-shot fine-tuning proves highly beneficial when we have limited resources with restricted task-specific data and aim to ensure the model generalizes effectively from these limited examples.

Full fine-tuning

Full fine-tuning involves training the model on a substantial dataset. This dataset is usually specific to the task or domain the model is intended to perform in. Full fine-tuning is more resource-intensive compared to few-shot fine-tuning and it requires more computational power and time, as the model needs to be trained over a larger set of data. This approach offers the advantage of highly specializing the model for the fine-tuned task.

- *Natural language prompt:* In our study, we chose to use FLAN-T5 and Nor-T5 architectures for prompt-based learning, because models like mBERT, ELECTRA, NB-BERT, and several others are not well suited for this specific approach. The primary reason is that these models are typically designed for contextual language understanding, where they predict the next word or token in a sentence based on the surrounding context. They do not inherently support prompt-based learning, which requires the ability to generate responses or perform actions based on explicit instructions or prompts provided by the user. FLAN-T5 and Nor-T5, on the other hand, have been specifically designed and fine-tuned for natural language prompt-based tasks, making them more suitable choices for this research. Our transformer-based methodology centered around the natural language prompt: '*Please classify the following sentence into just one of the mentioned categories: neutral, provocative, offensive, moderately hateful or hateful.*' This prompt was a key element in our exploration of different fine-tuning approaches, namely few-shot and full fine-tuning.

The provided Algorithm 1 defines a function that prepares data for fine-tuning a language model. It generates prompts for classification tasks by combining a fixed starting prompt with text samples from a dataset and an ending prompt. These prompts are tokenized using a tokenizer, and the resulting **input_ids** are stored in **dataset_dict['input_ids']**. Addition-

Algorithm 1 Preparing prompt for fine-tuning

```
procedure FUNCTION($dataset\_dict$)
    $prompt \leftarrow$ "$natural\ language$" + "$tweet$"
    $end\_prompt \leftarrow dataset\_dict['label']$
    $input\_ids \leftarrow tokenize: prompt$
    $labels \leftarrow tokenize: end\_prompt$
    $dataset\_dict['input\_ids'] \leftarrow input\_ids$
    $dataset\_dict['labels'] \leftarrow labels$
    return $dataset\_dict$
end procedure
```

Table 9 Training arguments for prompt-based fine-tuning

Parameters	Language model
Learning rate	1e–8
Num_train_epochs	5
Evaluation_strategy	'epoch'
Weight_decay	0.01
Per_device_train_batch_size	16
Logging_steps	1
Optim	'adamw_torch'

ally, the labels in **dataset_dict['labels']** are tokenized and stored in **dataset_dict['labels']**. All these conversions have been conducted using **PyTorch tensors**. The label variable consists of one of the class categories in our HS dataset. The tokenizer utilized in this algorithm is the identical tokenizer that was employed during the model's pretraining phase. The final object **dataset_dict** is then passed to the learning algorithm for the training, as mentioned in Table 9, which represents the configuration and hyperparameter details for the few-shot and full-instruction fine-tuning training processes.

In the training of the transformer-based model, a set of carefully chosen hyperparameters was utilized to fine-tune the learning process. A learning rate of 1e–8 was selected, maintaining a balance between convergence speed and stability. The model was subjected to training over 5 epochs, ensuring adequate exposure to the data while avoiding overfitting. The evaluation was conducted at the end of each epoch, allowing for consistent monitoring of the model's performance. To prevent the model's weights from growing too large and overfitting, a $weight\_decay$ of 0.01 was applied. The $batch\_size$ was set to 16 per device to optimize memory usage and computational efficiency. $Logging\_steps$ were set to 1 to ensure that the training process was transparent and that the progress could be closely tracked. Finally, the '$adamw\_torch$' optimizer was chosen for its ability to automatically adjust the learning rate and for being well suited for transformer-based models.

Table 10 Results of DL-based models

Model	*P*	*R*	Acc	*F*	Auc_Roc
LSTM	0.97	0.97	0.97	0.97	0.99
GRU	0.97	0.97	0.97	0.97	0.99
CNN-LSTM	0.96	0.96	0.96	0.96	0.99
BiLSTM-GRU	0.98	0.98	0.98	0.98	0.99

Results and discussion

For the evaluation of the results, standard metrics of accuracy, precision, recall, f1-score, and auc_roc were employed to quantify the model's classification performance. The dataset was divided into a training and testing split of 80% and 20%, respectively

$$\text{Accuracy} = \frac{T_P + T_N}{T_P + T_N + F_P + F_N} \tag{8}$$

$$\text{Precision} = \frac{T_P}{T_P + F_P} \tag{9}$$

$$\text{Recall} = \frac{T_P}{T_P + F_N} \tag{10}$$

$$\text{F1-Score} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{11}$$

DL-based models

The evaluation scores for DL-based models, employing FastText embeddings, are displayed in Table 10.

By analyzing 10, we can see that the FAST-RNN model exhibits a precision and recall of 0.98. Precision is a critical measure when the consequences of false positives are significant. A precision of 0.98 means that when the FAST-RNN model predicts an instance as positive, it is correct 98% of the time. This high precision indicates that the model is highly reliable in its positive predictions, making very few mistakes in this regard. With a recall of 0.98, the FAST-RNN model can correctly identify 98% of all actual positive instances. This suggests that it is particularly effective at capturing the relevant signals from the data without missing many actual positives. Moreover, the high accuracy score of 0.98 reflects the overall rate at which the model makes correct predictions for both positive and negative classes. This balanced performance is mirrored in the weighted F1-score, which is the harmonic mean of precision and recall, indicating that the model maintains a strong balance between precision and recall across all classes. Finally, an AUC-ROC score of 0.99 indicates an excellent ability of the model to discriminate between the positive and negative classes. A score close to 1.0 means that the model has a high true-positive rate and a low false-positive rate across different thresholds.

Table 11 Classificatioin report for FAST-RNN

Category	*P*	*R*	*F*	*Support*
Neutral	0.99	0.98	0.99	6813
Provocative	0.89	0.93	0.91	951
Offensive	0.92	0.99	0.95	301
Moderately hateful	0.98	0.88	0.93	109
Hateful	0.96	0.98	0.97	54

In comparison to the FAST-RNN model, the other DL-based models like LSTM, GRU, and CNN-LSTM also show robust performance with all metrics ranging from 0.96 to 0.97. Both LSTM and GRU models match the FAST-RNN's precision, recall, and accuracy, indicating their strong predictive capabilities, while the CNN-LSTM lags slightly behind but still demonstrates high scores of 0.96. Each model shows remarkable ability in sequence processing tasks, with the FAST-RNN slightly outperforming the rest, likely due to its hybrid architecture that leverages the strengths of both LSTM and GRU layers. The ROC-AUC score of 0.99 for all models, including FAST-RNN, LSTM, GRU, and CNN-LSTM, indicates a high degree of predictive accuracy, reflecting their strong ability to rank predictions correctly.

Table 11 presents the classification report of the proposed FAST-RNN model which shows the best performance in predicting each class within the test data. This is particularly notable in its prediction of hateful instances, which are a minority in the dataset. Despite this, our model achieved an impressive 97% F1-score in accurately identifying these instances. The model shows remarkable precision and recall in the 'Neutral' category, with scores of 0.99 and 0.98 respectively, suggesting a decent performance in identifying non-inflammatory content, which is often the bulk of data and sets the baseline for model performance. In more nuanced categories, such as 'Provocative' and 'Offensive,' the model exhibits precision scores of 0.89 and 0.92, with recall scores of 0.93 and 0.99, indicating its effective differentiation between subtly differing sentiments. The 'Moderately Hateful' category, despite having fewer instances, also sees a high F1-score of 0.93, underlining the model's capability to discern complex emotional nuances in a text. These results collectively highlight the robustness of the FAST-RNN model in handling both clear-cut and borderline cases, ensuring that it performs reliably across a diverse range of textual sentiments.

Figures 7 and 8 indicate a stable convergence, with the validation metrics closely tracking the training metrics across epochs. The graphs demonstrate a stable convergence and indicate that the validation scores are close to the training scores throughout the training epochs. The close alignment between training and validation accuracy, alongside a consistent decrease in loss for both training and validation, suggests that the model is learning generalizable patterns rather than overfitting the training data. This balance between learning and generalization, especially given the limited number of hateful instances, underscores the model's performance and generalizability. Figure 9 illustrates the confusion matrix multi-class classification HS detection using FAST-RNN.

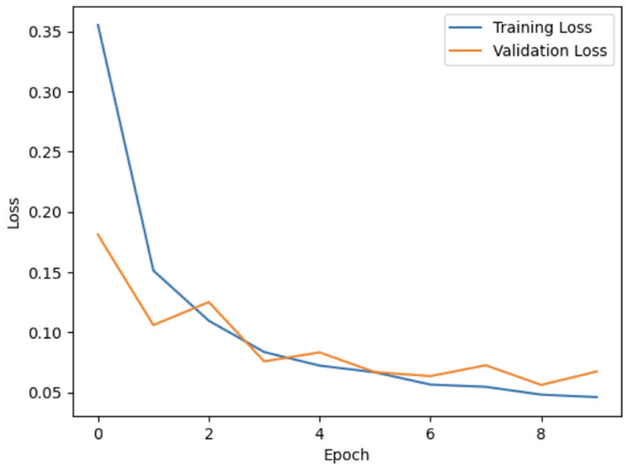


Fig. 7 Training and validation loss curve—FAST-RNN

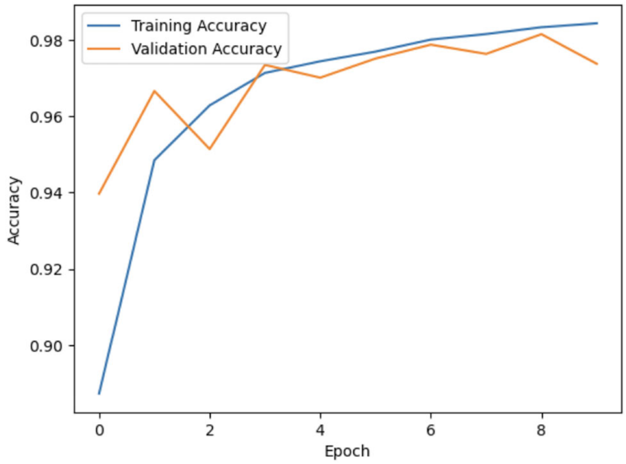
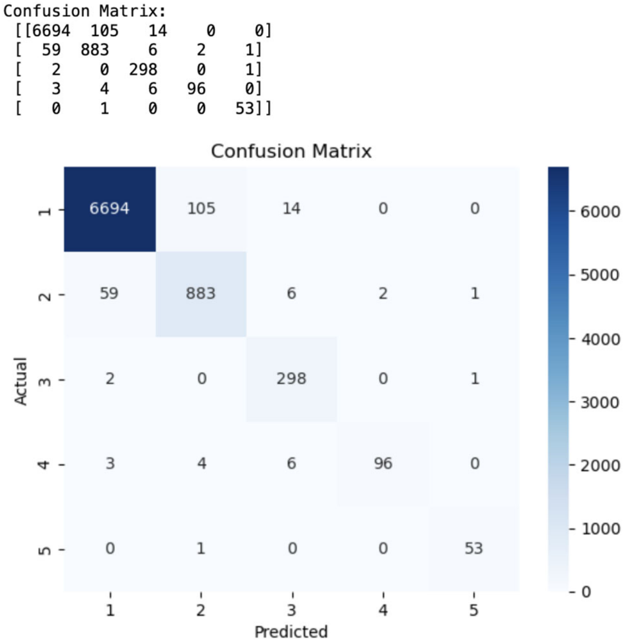


Fig. 8 Training and validation accuracy curve—FAST-RNN

Transformer-based models

Table 12 presents the results achieved from the multilingual transformers as well as the Norwegian transformer-based models.

$\text{ELECTRA}_{\text{base}}$ and ELECTRA_{large} show uniform performance across four metrics, each with a precision of 0.69, recall of 0.83, accuracy of 0.83, and a f1-score of 0.75. This indicates that scaling up the ELECTRA size from base to large does not impact the performance for these specific tasks. mBERT, with a precision of 0.78, is noteworthy for



```
Confusion Matrix:
 [[6694  105   14    0    0]
 [  59  883    6    2    1]
 [   2    0  298    0    1]
 [   3    4    6   96    0]
 [   0    1    0    0   53]]
```

Fig. 9 Confusion matrix—FAST-RNN

Table 12 Analysis of the results: transformer-based models

model	P	R	Acc	F	Auc_Roc
mBERT	0.78	0.82	0.82	0.79	0.81
ELECTRA$_{\text{base}}$	0.69	0.83	0.83	0.75	0.80
ELECTRA$_{\text{large}}$	0.69	0.83	0.83	0.75	0.80
scandi-BERT	0.79	0.81	0.81	0.80	0.81
nb-BERT$_{\text{base}}$	0.79	0.81	0.81	0.80	0.81
nb-BERT$_{\text{large}}$	0.81	0.81	0.81	0.81	0.82
Nor-BERT$_{\text{small}}$	0.77	0.83	0.82	0.79	0.82
Nor-BERT$_{\text{base}}$	0.78	0.83	0.83	0.80	0.85
Nor-BERT$_{\text{large}}$	0.80	0.82	0.83	0.81	0.85

its relatively high ability to produce relevant results over the total number of results it provides (precision), while its recall of 0.82 shows that it is quite competent at identifying relevant instances from the dataset. Its accuracy is at 0.82 and f1-score at 0.79 which is more than both ELECTRA variants and similar to Nor-BERT$_{\text{small}}$, suggesting a well-rounded performance. Scandi-BERT and nb-BERT$_{\text{base}}$, both with precision at 0.79 and recall at 0.81, demonstrate a similar capability in correctly classifying instances, and both maintain an accuracy of 0.81. The f1-score for these models stands at 0.80, indicating a robust balance between precision and recall. A performance improvement is noted when comparing nb-BERT$_{\text{base}}$ to nb-BERT$_{large}$, with the latter achieving a precision of 0.81, which is the highest precision score among all the models listed, matching its recall, accuracy, and f1-score.

Table 13 Analysis of the results with few-shot fine-tuning and generative configuration

model	P	R	Acc	F
FLAN-T5$_{\text{small}}$	0.69	0.80	0.80	0.74
FLAN-T5$_{\text{base}}$	0.78	0.80	0.80	0.79
Nor-T5$_{\text{small}}$	0.77	0.77	0.77	0.77

Nor-BERT variants show a progression in performance with size increment. Nor-BERT$_{\text{small}}$, with a precision of 0.77 and recall of 0.83, provides a good improvement with 0.82 accuracy and f1-score of 0.79. The Nor-BERT$_{\text{base}}$ model shows a slight improvement in precision to 0.78 while maintaining a similar recall. The highest f1-scores are observed with Nor-BERT$_{large}$ and nb-BERT$_{large}$. These models excel at not only identifying relevant instances but also at minimizing the number of irrelevant instances that are incorrectly identified as relevant. The performance enhancement from base to larger models is predominantly a result of the differences in their sizes (number of parameters). Generally, models with a high number of parameters can demonstrate better performance over those with a smaller parameter count. The analysis of results in transformer-based models suggests that the performance of transformers might not be as impressive relative to RNN models for specialized tasks such as multi-class HS detection. This could be due to the transformers' design, which is optimized to identify broad patterns in large datasets rather than the more nuanced patterns that specialized tasks might require. The AUC-ROC scores tell how well each model differentiates between the positive and negative classes: Nor-BERT_large, with an AUC-ROC of 0.85, is most effective, suggesting it has a greater likelihood of correctly identifying true positives and true negatives. mBERT's score of 0.81 and ELECTRA_base's score of 0.80, while lower, still represent a strong predictive ability, with only a marginal difference in classification confidence when compared to Nor-BERT_large.

Table 13 highlights the evaluation scores of models subjected to few-shot fine-tuning with a generative configuration mentioned in Table 8. Here, FLAN-T5$_{\text{small}}$ and FLAN-T5$_{\text{base}}$ demonstrate similar performance in terms of recall and accuracy, while the base model exhibits slightly better performance with precision and f1-scores of 0.78 and 0.79, respectively, which is a noticeable increase from the small variant, meaning it is more precise in its predictions. Nor-T5$_{\text{small}}$ maintains comparable recall, precision, accuracy, and f1-scores of 0.77, indicating a balanced performance to predict positive instances correctly and to identify the most positive instances.

Transitioning to full fine-tuning in Table 14, all models exhibit enhanced f1-scores, indicative of improved predictive relevance and balanced precision-recall, as compared to

Table 14 Analysis of the results with full fine-tuning and generative configuration

model	P	R	Acc	F
FLAN-T5$_{\text{small}}$	0.76	0.80	0.80	0.77
FLAN-T5$_{\text{base}}$	0.82	0.82	0.83	0.80
Nor-T5$_{\text{small}}$	0.77	0.78	0.77	0.78

the performance of these models in Table 13. FLAN-T5$_{\text{base}}$ records the highest precision increase to 0.82, an accuracy score of 0.83, and an f1-score of 0.80, indicating a balance between precision and recall and also suggesting that full fine-tuning significantly refines the model's predictive accuracy and overall performance. FLAN-T5$_{\text{small}}$ also shows marginal gains in precision and f1-score, underscoring the benefits of a more extensive fine-tuning process. In the case of Nor-T5$_{\text{small}}$, it exhibits a slight improvement as compared to few-shot fine-tuning.

Comparing the results of both these tables, it is evident that full fine-tuning combined with generative configurations yields improved model performance. Additionally, models that have a greater number of parameters tend to surpass the performance of those with fewer parameters.

Comparison of the results with the state-of-the-art

In this section, we compare our results with the baseline method [3]. The baseline study employed an unsupervised FastText model, which generally is less suited for categorical classification tasks. In comparison, our supervised FAST-RNN model when implemented with optimal regularization and hyperparameter tuning outperformed the baseline in terms of both accuracy and macro F1-score. The FAST-RNN model achieves a Macro F1-Score of 0.97 for the 'Hateful' category, far surpassing the baseline models' scores of 0.08 for BiLSTM and 0.06 for CNN-LSTM. Similarly, in the 'Offensive' category, our model attained a score of 0.95, significantly higher than the baseline scores of 0.27 and 0.35, respectively. Even in the 'Provocative' category, which often contains more subtle and nuanced language, our FAST-RNN model reached a score of 0.91, outperforming the baseline's 0.61 and 0.59. The employment of explainable AI through LIME has provided additional validation by elucidating the model's decision-making process, thereby granting further credibility to our findings, particularly in the challenging area of HS detection in low-resource language scenarios. Table 15 presents a comparison of the macro F1-scores between the baseline and our proposed model FAST-RNN, specifically focusing on non-neutral categories provocative, offensive, moderately hateful, and hateful.

Interpretability modeling with LIME

Local Interpretable Model-Agnostic Explanations (LIME) is a technique designed for local understanding and evaluating the predictions made by any learning algorithm. It provides insights into how a model's predictions align with the specific requirements of the given task. This method is particularly valuable in contexts where understanding the decision-making process of a model is as important as the accuracy of its predictions [8]. The equation for LIME aims to find an interpretable model The equation for LIME aims to find an interpretable model $\hat{g}$ from a class of models G that minimizes the loss $\mathcal{L}$ between the predictions of g and the complex model f, considering the locality kernel π_x, and $\Omega(g)$ is the complexity of the interpretable model g with lower complexity preferred for better interpretability, while also maintaining simplicity

$$\hat{g} = \arg\min_{g \in G} \mathcal{L}(f, g, \pi_x) + \Omega(g). \tag{12}$$

In this study, we examine the rationale behind the predictions made by our proposed FAST-RNN model by utilizing LIME. The utterances deemed most hateful were divided into two groups: moderately hateful and hateful. This classification was based on whether the statements provoked actions of violence or discrimination. The categorization into moderately hateful and hateful was influenced by definitions established in studies by Sanguinetti et al. [63] and Sharma et al. [66]. According to these definitions, utterances explicitly encouraging violence or discriminatory actions were classified as severely hateful. The degree of severity remained unchanged whether the authors merely justified such actions, expressed a desire for their occurrence, or showed a willingness to partake in them. Consequently, any utterances that in any manner incited such actions were included in this most severe category. Following are the definitions of all categories in our dataset.

1. **Hateful:** Hateful utterances are utterances that are partly or wholly motivated by hate or negative attitudes towards groups or individuals based on ethnicity, religion, sexuality, gender, age, political views, social status, or disabilities and which encourage violent actions based on this.
2. **Moderately Hateful:** Moderately hateful utterances are utterances that are partly or fully motivated by hate or negative attitudes towards groups or individuals based on ethnicity, religion, sexuality, gender, age, political views, social status, or disabilities. The utterances do not call to action but still violate the integrity and disparage a group or individual's dignity.

Table 15 Comparative macro F1-scores of baseline with proposed FAST-RNN and transformers

Baseline Models				
Model	Provocative	Offensive	Moderately Hateful	Hateful
BiLSTM	0.61	0.27	0.11	0.08
CNN-LSTM	0.59	0.35	0.05	0.06
Proposed FAST-RNN Model				
Model	Provocative	Offensive	Moderately Hateful	Hateful
FAST-RNN	0.91	0.95	0.93	0.97

3. **Offensive:** An utterance is defined as offensive if it contains hurtful, derogatory, or obscene comments, either directed towards an individual or a group.
4. **Provocative:** A provocative utterance contains aggressive language to express an opinion or can be perceived as inappropriate. This includes the use of profane words, patronizing language, or the use of irony and sarcasm to lower the credibility of an opponent.
5. **Neutral:** An utterance which contains neutral language and is a factual contribution to the debate.

For a clearer understanding, we have implemented LIME on two examples from each class, as detailed in Table 16. In the provided LIME visualization in Fig. 10, the model's decision to categorize the text as 'hateful' is strongly influenced by specific terms that resonate with the defined characteristics of HS. Words like "bomber" (bomber) and "kutter" (cut) are particularly weighted, suggesting a violent disposition towards the mentioned group, in this case, individuals of Pakistani ethnicity. The term "sendt," translating to "send," further contributes to this categorization as it implies an actionable directive, which is a crucial aspect of the classification criteria for HS within the dataset. This term indicates not just a negative sentiment but an incitement to take negative action based on ethnicity, aligning with our definition of hate speech. The model's identification of these terms reflects its capability to recognize and classify language that promotes hate-motivated actions against specific groups, thus validating the effectiveness of the algorithm in detecting hate speech as defined by our criteria.

Similarly, in Fig. 11, the LIME analysis elucidates the model's inference process, which strongly suggests the text as hateful with an 87% probability. Central to this classification is the verb "sende", which implies an action, and in the given context, an action against the "somaliere" (Somalis) community. The sequence of highlighted words constructs a narrative supporting the removal of this group from the country, which is a clear indication of HS according to the definition. The model's high weighting on these specific terms indicates its capability to parse and understand the intent behind the words, recognizing the call to action that constitutes HS within our dataset parameters. The model's interpretation aligns with the dataset's criteria, demonstrating its nuanced ability to detect incitements to discriminatory actions based on ethnicity.

In Fig. 12, the LIME visualization isolates significant terms that collectively contribute to the text being classified as "moderately hateful." Terms like "klankultur" (clan culture), "avskyelig" (disgusting), and "press" (pressure) are weighted heavily, indicating a perception of societal burden. In this case, the language implies a negative opinion about the influence of Pakistani individuals on public services and society. Though the statement does not directly encourage harmful actions, it crosses the line of respectful conversation by disrespecting a particular ethnic group. This portrayal of an entire community as a stressor on educational and health services, marked by terms that imply revulsion and financial burden, aligns with the class category "4" classification definition in the dataset. This nuanced detection by the model highlights its ability to discern between outright calls to action characteristic of more severe HS and the insidious nature of moderately hateful language that erodes respect for communal harmony and individual dignity.

Similarly, in 13, the highlighted word "Islamisert" and "kolonisert" dominate the narrative of the model's interpretation with high probability scores, implying societal transformation or takeover, which is interpreted as negative. The text projects a future scenario where the influence of the Muslim community is portrayed in terms of colonization and Islamization, terms that carry a heavy historical and negative connotation. Despite the absence of a call to action, the language used disparages the community's dignity and integrates notions of cultural subversion, which are characteristic of 'moderately hateful' content as defined in the dataset.

For the first visualization 14, the model strongly identifies the term top term "feita" (ugly/fat) with the highest weight as offensive which is directed at individuals like 'bergens' and 'solberg'. This word, particularly when used to describe a person, carries a negative connotation that is both hurtful and derogatory. The term "slengt" (thrown) can also imply a dismissive or contemptuous attitude, further supporting the offensive classification.

For the second visualization 15, the model has highlighted explicit terms such as "pikk" (dick/cock), "elsker" (loves), "ri" (tear/rip), and the phrase "stor pikk" (big dick/cock), which are sexually explicit and considered obscene. The

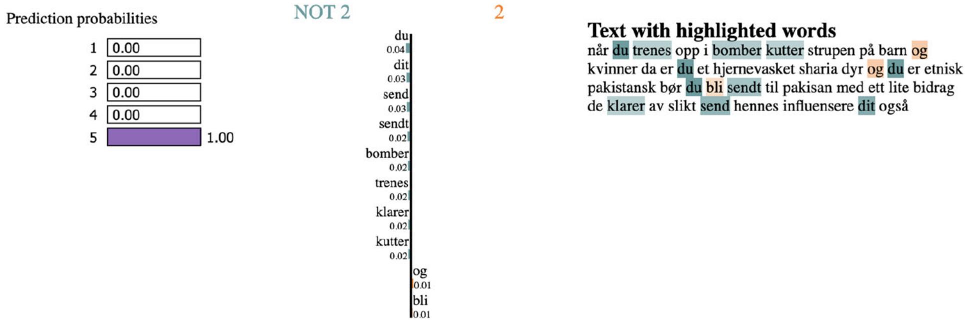


Fig. 10 Example 1: hateful instance visualization with LIME

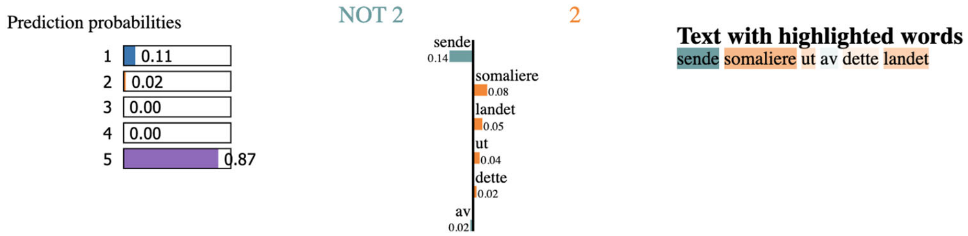


Fig. 11 Example 2: hateful instance visualization with LIME

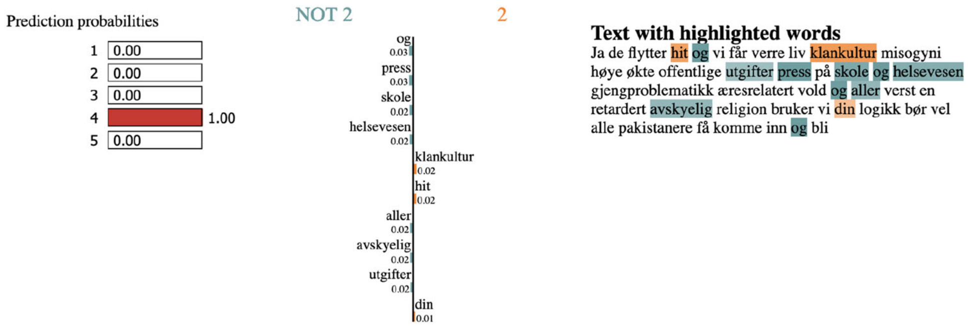


Fig. 12 Example 1: moderately hateful instance visualization with LIME

use of these terms in the context provided is inappropriate, derogatory, and clearly intended to be offensive, especially when directed at an individual or group. This kind of language falls under the offensive category, because it is hurtful and violates social norms of decency.

In Fig. 16, the LIME visualization highlights the use of the terms "sannheter" (truths), "nyanser" (nuances), "ufeilbarlige" (infallible), and "fremstillinger" (representations), which together create a narrative that can be perceived as dismissive and patronizing. These terms, particularly in the context provided, suggest an ironic or sarcastic critique of media or societal understanding, which may be provocative to those who hold opposing views. The use of these terms in a way that challenges the subject's credibility or oversimplifies complex issues fits the definition of provocative content in the dataset. The model's detection of these nuanced uses of language highlights its sensitivity to the subtleties of provocative speech, which is not overtly aggressive but can still incite strong reactions from an opponent's standpoint.

In Fig. 17, the highlighted word "orgier" (orgies), with its significant weight, stands out as a term that traditionally relates to excessive, unrestrained, or scandalous sexual activity. When mentioned in conjunction with "Florida", a place known for its vibrant nightlife and cultural diversity, it might suggest a provocative statement about certain behaviors or events in that location. The model's 100% confidence

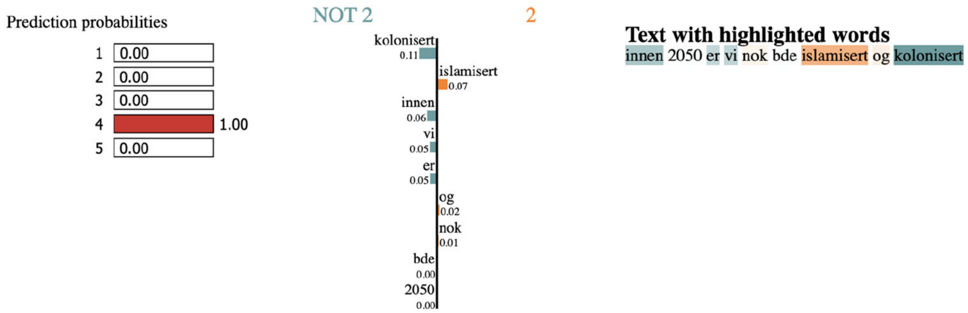


Fig. 13 Example 2: moderately hateful instance visualization with LIME

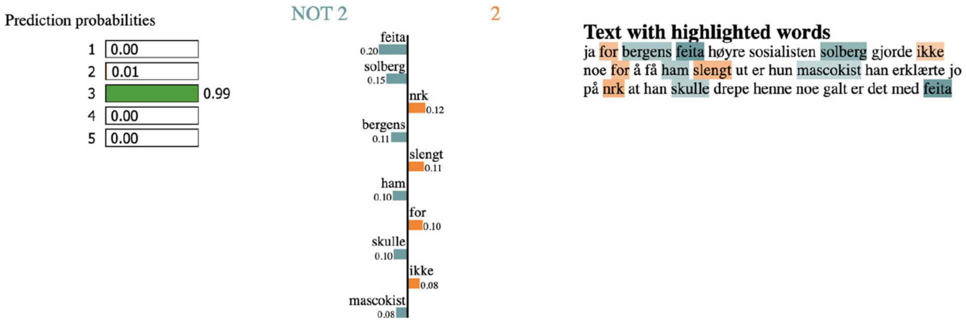


Fig. 14 Example 1: offensive instance visualization with LIME

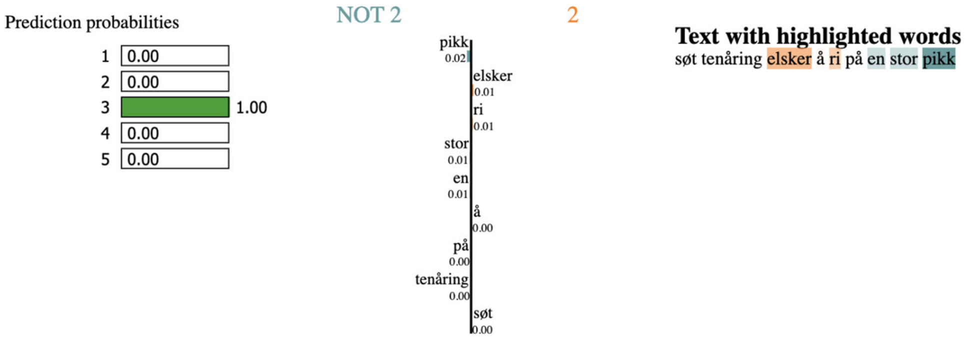


Fig. 15 Example 2: offensice instance visualization with LIME

in classifying this utterance as "provocative" indicates that the language used here is likely meant to shock or provoke a response from the audience. It fits the definition of provocative content that includes aggressive language or statements that can be perceived as inappropriate, such as the use of profane words or the depiction of scandalous behavior. While the statement does not contain outright offensive or hateful language, the implication of the terms used is sufficient to provoke or challenge societal norms, thereby justifying its classification within the dataset.

Figures 18 and 19 highlight the examples of the neutral class from our dataset. In Fig. 18, none of the words are assigned any significant probability distribution, and Example 1 from Table 16 also conveys a neutral sentiment. Consequently, our learning algorithm has accurately predicted this as neutral, confirming the absence of hate-related content.

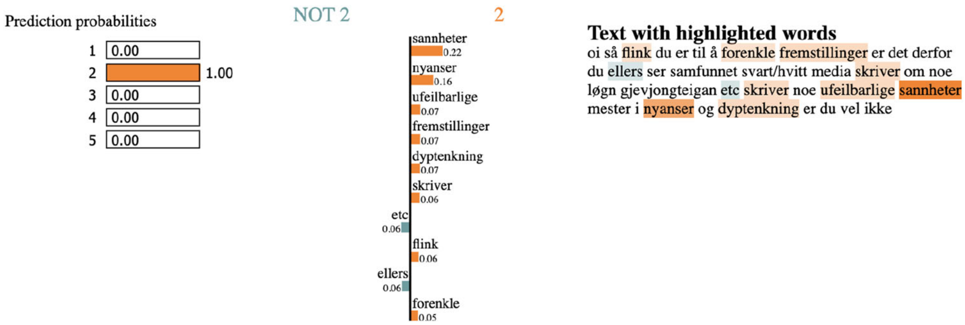
Figure 19 presents a more complex set of terms where "forbanna" (angry) could typically connote a negative sentiment. However, in the broader context of the discussion about cultural values, this expression of emotion does not translate into offensive or aggressive speech. The model's interpretation of these terms, while acknowledging the presence of strong emotion, appropriately recognizes the absence of targeted negativity or incitement, thus validating the neutral categorization.

After analyzing figure 20, the LIME visualization indicates an incorrect neutral classification by the model. The actual sentiment of the text implies a hateful intent, especially with the use of "send" in a context suggesting exile or banishment. This wrong prediction made by the learning algorithm shows the need to improve its ability to recognize context elements. For the second misclassified example 21, the model again incorrectly classifies the text as neutral, with a 100% probability. The text includes a term that refers to conflict with "Islam," and when combined with "ytringsfriheten" (freedom of speech) being the "det frste ofrest" (the first victim), it implies a negative sentiment towards the religion that could be perceived as 'moderately hateful.' This suggests animosity without an explicit call to action, which should have been flagged as such, rather than neutral. This misclassification highlights a potential area for improvement in the algorithm's ability to detect and accurately categorize subtle forms of HS.

Conclusion and future work

This research advances the field of multi-class HS detection by introducing an effective model for the Norwegian language, employing a BiLSTM-GRU architecture, known as FAST-RNN. Through rigorous regularization and hyperparameter tuning, the FAST-RNN model has demonstrated superior performance over the baseline across all evaluation metrics. The application of supervised FastText embedding has proven especially beneficial for categorical classification tasks. Additionally, this work has explored the capabilities of language-specific and multilingual transformer-based models enhanced by generative configuration and hyperparameter tuning. Moreover, prompt-based fine-tuning, including both few-shot and full fine-tuning, revealed that the latter substantially improved model outcomes due to the ability to provide more examples and the selection of an optimal generative

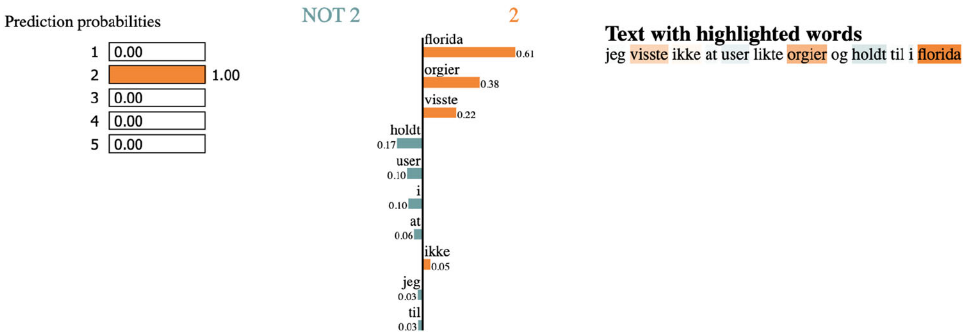
Fig. 16 Example 1: provocative instance visualization with LIME



Text with highlighted words

oi så flink du er til å forenkle fremstillinger er det derfor du ellers ser samfunnet svart/hvitt media skriver om noe løgn gjevjongteigan etc skriver noe ufeilbarlige sannheter mester i nyanser og dyptenkning er du vel ikke

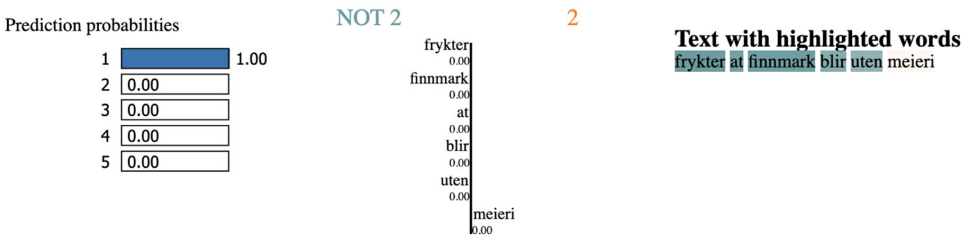
Fig. 17 Example 2: provocative instance visualization with LIME



Text with highlighted words

jeg visste ikke at user likte orgier og holdt til i florida

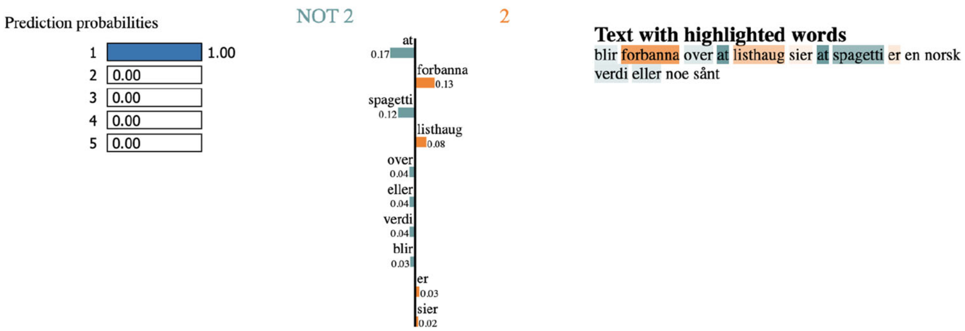
Fig. 18 Example 1: neutral instance visualization with LIME



Text with highlighted words

frykter at finnmark blir uten meieri

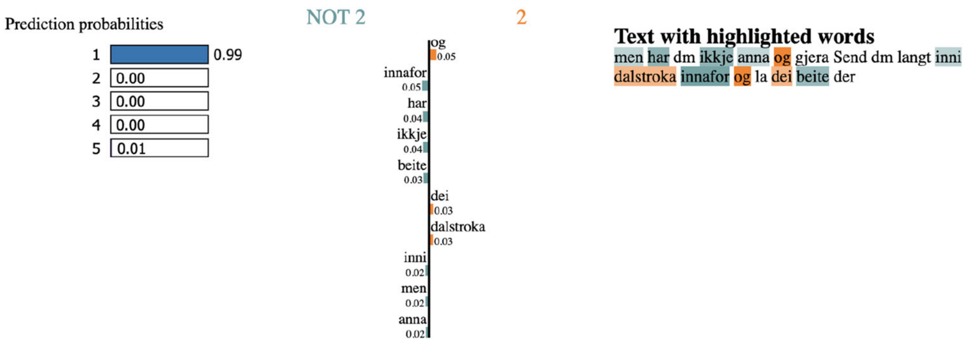
Fig. 19 Example 2: neutral instance visualization with LIME



Text with highlighted words

blir forbanna over at listhaug sier at spagetti er en norsk verdi eller noe sånt

Fig. 20 Example 1: wrong prediction visualization with LIME



Text with highlighted words

men har dm ikkje anna og gjera Send dm langt inni dalstroka innafor og la dei beite der

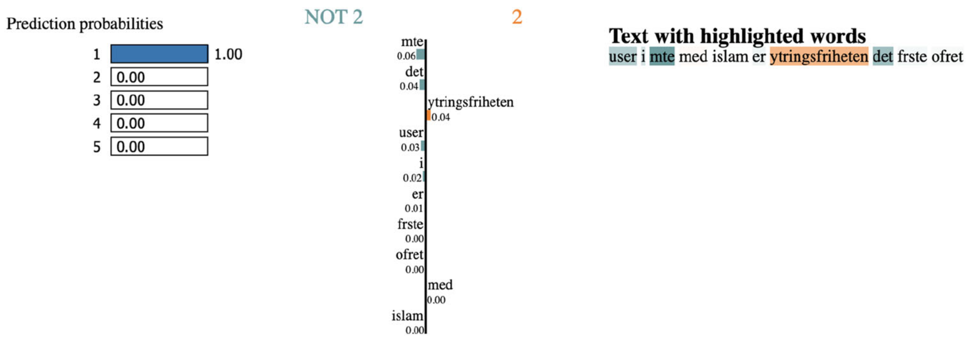


Fig. 21 Example 2: wrong prediction visualization with LIME

Table 16 Multi-class Norwegian HS examples for LIME

Category	Original Text	English Translation
Neutral	frykter at finnmark blir uten meieri	fear that finnmark will be left without a dairy
Neutral	blir forbanna over at listhaug sier at spagetti er en norsk verdi eller noe sånt	getting pissed off that listhaug says that spaghetti is a norwegian value or something like that
Provocative	oi så flink du er til å forenkle fremstillinger er det derfor du ellers ser samfunnet svart/hvitt media skriver om noe løgn gjevjongteigan etc skriver noe ufeilbarlige sannheter mester i nyanser og dyptenkning er du vel ikke	oh you are so good at simplifying representations that is why you otherwise see society black/white the media writes about something a lie gjevjongteigan etc writes something infallible truths you are probably not a master of nuances and deep thinking
Provocative	jeg visste ikke at user likte orgier og holdt til i florida	i did not know user liked orgies and lived in florida
Offensive	ja for bergens feita hoyre sosialisten solberg gjorde ikke noe for å få ham slengt ut er hun mascokist han erklarte jo på nrk at han skulle drepe henne noe galt er det med feita	yes, because bergens fat rightwing socialist solberg did not do anything to get him thrown out she is a masochist he explained on the phone that he was going to kill her is there something wrong with fat
Offensive	søt tenåring elsker å ri på en sto	cute teen loves to ride a big dick
Moderately Hateful	Ja de flytter hit og vi får verre liv klankultur misogyni høye økte offentlige utgifter press på skole og helsevesen gjengproblematikk resrelatert vold og aller verst en retardert avskyelig religion bruker vi din logikk bør vel alle pakistanere få komme inn og bli	yes they move here and we get a worse life clan culture misogyny high increased public expenditure pressure on schools and health care gang problems race related violence and worst of all a retarded disgusting religion if we use your logic i guess all pakistanis should be allowed to come in and stay
Moderately Hateful	innen 2050 er vi nok bde islamisert og kolonisert	by 2050 we will probably be both Islamized and colonized
Hateful	når du trenes opp i bomber kutter strupen på barn og kvinner da er du et hjernevasket sharia dyr og du er etnisk pakistansk bør du bli sendt til pakisan med ett lite bidrag de klarer av slikt send hennes influensere dit også	when you are trained in bombs cut the throats of children and women then you are a brainwashed sharia animal and you are ethnically pakistani you should be sent to pakisan with a small contribution they can manage from this send her influencers there too
Hateful	sende somaliere ut av dette landet	send somalis out of this country

configuration. The implementation of LIME for explainable AI has further strengthened our approach, providing clarity and understanding of the model's decision-making process. Transformer-based models did not exhibit the expected level of performance enhancement. This can be attributed to their reliance on large and complex datasets, which are often not available for less-resourced languages like Norwegian. Additionally, we observed that generally models with fewer parameters did not yield optimal results. In the future, we are determined to leverage advanced multilingual transformers such as mT5 and GPT models having a high number of parameters to cover more contextual information for multi-label and multi-class classification tasks in multilingual HS-related contexts, particularly for low-resource languages. Our approach will be to strategically navigate issues such as data sparsity and model adaptability to different languages, with a commitment to enhancing the performance of HS detection systems for various other low-resourced languages.

Acknowledgements This research work has been acknowledged by SOCYTI (https://www.bigdata.vestforsk.no/ongoing/socyti). The SOCYTI project has received funding from the Research Council of Norway as a Researcher Project for Technological Convergence related to Enabling Technologies under Grant Agreement No. 331736.

Author Contributions Ehtesham Hashmi: conceptualization, data analysis, formal analysis, research execution, design of methods, resources, software, writing original draft, and investigation. Sule Yildirim Yayilgan: visualization, supervision, project management, funding acquisition, research conduct, and validation.

Funding Open access funding provided by Norwegian University of Science and Technology.

Data availability and access Not Applicable.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Ethical and informed consent for data used Not applicable.

References

1. Akuma S, Lubem T, Adom IT (2022) Comparing bag of words and tf-idf with different models for hate speech detection from live tweets. Int J Inform Technol 14(7):3629–3635
2. Ali R, Farooq U, Arshad U et al (2022) Hate speech detection on twitter using transfer learning. Comput Speech Lang 74:101365
3. Andreassen SM, Seim GT (2020) Detecting and grading hateful messages in the norwegian language. Master's thesis, NTNU
4. Aswad E (2016) The role of us technology companies as enforcers of Europe's new internet hate speech ban. HRLR Online 1:1
5. Awal MR, Lee RKW, Tanwar E, et al (2023) Model-agnostic meta-learning for multilingual hate speech detection. IEEE Trans Comput Soc Syst
6. Ayo FE, Folorunso O, Ibharalu FT et al (2021) A probabilistic clustering model for hate speech classification in twitter. Expert Syst Appl 173:114762
7. Batarfi HA, Alsaedi OA, Wali AM, et al (2023) Impact of data augmentation on hate speech detection. In: International Conference on Innovations for Community Services, Springer, pp 187–199
8. Biecek P, Burzykowski T (2021) Local interpretable model-agnostic explanations (lime). Explanat Model Anal Explore Explain Examine Predict Models 1:107–124
9. Bigoulaeva I, Hangya V, Gurevych I, et al (2023) Label modification and bootstrapping for zero-shot cross-lingual hate speech detection. Lang Resour Evaluat:1–32
10. Bosco C, Felice D, Poletto F, et al (2018) Overview of the evalita 2018 hate speech detection task. In: Ceur workshop proceedings, CEUR, pp 1–9
11. Bromell D (2022) Regulating free speech in a digital age: hate, harm and the limits of censorship. Springer Nature, Berlin
12. Chhabra A, Vishwakarma DK (2023) A literature survey on multimodal and multilingual automatic hate speech identification. Multimed Syst:1–28
13. Chung HW, Hou L, Longpre S, et al (2022) Scaling instruction-finetuned language models. arXiv preprint arXiv:2210.11416
14. Clark K, Luong MT, Le QV, et al (2020) Electra: pre-training text encoders as discriminators rather than generators. arXiv:2003.10555
15. Costa VG, Pedreira CE (2023) Recent advances in decision trees: an updated survey. Artif Intell Rev 56(5):4765–4800
16. Davidson T, Warmsley D, Macy M, et al (2017) Automated hate speech detection and the problem of offensive language. In: Proceedings of the international AAAI conference on web and social media, pp 512–515
17. Devlin J, Chang MW, Lee K, et al (2019) Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv:1810.04805
18. Elzayady H, Mohamed MS, Badran KM et al (2023) A hybrid approach based on personality traits for hate speech detection in Arabic social media. Inte J Elect Comput Eng 13(2):1979
19. Fan L, Yu H, Yin Z (2020) Stigmatization in social media: documenting and analyzing hate speech for Covid-19 on twitter. Proc Assoc Inform Sci Technol 57(1):e313
20. Fersini E, Nozza D, Rosso P, et al (2018) Overview of the evalita 2018 task on automatic misogyny identification (ami). In: CEUR workshop proceedings, CEUR-WS, pp 1–9
21. Founta A, Djouvas C, Chatzakou D, et al (2018) Large scale crowdsourcing and characterization of twitter abusive behavior. In: Proceedings of the international AAAI conference on web and social media
22. Founta AM, Chatzakou D, Kourtellis N, et al (2019) A unified deep learning architecture for abuse detection. In: Proceedings of the 10th ACM conference on web science, pp 105–114
23. Gagliardone I, Gal D, Alves T, et al (2015) Countering online hate speech. Unesco Publishing
24. García-Díaz JA, Cánovas-García M, Colomo-Palacios R et al (2021) Detecting misogyny in Spanish tweets. An approach based on linguistics features and word embeddings. Fut Gen Comput Syst 114:506–518
25. Ghosh K, Senapati A, Narzary M et al (2023) Hate speech detection in low-resource bodo and assamese texts with ml-dl and bert models. Scalab Comput Pract Exp 24(4):941–955
26. Godioli A, Little LE (2022) Different systems, similar challenges: humor and free speech in the united states and Europe. Humor 35(3):305–327
27. Gomez Martin V (2023) Harm, offense, and hate speech. In: Crisis of the Criminal Law in the Democratic Constitutional State: Manifestations and Trends. Springer, p 119–135
28. Griffin R, Vander Maelen C (2023) Codes of conduct in the digital services act: exploring the opportunities and challenges. Available at SSRN
29. Holtzman A, Buys J, Du L, et al (2020) The curious case of neural text degeneration. arXiv:1904.09751
30. Jahan MS, Oussalah M (2023) A systematic review of hate speech automatic detection using natural language processing. Neurocomputing:126232
31. Khan L, Amjad A, Afaq KM et al (2022) Deep sentiment analysis using cnn-lstm architecture of english and roman urdu text shared in social media. Appl Sci 12(5):2694
32. Khanday AMUD, Rabani ST, Khan QR et al (2022) Detecting twitter hate speech in covid-19 era using machine learning and ensemble learning techniques. Int J Inform Manag Data Insights 2(2):100120
33. Kim JY, Kesari A (2021) Misinformation and hate speech: the case of anti-Asian hate speech during the covid-19 pandemic. J Online Trust Saf 1(1)
34. Kindermann D (2023) Against 'hate speech'. J Appl Philos
35. Kumar S, Marklund H, Van Roy B (2023) Maintaining plasticity via regenerative regularization. arXiv preprint arXiv:2308.11958
36. Kummervold PE, De la Rosa J, Wetjen F, et al (2021) Operationalizing a national digital library: the case for a norwegian transformer model. arXiv preprint arXiv:2104.09617

37. Kutuzov A, Barnes J, Velldal E, et al (2021) Large-scale contextualised language modelling for norwegian. arXiv preprint arXiv:2104.06546
38. Ma R, Miao J, Niu L et al (2019) Transformed 1 regularization for learning sparse deep neural networks. Neural Netw 119:286–298
39. Mandl T, Modha S, Kumar M A, et al (2020) Overview of the hasoc track at fire 2020: Hate speech and offensive language identification in tamil, malayalam, hindi, english and german. In: Proceedings of the 12th Annual Meeting of the Forum for Information Retrieval Evaluation, pp 29–32
40. Mansoor HM (2023) Diversity and pluralism in arab media education curricula: an analytical study in light of unesco standards. Hum Soc Sci Commun 10(1):1–11
41. Mazari AC, Boudoukhani N, Djeffal A (2023) Bert-based ensemble learning for multi-aspect hate speech detection. Cluster Comput:1–15
42. Mehta H, Passi K (2022) Social media hate speech detection using explainable artificial intelligence (xai). Algorithms 15(8):291
43. Meske C, Bunde E (2023) Design principles for user interfaces in ai-based decision support systems: the case of explainable hate speech detection. Inform Syst Front 25(2):743–773
44. Mikolov T, Grave E, Bojanowski P, et al (2017) Advances in pre-training distributed word representations. arXiv preprint arXiv:1712.09405
45. Mittal D, Singh H (2023) Enhancing hate speech detection through explainable ai. In: 2023 3rd International Conference on Smart Data Intelligence (ICSMDI), IEEE, pp 118–123
46. Nagar S, Barbhuiya FA, Dey K (2023) Towards more robust hate speech detection: using social context and user data. Soc Netw Anal Min 13(1):47
47. Nemade S, Mane SB, Nandgaonkar S (2023) Detection and classification of aggressive comments and hate speech. In: 2023 International Conference on Computational Intelligence and Sustainable Engineering Solutions (CISES), IEEE, pp 55–60
48. Nobata C, Tetreault J, Thomas A, et al (2016) Abusive language detection in online user content. In: Proceedings of the 25th international conference on world wide web, pp 145–153
49. i Orts ÒG (2019) Multilingual detection of hate speech against immigrants and women in twitter at semeval-2019 task 5: Frequency analysis interpolation for hate in speech detection. In: Proceedings of the 13th International Workshop on Semantic Evaluation, pp 460–463
50. Papcunová J, Martončik M, Fedáková D et al (2023) Hate speech operationalization: a preliminary examination of hate speech indicators and their structure. Complex Intell Syst 9(3):2827–2842
51. Parker S, Ruths D (2023) Is hate speech detection the solution the world wants? Proc Natl Acad Sci 120(10):e2209384120
52. Peng H (2020) A comprehensive overview and survey of recent advances in meta-learning. arXiv preprint arXiv:2004.11149
53. Pereira-Kohatsu JC, Quijano-Sánchez L, Liberatore F et al (2019) Detecting and monitoring hate speech in twitter. Sensors 19(21):4654
54. Platt M, Platt D (2023) Effectiveness of generative artificial intelligence for scientific content analysis. In: 17th International Conference on Application of Information and Communication Technologies, IEEE
55. Ptaszynski M, Pieciukiewicz A, Dybała P (2019) Results of the poleval 2019 shared task 6: First dataset and open shared task for automatic cyberbullying detection in polish twitter
56. Qiao C, Huang B, Niu G, et al (2018) A new method of region embedding for text classification. In: ICLR (Poster)
57. Raffel C, Shazeer N, Roberts A et al (2020) Exploring the limits of transfer learning with a unified text-to-text transformer. J Mach Learn Res 21(1):5485–5551
58. Risch J (2023) Toxicity. 86272(12):219–230
59. Rizwan H, Shakeel MH, Karim A (2020) Hate-speech and offensive language detection in roman urdu. In: Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP), pp 2512–2522
60. Sabiri B, El Asri B, Rhanoui M (2022) Mechanism of overfitting avoidance techniques for training deep neural networks. In: ICEIS (1), pp 418–427
61. Saleh H, Alhothali A, Moria K (2023) Detection of hate speech using bert and hate speech word embedding with deep model. Appl Artif Intell 37(1):2166719
62. Samuel D, Kutuzov A, Touileb S, et al (2023) Norbench–a benchmark for norwegian language models. arXiv preprint arXiv:2305.03880
63. Sanguinetti M, Poletto F, Bosco C, et al (2018) An italian twitter corpus of hate speech against immigrants. In: Proceedings of the eleventh international conference on language resources and evaluation (LREC 2018)
64. Satpute RS, Agrawal A (2023) A critical study of pragmatic ambiguity detection in natural language requirements. Int J Intell Syst Appl Eng 11(3s):249–259
65. Schmidt A, Wiegand M (2017) A survey on hate speech detection using natural language processing. In: Proceedings of the fifth international workshop on natural language processing for social media, pp 1–10
66. Sharma S, Agrawal S, Shrivastava M (2018) Degree based classification of harmful speech using twitter data. arXiv preprint arXiv:1806.04197
67. Trajano D, Bordini RH, Vieira R (2023) Olid-br: offensive language identification dataset for brazilian portuguese. Lang Resour Evaluat:1–27
68. Umer M, Imtiaz Z, Ahmad M et al (2023) Impact of convolutional neural network and fasttext embedding on text classification. Multimed Tools Appl 82(4):5569–5585
69. del Valle-Cano G, Quijano-Sánchez L, Liberatore F et al (2023) Socialhaterbert: a dichotomous approach for automatically detecting hate speech on twitter through textual analysis and user profiles. Expert Syste Appl 216:119446
70. Vaswani A, Shazeer N, Parmar N, et al (2023) Attention is all you need. arXiv:1706.03762
71. Vismara M, Girone N, Conti D et al (2022) The current status of cyberbullying research: a short review of the literature. Curr Opin Behav Sci 46:101152
72. Vučković J, Lučić S (2023) Hate speech and social media. TEME:191–207
73. Waseem Z (2016) Are you a racist or am i seeing things? Annotator influence on hate speech detection on twitter. In: Proceedings of the first workshop on NLP and computational social science, pp 138–142
74. Waseem Z, Hovy D (2016) Hateful symbols or hateful people? Predictive features for hate speech detection on twitter. In: Proceedings of the NAACL student research workshop, pp 88–93
75. William P, Gade R, esh Chaudhari R, et al (2022) Machine learning based automatic hate speech recognition system. In: 2022 International conference on sustainable computing and data communication systems (ICSCDS), IEEE, pp 315–318
76. Yildirim MM, Nagler J, Bonneau R et al (2023) Short of suspension: how suspension warnings can reduce hate speech on twitter. Perspect Polit 21(2):651–663

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.