International Journal of Data Science and Analytics (2025) 20:3053–3068
https://doi.org/10.1007/s41060-024-00650-6

REVIEW

Deep learning for hate speech detection: a comparative study

Jitendra Singh Malik[1] · Hezhe Qiao[2] · Guansong Pang[2] · Anton van den Hengel[1]

Received: 13 July 2024 / Accepted: 17 September 2024 / Published online: 22 October 2024

Abstract
Automated hate speech detection is an important tool in combating the spread of hate speech, particularly in social media. Numerous methods have been developed for the task, including a recent proliferation of deep-learning based approaches. A variety of datasets have also been developed, exemplifying various manifestations of the hate-speech detection problem. We present here a large-scale empirical comparison of deep and shallow hate-speech detection methods, mediated through the three most commonly used datasets. Our goal is to illuminate progress in the area, and identify strengths and weaknesses in the current state-of-the-art. We particularly focus our analysis on measures of practical performance, including detection effectiveness, computational efficiency, capability in using pre-trained models, and domain generalization. In doing so we aim to provide guidance as to the use of hate-speech detection in practice, quantify the state-of-the-art, and identify future research directions.

Keywords Hate speech detection · Natural language processing · Deep learning · Machine learning · Transformers

1 Introduction

Social media has experienced incredible growth over the last decade, both in its scale and importance as a form of communication. The nature of social media means that anyone can post anything they desire, putting forward any position, whether it is enlightening, repugnant or anywhere between. Depending on the forum, such posts can be visible to many millions of people. Different forums have different definitions of inappropriate content and different processes for identifying it, but the scale of the medium means that automated methods are an important part of this task. Hate-speech is an important aspect of this inappropriate content.

Hate-speech is a subjective and complex term with no single definition, however. Irrespective of the definition of the term or the problem, it is clear that automated methods for detecting hate-speech are necessary in some circumstances. In such cases it is critical that the methods employed are accurate, effective, and efficient.

A variety of methods have been explored for the hate speech detection task, including traditional classifiers [6, 17, 34, 45, 60], deep learning-based classifiers [1, 4, 7, 8, 38], or the combination of both approaches [7, 26, 40]. Classifiers like support vector machines (SVM), extreme gradient boosting (XGB), and multi-layer perceptrons (MLP) are commonly used in this task, which typically require vector representations of the text data. Bag of words models are commonly used, together with TF-IDF (term frequency - inverse document frequency) [2, 55]. With the progress in deep learning-based embeddings, tools such as word2vec [37], Glove [47], FastText [9, 28], and transformer-based methods [15, 19] have been applied to obtain more expressive representations. Both traditional and deep classifiers can be applied to these embedding-based representations pre-trained using the representation learning tools. This substantially increases the pool of methods for hate speech detection, resulting in a large set of possible hate speech

Jitendra Singh Malik and Hezhe Qiao have contributed equally to this work.

✉ Guansong Pang
gspang@smu.edu.sg

Jitendra Singh Malik
jmjmalik22@gmail.com

Hezhe Qiao
hezheqiao.2022@phdcs.smu.edu.sg

Anton van den Hengel
anton.vandenhengel@adelaide.edu.au

[1] School of Computer Science, University of Adelaide, Adelaide, South Australia

[2] School of Computing and Information Systems, Singapore Management University, Singapore, Singapore

detection solutions with different applicability in diverse real-world application contexts.

On the other hand, there have been a number of dataset benchmarks introduced and released for the evaluation of the performance of these methods, such as Davidson [17], Founta [21] and Twitter Sentiment Analysis (TSA).[1] These datasets differ largely from each other in terms of the classes of hatred texts (e.g., sexists, racists, abusive, and offensive tweets), data collection and labeling methods, and data distribution, representing the challenges and application demands from different perspectives.

To provide insightful application guidelines, in this paper we aim to provide a thorough empirical evaluation and comparison of different types of hate speech detection methods on these datasets. Through this evaluation study, we answer the following four key questions in hate speech detection. (i) How is the effectiveness of different popular detection models on diverse hate speech datasets? This is important because practitioners who have different application contexts often need to choose from the large pool of detectors. (ii) Are there any specific models that achieve generally more desired performance than the other models (in terms of both detection effectiveness and efficiency)? Both effectiveness and efficiency are crucial since handling those massive online text data in a timely manner requires computationally scalable and accurate detectors. (iii) How effective do popular pre-training methods work with detection models? Pre-training methods have been playing a major role in driving the development of many machine learning and natural language processing areas [4, 11, 56], including hate speech detection. It is thus important to evaluate the effectiveness of combining different pre-training and hate speech detectors. (iv) How is the generalizability of detection models in tackling domain shifts? This question is included because, due to the diversity in the classes of hate speech, ways of expressing hatred texts, and difference across languages, cross-domain hate speech detection has been emerging as one of the most important problems [20, 53]. To our best knowledge, there is no such comprehensive empirical evaluation. The most related work are [7, 16]. Badjatiya et al. [7] presents an empirical comparison of multiple classifiers (including both traditional and deep classifiers) for detecting hatred tweets, but it focuses on the detection effectiveness on on a single benchmark dataset of 16K tweets. Corazza et al. [16] exclusively focuses on empirical evaluation of identifying hate speech across different languages. Our work significantly complements these two studies in both of depth and breadth of the empirical evaluation.

To summarize, this work makes the following two major contributions.

- This paper presents a large-scale empirical evaluation of hate speech detection methods to provide insights into their detection effectiveness, computational efficiency, capability in using pre-trained models, and domain generalizability, offering important guidelines for their deployment in real-world applications. As far as we know, this is the first work dedicated to performing such a comparative study to investigate these questions.
- We perform a comprehensive evaluation study that involves 14 shallow/deep classification-based hate speech detectors, which are empowered by different word representation methods ranging from TF-IDF, Glove-based word embeddings to advanced transformers. To have an in-depth analysis of the performance in diverse application contexts, these detectors are evaluated on three large and publicly available hate speech detection benchmarks that contain different types of hatred tweet classes from different data sources. All codes are made publicly available at GitHub.[2]

This paper is organized as follows. We present a brief review of hate speech approaches in Sect. 2, followed by our evaluation approach in Sect. 3 and a series of our empirical evaluation results in Sect. 4. Lastly, the work is concluded in Sect. 5.

2 Hate speech detection approaches

Many methods have been introduced for hate speech detection [20, 48, 50, 62]. In this work we categorize them into two major groups, including traditional (shallow) classification methods and deep learning methods. The deep learning methods can be further categorized into word embedding-based methods and transformer-based methods, as illustrated in Fig. 1. We have a brief review of these methods in this section and perform comprehensive empirical evaluation of them in the next section.

2.1 Shallow methods

By shallow detection methods, we refer to hate speech detectors that use traditional word representation methods to encode words and apply shallow classifiers to perform the detection. Various types of such feature representations, such as TF-IDF [2, 55] and n-grams [10, 13, 17, 44], have demonstrated good performance, when combined with traditional classification models [10, 25, 31, 36, 53]. Additionally,

[1] https://www.analyticsvidhya.com/datahack/contest/practice-problem-twitter-sentiment-analysis/?utm_source=av_blog\&utm_medium=practice_blog_text_classification\#DiscussTab.

[2] https://github.com/jmjmalik22/Hate-Speech-Detection.

clustering-based word representation methods [18, 44, 59, 61, 64] have also been a popular method that can present the texts in lower dimensions. To capture the underlying sentiments in the texts, sentiment lexicon and the embedded polarity's degree are found to be helpful when modeling the texts for hate speech detection [10, 10, 17, 18, 24, 44]. Other semantics, such as part of speech and other relevant linguistic features [10, 13, 17, 24, 64], as well as word dependency (e.g., 'we versus them') [13, 64] can also be important to have more accurate detection of offensive texts.

In terms of classification models, support vector machines (SVM) is one of the most popular methods used in hate speech detection [7, 13, 17, 25, 36, 59]. Other popular classifiers for this task include naive Bayes [13, 17, 31, 36], logistic regression [17, 31, 60], random forest [17], and gradient boosting decision tree models [52].

2.2 Deep learning methods

Deep learning methods refer to deep neural network-based hate speech detectors. The input data to these neural networks can be in any form of feature encoding, including traditional methods like TF-IDF and recently emerged word embedding or pre-training methods. The latter approach is generally more effective than the former approach, because it helps avoid traditional feature engineering or feature construction methods. It instead learns feature representations from the presented texts. Some popular deep neural network architectures include convolutional neural networks (CNN), long short term memory (LSTM) and bi-directional LSTM (Bi-LSTM) [7, 18, 23, 46]. In hate speech detection, CNN models learn compositional features of words or characters [7, 23, 46], where LSTM models are used for learning the words that have a long-range dependency of the characters [7, 18].

2.2.1 Word embeddings-based methods

Word embedding techniques leverage distributed representations of words to learn their vectorized representations to enable downstream text mining tasks [28, 32, 37, 47]. The resulting embeddings allow the words with similar meaning to have similar representations in a vector space. There have been many word embeddings methods introduced over the years, such as word2vec [37], Glove [47], and FastText [9, 28]. One key intuition behind these models is that the word-word co-occurrence probabilities have the potential for encoding some form of semantic meaning between the words. Readers are referred to [11] for detailed review of these embedding methods.

Word embedding models have been widely used to enable hate speech detection and other relevant tasks, such as sentiment analysis, in a large number of studies [3, 23, 27, 40, 42, 49, 54, 60, 63]. The learned word embeddings can be combined with traditional classifiers [7, 63], or deep neural network-based classifiers, such as recurrent neural networks (RNN) [3, 54], gated recurrent units (GRU) [40], LSTM [12, 42], and CNN [23]. The word embedding models help capture semantic and syntactic word relations for detecting hatred tweets, empowering impressive detection effectiveness on different datasets.

2.2.2 Transformer-based methods

The above models rely on the usage of LSTM, Bi-LSTM, and CNN, along with the combination of Glove or other classical word embedding technique, which obtain promising performance but they are often not as accurate as the modern transformer-based embedding techniques, such as Small BERT [57], BERT [19, 22], ELECTRA [15], and AlBERT [57]. Readers can refer to [56] for detailed introduction of the transformer models. The transformers can be effectively combined with CNN, LSTM, multi-layer perceptrons (MLP) [43], Bi-LSTM [5, 16, 35], or some metric learning like contrastive learning [30]. They enable remarkable performance across hate speech detection datasets in different languages, such as datasets in French, English and Arabic [35], in Italian, English, Korean and German [16, 33, 39], or in English, Hindi, and German [51].

3 Our approach for comparative study

We aim at answering the following four key questions in hate speech detection through our comparative study.

- Q1: How is the effectiveness of different popular detection models on different hate speech datasets?
- Q2: Are there any specific models that achieve generally more desired performance than the other models (in terms of both detection effectiveness and efficiency)?
- Q3: How effective do popular pre-training methods work with detection models?
- Q4: How is the generalizability of detection models in tackling domain shifts?

To this end, we perform large-scale empirical evaluation of a large set of shallow and deep methods on three publicly available popular dataset benchmarks.

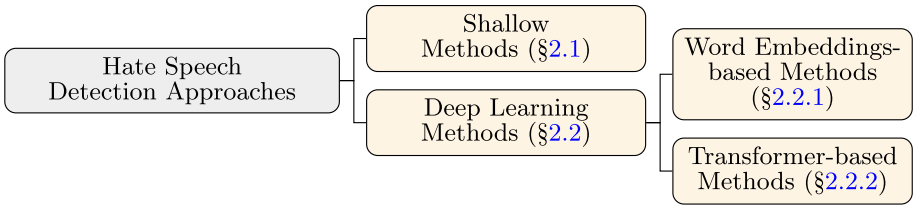

Fig. 1 Overview of the taxonomy of hate speech detection approaches

Table 1 Datasets and their key statistics

Dataset	Class and statistics
Davidson	Offensive—19,190 (77.4%)
	Hate—1,430 (5.8 %)
	Neither—4,163 (16.80%)
	Total ∼ 25*k*
Founta	None—53851(53.8 %)
	Hate—4,965 (4.96 %)
	Abusive—27,150 (27.15 %)
	Spam—14,030 (14.03%)
	Total ∼ 100k
TSA	Not Racist/Sexist—29,720 (92.99%)
	Racist/Sexist—2,242 (7.01 %)
	Total ∼ 32k

3.1 Datasets

Three widely-used datasets, including **Davidson** [17], **Founta** [21] and Twitter Sentiment Analysis (**TSA**),[3] are adopted in our experiments. Note that to address privacy concerns, the data only constitute of text sentences and the category they belong to. The details of the users are removed according to policy of different websites like Twitter.[4] An introduction to each of the datasets is presented below, with its key statistics summarized in Table 1.

Davidson [17] is collected starting with the hate speech lexicon that contains different words and phrases identified by various internet users as hate speech. By using the Twitter API, its authors then search for tweets containing terms from the lexicon, resulting in a set of 85.4 million tweets from 33,458 Twitter users. From this corpus, a random set of 25k tweets is sampled and then manually coded by CrowdFlower (CF) workers. Workers are asked to label each tweet as one of three categories *Offensive, Hate, Neither hate speech no offensive*.

Founta is recently published in [21], which contains around 100k human annotated tweets with about 27% being abusive, 14% being spam, and 5% being hate speech. The data was completed in different rounds. In the first round, annotators divided the tweets into three classes *normal, spam and inappropriate*. Thereafter, annotators were explained to further reclassify the tweets of the category *inappropriate*. The final version of the dataset includes four classes—*normal, spam, hate, and abusive.*

TSA is released by Analytic Vidhya[5] on the competition—Twitter Sentiment Analysis.[6] The dataset is composed by about 32K tweets, and contains only two classes namely 'Not Racist/Sexist' and 'Racist/Sexist'

3.2 Detection models

We consider three types of detection methods base on how they perform word embeddings, as the feature representation is the key to hate speech detection.

Traditional classifiers. This type of methods comprises of applying shallow classifiers—Support Vector Machine (SVM), extreme gradient boosting (XGB), and multi-layer perceptrons (MLP)—on top of the TF-IDF-based word representations. Particularly, the TF-IDF algorithm is a statistical method to measure the relevancy of a word in a document in a collection of the documents. The TF-IDF-based vector embeddings are then used with one of the traditional classifiers [7].

Deep models with glove embeddings. This approach uses the Glove-based embedding model with deep classifiers using CNN, MLP or Bi-LSTM-based network architecture. The classifier is trained with cross-entropy loss.

Deep models with transformer-based pretraining. This approach is focused upon the transformer models, including Small BERT [57], BERT [19], ELECTRA [15], and AlBERT [57], with each in combination with CNN and MLP separately. BERT and other Transformer encoder architectures have been successful on a variety of tasks in NLP (natural language processing). They compute vector-space representations of natural language that are suitable for use in deep learning models. The BERT family of models uses the Transformer encoder architecture to process each token of input

[3] https://www.analyticsvidhya.com/datahack/contest/practice-problem-twitter-sentiment-analysis/?utm_source=av_blog&utm_medium=practice_blog_text_classification#DiscussTab.

[4] https://developer.twitter.com/en/developer-terms/agreement-and-policy.

[5] https://www.datahack.analyticsvidhya.com/contest/practice-problem-twitter-sentiment-analysis/?utm_source=av_blog&utm_medium=practice_blog_text_classification#DiscussTab.

[6] https://www.kaggle.com/dv1453/twitter-sentiment-analysis-analytics-vidya.

text in the full context of all tokens before and after. The classifier is also trained with cross-entropy loss.

All these together result in 14 popular/state-of-the-art models in this study, including three TF-IDF-based traditional classifiers, three Glove-based models, and eight transformer-based models.

3.3 Implementation details

We present the implementation details from three consecutive steps, including embedding, detection model, and optimization.

Embedding. A sentence of length n can be represented as $w_1, w_2....w_n$ where each word can be represented as a real valued vector. For word embedding, we deploy TF-IDF, the pre-trained Glove embedding of Tweets, or transformer-based embedding. In TF-IDF, each tweet is represented by a vector with a dimensionality size of the dictionary (a collection of unique words across all training tweets) size, in which each entry is denoted by the multiplication of the frequency (TF) and inverse document frequency (IDF) of a specific term (word). This simple embedding method is used in three traditional classifiers: SVM, XGB and MLP. Note that due to extensive computation it takes, we limited the dictionary size to maximum 10k in our experiments.

Glove is generally a log-bilinear model with a weighted least-squares objective. The main intuition underlying the model is the simple observation that the common word-word co-occurrence patterns have the potential for encoding some form of semantic. The advantage of this model is that we can leverage massive datasets with billions of words that one may not have access to, to capture word meanings in a statistically robust manner. In our implementation, we use pre-trained Glove word vectors with a dimensionality size of 100. This Glove-based embedding layer is subsequently connected to CNN, MLP, or Bi-LSTM to train deep classifiers.

The BERT family of models have similar advantages as Glove, but can produce more meaningful word embeddings. In our experiment, we utilize the pre-trained BERT models available on TensorFlow Hub,[7] which allows us to easily integrate BERT in our implementation. Four different types of transformer-based embedding models are used, including

- BERT,[8] which is pre-trained on a large corpus of text, and then fine-tuned for specific tasks [19].
- Small BERT,[9] which is an instance of the original BERT architecture with a smaller number L of layers (i.e., residual blocks) combined with a smaller hidden size H and a matching smaller number A of attention heads [57]. Small BERTs have the same general architecture but fewer and/or smaller Transformer blocks, enabling a good trade-off between speed, size and quality.
- ALBERT,[10] which is "A Lite" version of BERT with greatly reduced number of parameters [57]. ALBERT computes dense vector representations for natural language by using a deep neural network with the transformer architecture.
- ELECTRA,[11] which is a BERT-like model that is pre-trained as a discriminator in a set-up resembling a generative adversarial network (GAN) [15].

Similar to [5, 5, 35, 58], the BER models are subsequently connected to a CNN/MLP architecture to train the deep detectors.

Network architecture. In this section we introduce the implementation of each model in our study. More specifically, we use SVM with linear kernel and XGB with default parameters as in the Scikit-learn library.[12]

For MLP, its architecture consists of one dense layer, a dropout layer with the probability of 0.1, and one classification layer. For the CNN architecture, as shown in Fig. 2, it consists of two CNN layer with filter size of 32 and 64, respectively. The output from the last CNN layers is fed to a maxpooling layer, followed by a dense layer with 256 neurons and a dropout layer with the probability of 0.1, before feeding to the last output layer.

In Bi-directional LSTM that is used in the Glove-based model, it is composed of Glove Embedding Layer (dimension=100), followed by Bi-LSTM with recurrent dropout of a probability equal to 0.2, a Global Max pooling layer of one dimension, a Batch Normalistaion layer. It is then followed by the combination of dropout layer with a probability of 0.5 and a dense layer with the relu activation, and lastly a softmax classification layer to predict the the classes. The complete architecture is shown in Fig. 3.

Optimization. SVM and XGB are trained with the recommended settings in Scikit-learn. The neural network-based classifiers are trained with a batch size of 32, 128 or 256 for 10–25 epochs. Adam optimizer is used with a learning rate of 2e−5. As input for transformer embedding, we tokenize each tweet with the BERT tokenizer. It contains invalid characters removal, punctuation splitting, and lowercasing the words. Based on the original BERT, we split words to sub word units using WordPiece tokenization. As tweets are short texts, we set the maximum sequence length to 64 and in any

[7] https://www.tensorflow.org/hub.

[8] bert_en_uncased_L-12_H-768_A-12/3 at https://tfhub.dev/google/collections/bert/1.

[9] small_bert/bert_en_uncased_L-2_H-12_A-2/1 at https://tfhub.dev/google/collections/bert/1.

[10] albert_en_base/2 at https://tfhub.dev/google/albert_base/3.

[11] google/electra_small/2 at https://tfhub.dev/google/collections/electra/1.

[12] https://scikit-learn.org/.

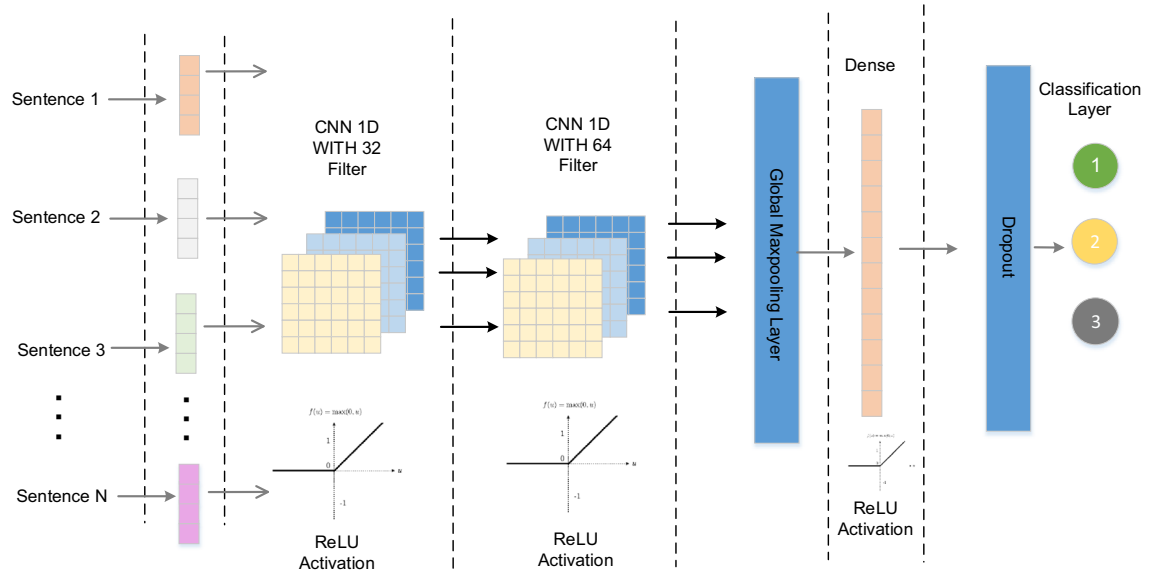

Fig. 2 CNN Architecture used in this study

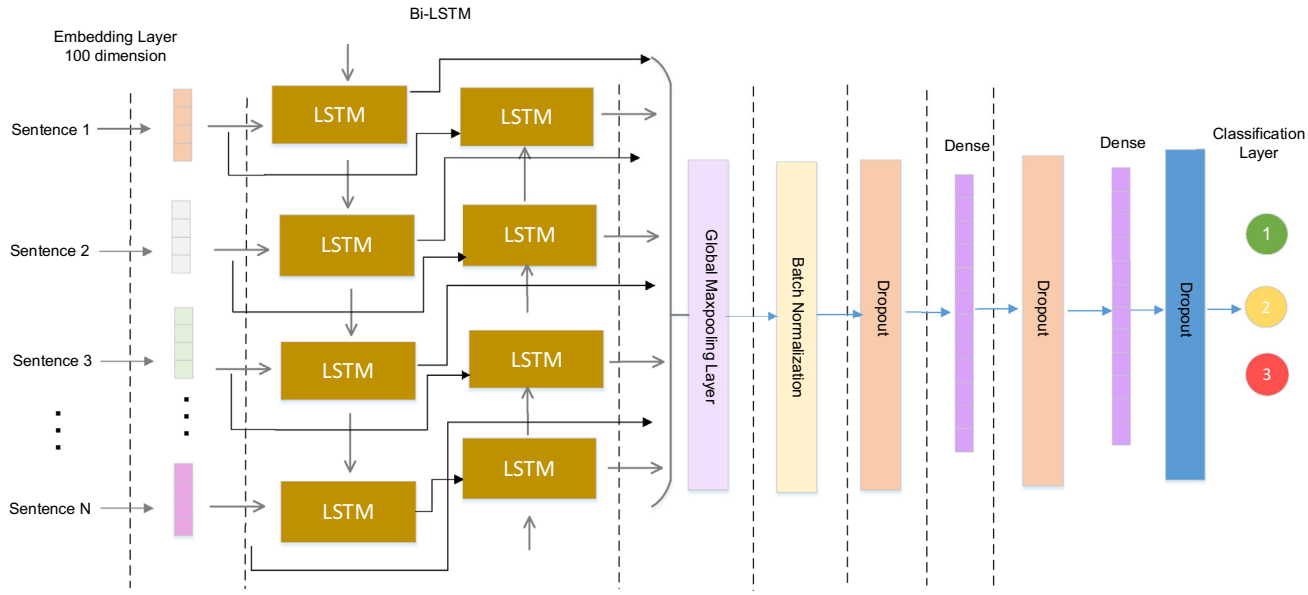

Fig. 3 Bi-LSTM Network Architecture used in this study

shorter or longer length case it is padded with zero values or truncated to the maximum length. We use the cross entropy loss function and the Adam optimizer to optimize the models.

All experiments are performed using a Google co-laboratory tool which is a free research tool with a Tesla K80 GPU and 12 G RAM and Kaggle environment.

3.4 Performance evaluation metrics

Per-class metric. The results of different methods are measured based on Precision, Recall and F_1 score [14] defined as below:

- Precision p_i - Precision is the fraction of true positive examples among the examples that the model classified as positive for a specific class.
- Recall r_i - Recall is the fraction of examples classified as positive among the total number of positive examples per class.
- F_1 score - F_1 score can be defined as the harmonic mean of precision and recall: $F_1 = 2\frac{p_i . r_i}{(p_i + r_i)}$.

Evaluation metric. For the overall performance, we consider both of macro F_1 score and weighted average F_1 score of the classification results, as the data is highly imbalanced in the three data sets in Table 1.

- Macro F_1 score calculates the mean of the F_1 scores across all classes: $Macro\ F_1 = \frac{1}{C}\sum_{i=1}^{i=C} F_{1i}$, where C is the number of classes and F_{1i} is the F_1 score of class i. It considers all classes equally.
- Weighted F_1 score, on the other hand, considers the individual weight of each class [29], and is defined as $Weighted\ F_1 = \frac{1}{N}\sum_{i=1}^{i=N} F_{1i}N_i$, where $N = \sum_{i=1}^{i=N} N_i$ and N_i is the sample size of class i.

4 Experimental results & analysis

4.1 Q1: Effectiveness of different models on popular benchmarks

The macro and weighted average results of all 14 hate speech detection models on three popular benchmarks are shown in Table 2. Below we discuss the results on each dataset.

Results on the Davidson dataset. In terms of macro F_1 score, it can be observed that Glove embedding is better than TF-IDF in predicting hate speech more accurately. The results obtained by Glove embedding with CNN and Bi-LSTM are similar to the highest score obtained by TF-IDF embeddings with XGB. This high score was almost touched twice by the glove embedding. The TF-IDF-based MLP and SVM models obtain similarly good performance as the Glove-based MLP model. In terms of weighted average F_1 score, it can be found that XGB with TF-IDF produced very competitive results, which is more effective than all Glove-based models.

Compared to the TF-IDF-based and Glove-based models, the transformer-based models achieve a significant increase in both metrics, achieving the best macro F_1 score of 0.76 and the best weighted average F_1 score of 0.91. More specifically, in terms of macro F_1, all the transformers models outperform every single model in combination with TF-IDF or Glove by a large margin. These results demonstrate that the transformers are able to perform better on both large and small classes, especially on the small classes (i.e., the hate class), when compared to Glove and TF-IDF. Small BERT is less effective than TF-IDF-based XGB in weighted average F_1 score, indicating that Small BERT underperforms XGB on the large class (i.e., the offensive class). Overall, BERT-based CNN and ELECTRA-based MLP turn out to be the best performers in this dataset.

Results on the Founta dataset. The Founta dataset is more diversified than the other two datasets. Therefore, the performance results within the models vary more compared to the other two datasets. Interestingly, the TF-IDF-based models are generally more effective than the Glove-based models. In terms of macro average, TF-IDF-based SVM obtains very good results with a Macro F_1 score of 0.64, outperforming all Glove-based models and performing comparably well to some transformer-based models. Nevertheless, transformer embeddings in this case enable the most superior results to other embeddings. BERT, ALBERT, and ELECTRA (with CNN) prove to be much more successful than other models.

Similar to the results on Davidson, the TF-IDF-based XGB outperforms the other combination with SVM and MLP in weighted average F_1, obtaining an F_1 score of 0.78 which also achieves the second highest weighted F_1 score among all the models. transformer-based models come with the best results. Particularly, the BERT and Al-BERT models dominate the other models with the weighted F_1 score reaching to 0.79 in both CNN and MLP, followed by ELECTRA at 0.78 with MLP.

Results on the TSA dataset. The results of most methods on the TSA dataset are much better than the result on the other two datasets, which may be due to the reason that fewer classes and more training samples are presented for each class (see Table 1). The results in terms of the macro F_1 show similar performance as in previous datasets, with transformers embeddings performing far better than the other two embeddings, achieving a macro F_1 score as high as 0.90. On this dataset, the TF-IDF-based XGB is not as successful as the TF-IDF-based MLP that achieves the macro F_1 score of 0.81. The TF-IDF-based MLP also outperforms all Glove-based models. In transformer embeddings, BERT (with CNN/MLP) and Al-BERT (with CNN) perform far better than other embeddings with the F_1 score of 0.90, followed by ELECTRA, with an F_1 score of 0.89 (CNN, MLP).

The weighted average score results are very different from the macro average. The TF-IDF-based MLP and XGB obtain the best results with the weighted F_1 score of 0.95 and 0.96, respectively. In terms of Glove embedding, the BI-LSTM model performs similarly well to the TF-IDF-based MLP, and outperforms the CNN/MLP model. The transformer-based models are more effective, with the results obtained by BERT, Al-BERT, and ELECTRA achieving an F_1 score of 0.97/0.98. Small BERT and ALBERT (with ML) are slightly less effective compared to other transformer-based models.

4.2 Q2: Superiority of specific models in both effectiveness and efficiency

Effectiveness. As shown in Table 2, the Glove and TF-IDF models do not have consistently superior effectiveness over each other. It is interesting that the TF-IDF-based XGB is able to perform better or on par, compared to the results obtained by Glove-based neural network models. Similar to our study,

Table 2 Macro and weighted average F_1 score performance

Embedding	Model	Macro			Weighted avg.		
		P	**R**	F_1	**P**	**R**	F_1
Results on the Davidson dataset							
TF-IDF	TF-IDF + SVM	0.69	0.64	0.66	0.86	0.87	0.86
	TF-IDF + XGB	0.74	0.69	0.70	0.89	*0.90*	*0.90*
	TF- IDF + MLP	0.68	0.63	0.66	0.85	0.86	0.85
Glove	Glove + CNN	0.66	0.73	0.69	0.88	0.85	0.86
	Glove + MLP	0.72	0.61	0.65	0.85	0.86	0.85
	Glove + Bi-LSTM	0.67	0.75	0.69	0.88	0.84	0.86
Transformers	Small BERT + CNN	0.72	**0.82**	*0.75*	**0.91**	0.85	0.87
	Small BERT + MLP	0.71	*0.81*	0.74	**0.91**	0.85	0.87
	BERT + CNN	**0.78**	0.75	**0.76**	**0.91**	**0.91**	**0.91**
	BERT + MLP	0.75	0.72	0.74	*0.90*	*0.90*	*0.90*
	Al-BERT + CNN	0.76	0.69	0.72	0.89	*0.90*	*0.90*
	Al-BERT + MLP	*0.77*	0.72	0.74	*0.90*	**0.91**	*0.90*
	ELECTRA + CNN	0.75	0.75	*0.75*	**0.91**	**0.91**	**0.91**
	ELECTRA + MLP	0.75	0.76	**0.76**	**0.91**	**0.91**	**0.91**
Results on the Founta dataset							
TF-IDF	TF-IDF + SVM	0.63	*0.71*	0.64	*0.80*	0.73	0.75
	TF- IDF + XGB	**0.74**	0.59	0.62	0.79	**0.80**	*0.78*
	TF- IDF + MLP	0.64	0.61	0.62	0.75	0.76	0.76
Glove	Glove + CNN	0.58	0.54	0.52	0.73	0.70	0.69
	Glove + MLP	0.59	0.61	0.59	0.74	0.72	0.73
	Glove + Bi-LSTM	0.63	0.65	0.63	0.77	0.75	0.76
Transformers	Small BERT + CNN	0.63	*0.71*	0.65	*0.80*	0.73	0.75
	Small BERT + MLP	0.64	*0.71*	0.66	0.79	0.74	0.76
	BERT + CNN	*0.68*	0.67	*0.67*	0.79	*0.79*	**0.79**
	BERT + MLP	*0.68*	0.67	**0.68**	0.79	*0.79*	**0.79**
	Al-BERT + CNN	*0.68*	0.68	**0.68**	0.79	*0.79*	**0.79**
	Al-BERT + MLP	*0.68*	0.67	*0.67*	0.79	*0.79*	**0.79**
	ELECTRA + CNN	0.65	**0.73**	0.66	*0.80*	0.73	0.75
	ELECTRA + MLP	0.66	**0.73**	**0.68**	**0.81**	0.76	*0.78*
Results on the TSA dataset							
TF-IDF	TF-IDF + SVM	0.85	0.72	0.77	0.91	0.92	0.91
	TF- IDF + XGB	0.70	**0.99**	0.76	**0.98**	0.95	0.96
	TF- IDF + MLP	0.84	0.78	0.81	0.95	0.95	0.95
Glove	Glove + CNN	0.74	0.81	0.77	0.94	0.93	0.94
	Glove + MLP	0.74	0.81	0.77	0.94	0.93	0.94
	Glove + Bi-LSTM	0.78	0.84	0.80	0.95	0.94	0.95
Transformers	Small BERT + CNN	0.84	0.83	0.84	0.96	0.96	0.96
	Small BERT + MLP	0.82	0.82	0.82	0.95	0.95	0.95
	BERT + CNN	**0.93**	0.88	**0.90**	**0.98**	**0.98**	**0.98**
	BERT + MLP	**0.93**	0.87	**0.90**	*0.97*	**0.98**	*0.97*
	Al-BERT + CNN	*0.91*	*0.90*	**0.90**	*0.97*	*0.97*	*0.97*
	Al-BERT + MLP	0.86	0.84	0.85	0.96	0.96	0.96
	ELECTRA + CNN	0.90	0.87	*0.89*	*0.97*	*0.97*	*0.97*
	ELECTRA + MLP	0.90	0.87	*0.89*	*0.97*	*0.97*	*0.97*

The best performance is highlighted in bold values, while the second best is in italic values

it was also observed in [3] that XGB was able to obtain better results compared to Glove-based deep models [1].

The use of Bidirectional LSTM is quite successful in our experiments and obtains good results across three datasets. This indicates that Bi-LSTM is successful in recognizing the underlying relationships in a set of data better than most of the other models except transformers.

In terms of transformers embedding, the use of BERT, ELECTRA, and Al-BERT provide the best results in all three datasets, showing quite consistently superior performance over the competing models. The higher performance of the transformer embeddings can also be found in other studies [41], in which the base version of BERT was used in combination with CNN and LSTM networks.

The main reason is that text context is crucial in hate speech detection because subtle cues (like sarcasm, negation, or emphasis) often determine whether a statement is hateful or not. The transformers use a self-attention mechanism, which allows them to focus on different parts of the input texts with varying importance. The self-attention mechanism enables the transformers to capture long-range dependencies and relationships between words more effectively, compared to LSTMs that tend to struggle with long-term dependencies. In addition, hate speech often involves subtle linguistic features like sarcasm, double meanings, or implicit bias. Transformer models are better suited to capture these complexities because of their capacity to model language at different levels of abstraction.

Computational efficiency. In order to measure the computational time of the models, we consider the running time to complete one epoch for every model. The results of computational time are shown in Fig. 4. It is observed that while the effectiveness provided by the transformers is higher than the other embeddings, it does require much more time to train on the dataset. It can be found that AI-BERT is the most time-consuming model among the transformers, followed by the BERT model in all there datasets we have tested. Note that Bi-LSTM in combination with Glove is significantly more costly than Glove-based CNN and MLP models, which also take a long time to train than ELECTRA and Small-BERT. Small BERT is the most efficient model among all the transformers due to the smaller number of trainable parameters than the other versions of BERT.

Overall, ELECTRA-based MLP models seem to be the most practical method that can normally achieve the best classification effectiveness while at the same time being sufficiently computation efficient.

4.3 Q3: Pre-training in deep hate speech detection models

This section discusses how large-scale pre-training helps the subsequent hate speech detection. There are two types of pre-training models used in our study: the Glove-based models and transformer-based models. Glove is pre-trained on Twitter with 2B tweets, 27B tokens, a vocabulary of size 1.2M, and 100 dimensions. Transformers are pre-trained on the Wikipedia and BooksCorpus. TF-IDF-based models can be treated as non-pre-trained models. As shown in Table 2, the pre-trained models (e.g., Glove-based MLP and transformer-based MLP) perform generally better than the plain models (e.g., TF-IDF-based MLP) on the three datasets. The transformer-based pre-training is typically much better than the TF-IDF-based MLP models.

Compared to Glove-based pre-training, different transformer-based pre-training methods perform consistently better across all three datasets in both macro F_1 and weighted average F_1 measures. This also applies to both CNN and MLP-based classifiers.

4.4 Q4. Cross-domain hate speech detection

In this section, we investigate the effectiveness of different models in generalizing from one domain to another domain to detect hate speech. The domain difference is mainly due to the source of the datasets. One problem here is that the three datasets have different sets of classes. To tackle this issue, we remove less relevant classes and focus on detecting hatred tweets. The resulting datasets are shown in Table 3.

We then perform three sets of cross-domain experiments, with each model trained on one of the three datasets and evaluated on the other two datasets, *i.e.*, one dataset is used as the source domain, while the other two datasets are used as the target domain. In addition, we also present the results of the test data of the source domain dataset to serve as baseline performance. The three cross-domain scenarios are illustrated in Fig. 5. In this experiment, we focus on models that obtain very good performance in Table 2 as less effective models in non-cross-domain settings are expected to perform poorly in cross-domain settings.

Results in Scenario 1. Experiment scenario 1 is to train models on Davidson and test models on the other two datasets. The results are presented in Table 4. Compared to the results on the test data of the source domain dataset Davidson, the cross-domain performance drops significantly on both Founta and TSA, especially on Founta where all Glove and transformers models obtain F_1 score at around 0.5–0.6 in both macro F_1 and weighted average F_1 scores. The performance on TSA gets better for most of the models, particularly on the weighted average results. This indicates the non-hatred tweets in TSA share some common features with that in Davidson, resulting in a good performance on the non-hatred tweet classification and thus a high weighted average F_1 score.

In the case when the testing is performed on the Davidson dataset, ELECTRA-based CNN models outperform the other

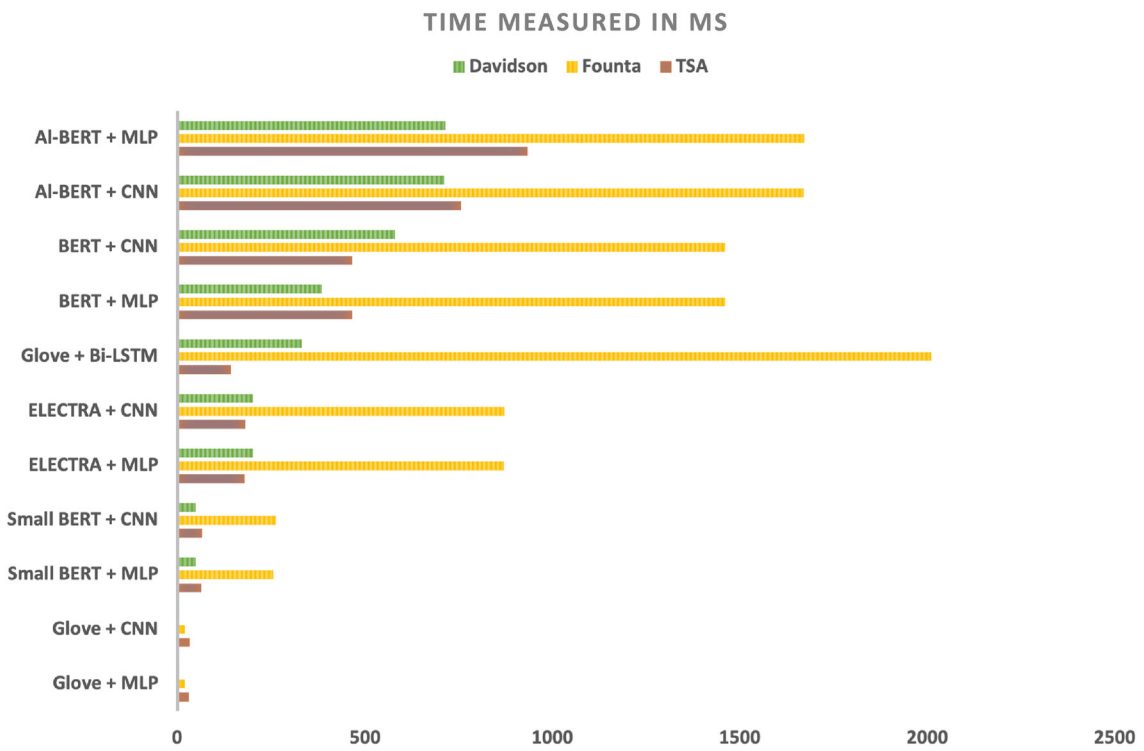

Fig. 4 Computational time of the models per epoch measured in ms

Table 3 Datasets for cross-domain experiments

Dataset	Details
Davidson	Hate—1430 (34.05 %)
	Neither—4163 (65.95%)
	Total ∼ 5.5k
Founta	None—52885 (62.3 %)
	Hate (Combination of hate - 4965 and Abusive - 27150 (37.7 %))
	Total ∼ 85k
TSA	Neither (not racist/sexist)—29720 (92.99%)
	Hate (racist/sexist)—2242 (7.01%)
	Total ∼ 32k

models and obtain the best performance with an F_1 score of 0.94, followed by that of other transformer embeddings. Glove models are good at detection but it is still not comparable with transformers embeddings. In terms of weighted average score, the model with the best performance is ELECTRA with CNN, achieving an F_1 score of 0.95, followed by other transformed embeddings. Glove-based CNN models provide a score notably equal to Small BERT-based models. Note that the results here are different from that in Table 2, as the datasets are reduced and typically become simpler datasets.

Results in Scenario 2. In the experiment scenario 2, we train models on the Founta dataset and evaluate them on the other two datasets, with the results on the Founta test set as the baseline. The results are presented in Table 5. The models generalize poorly from Founta to Davidson, which is similar to the results of generalizing from Davidson to Founta. However, it is interesting that when we test the models on Davidson dataset, transformer embeddings used with MLP-based models enable really good results, with macro average F_1 leading to 0.81 and weighted average F_1 leading to 0.85.

On the Founta Dataset, all models achieve excellent performance since there is no domain shift in this case. Here the transformer embeddings provide better results than Glove models which reach 0.93 in macro average and 0.93−0.94 in a weighted average for ELECTRA and Small BERT. Glove models also achieve good results with F_1 score of 0.90−0.91.

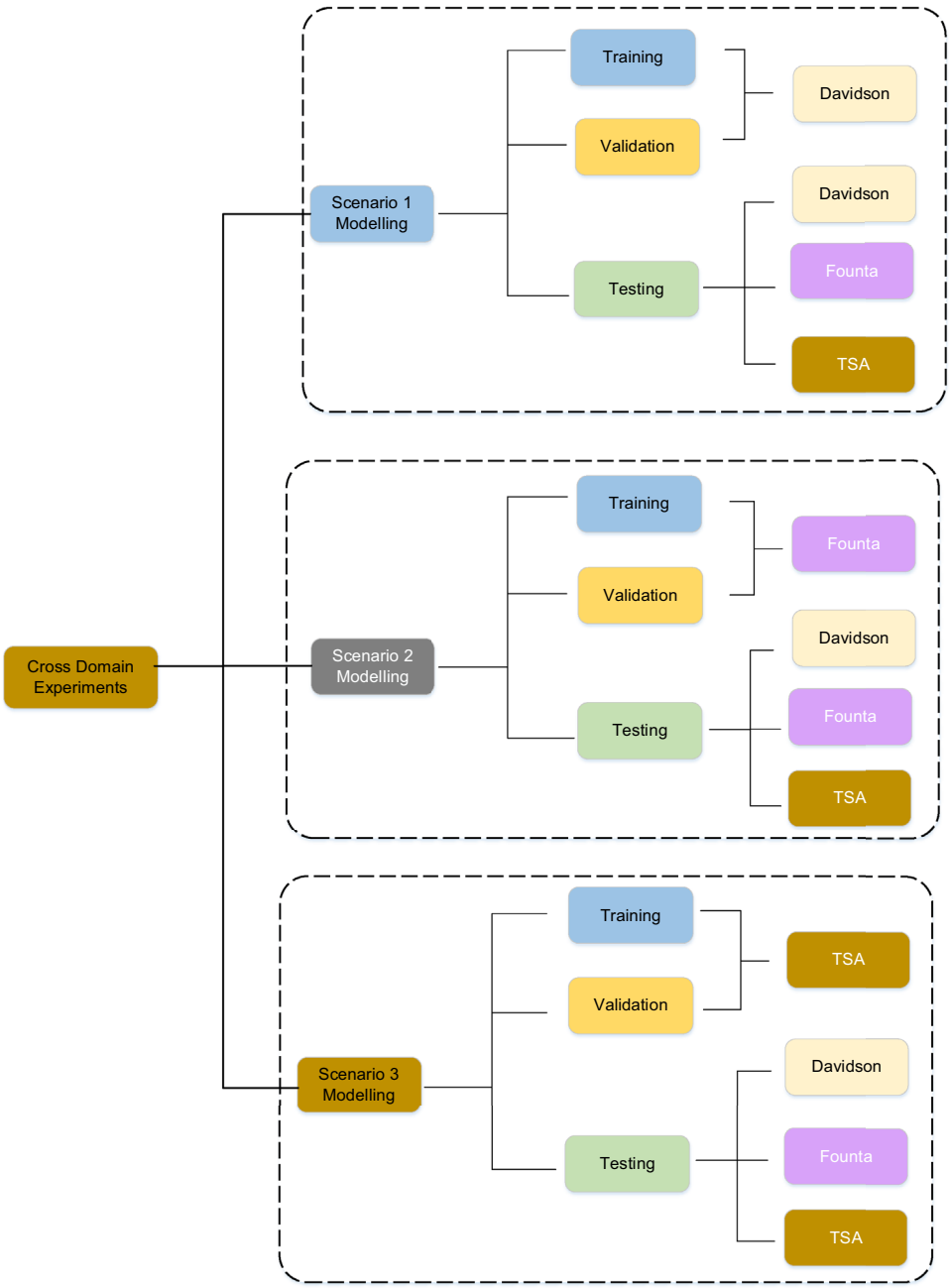

Fig. 5 Three scenarios of cross-domain experiments

On the experiments on the TSA dataset, transformer embeddings obtain well performance reaching to 0.49−0.50 on Precison for Small BERT and ELECTRA for macro average. For weighted average, the performance is noted by 0.86 for ELECTRA, Outperforming Glove models with 0.68 and 0.75.

Results in Scenario 3. Here the models are trained on the TSA dataset and evaluated on the other two datasets. The results are shown in Table 6.

When the testing is performed on the Davidson dataset, transformer embeddings are better than Glove models. Small BERT with CNN can obtain a macro F_1 score of 0.61, followed by its combination with MLP at 0.60. Glove models obtain a precision of 0.45/0.43. In terms of weighted average score, Small BERT with CNN is still the best performer, achieving a weighted average F_1 score of 0.73, followed by Small BERT with MLP that obtains a score of 0.72. ELECTRA is the second best with F_1 score ranging in 0.70 and 0.71. Glove lags behind with F_1 score at 0.61/0.62.

In case of the Founta data, the pattern in prediction is similar to that in Scenario 1. The best performance here is only 0.47 in macro F_1 score and 0.54 in weighted average F_1 score, both of which are achieved by Small BERT-based

Table 4 Scenario 1—Cross-domain experimental results with the Davidson dataset as the source domain dataset

Embedding	Model	Macro			Weighted avg.		
		P	**R**	F_1	**P**	**R**	F_1
Testing—Davidson							
Glove	Glove + CNN	0.90	0.90	0.90	0.92	0.92	0.92
	Glove + MLP	0.84	0.84	0.84	0.88	0.88	0.88
Transformers	Small BERT + CNN	0.89	0.90	0.90	0.92	0.92	0.92
	Small BERT + MLP	0.90	0.90	0.90	0.92	0.92	0.92
	ELECTRA + CNN	0.93	**0.93**	0.93	0.94	0.94	0.94
	ELECTRA + MLP	**0.95**	**0.93**	**0.94**	**0.96**	**0.96**	**0.95**
Testing—Founta							
Glove	Glove + CNN	**0.55**	**0.54**	**0.54**	**0.57**	**0.59**	**0.57**
	Glove + MLP	0.50	0.50	0.49	0.53	0.57	0.54
Transformers	Small BERT + CNN	0.51	0.51	0.50	0.54	0.55	0.54
	Small BERT + MLP	0.51	0.51	0.50	0.54	0.55	0.54
	ELECTRA + CNN	0.49	0.49	0.49	0.52	0.55	0.53
	ELECTRA + MLP	0.49	0.49	0.49	0.52	0.55	0.53
Testing—TSA							
Glove	Glove + CNN	0.49	0.46	0.46	0.87	0.73	0.79
	Glove + MLP	0.50	0.49	0.46	0.87	0.70	0.77
Transformers	Small BERT + CNN	0.55	**0.60**	0.56	**0.89**	0.83	0.85
	Small BERT + MLP	0.55	0.59	0.56	**0.89**	0.85	0.87
	ELECTRA + CNN	0.61	0.56	**0.58**	**0.89**	0.91	**0.90**
	ELECTRA + MLP	**0.63**	0.56	**0.58**	**0.89**	**0.92**	**0.90**

For each dataset, the best performance per column within each metric is boldfaced

Table 5 Scenario 2—Cross-domain experimental results with Founta as the source domain

Embedding	Model	Macro			Weighted avg.		
		P	**R**	F_1	**P**	**R**	F_1
Testing—Davidson							
Glove	Glove + CNN	0.55	0.56	0.55	0.66	0.60	0.62
	Glove + MLP	0.49	0.49	0.48	0.60	0.58	0.59
Transformers	Small BERT + CNN	0.76	0.80	0.77	0.83	0.81	0.82
	Small BERT + MLP	0.79	0.83	**0.81**	0.86	0.84	**0.85**
	ELECTRA + CNN	**0.80**	0.83	0.80	0.86	**0.85**	**0.85**
	ELECTRA + MLP	**0.80**	**0.85**	**0.81**	**0.87**	0.84	**0.85**
Testing—Founta							
Glove	Glove + CNN	0.91	0.91	0.91	0.92	0.92	0.92
	Glove + MLP	0.90	0.90	0.90	0.91	0.91	0.91
Transformers	Small BERT + CNN	0.93	**0.93**	**0.93**	0.93	0.93	0.93
	Small BERT + MLP	0.93	**0.93**	**0.93**	0.93	0.93	0.93
	ELECTRA + CNN	**0.94**	**0.93**	**0.93**	**0.94**	**0.94**	**0.94**
	ELECTRA + MLP	0.93	**0.93**	**0.93**	**0.94**	**0.94**	**0.94**
Testing—TSA							
Glove	Glove + CNN	0.50	0.50	0.42	**0.88**	0.57	0.68
	Glove + MLP	0.49	0.48	0.45	0.87	0.67	0.75
Transformers	Small BERT + CNN	**0.51**	**0.51**	0.49	0.87	0.78	0.82
	Small BERT + MLP	0.50	**0.51**	0.49	0.87	0.76	0.81
	ELECTRA + CNN	0.50	0.50	**0.50**	0.87	0.85	**0.86**
	ELECTRA + MLP	0.50	0.50	**0.50**	0.87	**0.86**	**0.86**

For each dataset, the best performance per column within each metric is boldfaced

Table 6 Scenario 3—Cross-domain experimental results with TSA as the source domain

Embedding	Model	Macro			Weighted avg.		
		P	**R**	F_1	**P**	**R**	F_1
Testing—Davidson							
Glove	Glove + CNN	0.45	0.48	0.45	0.58	0.67	0.61
	Glove + MLP	0.45	0.49	0.43	0.58	0.71	0.62
Transformers	Small BERT + CNN	0.70	**0.60**	**0.61**	0.74	0.77	**0.73**
	Small BERT + MLP	0.67	0.59	0.60	0.72	0.76	0.72
	ELECTRA + CNN	0.67	0.57	0.57	0.72	0.76	0.71
	ELECTRA + MLP	**0.75**	0.56	0.55	**0.76**	**0.77**	0.70
Testing—Founta							
Glove	Glove + CNN	0.48	0.49	**0.47**	0.51	0.55	0.52
	Glove + MLP	0.52	0.50	0.43	**0.55**	0.61	0.51
Transformers	Small BERT + CNN	**0.53**	**0.51**	0.45	**0.55**	0.61	0.53
	Small BERT + MLP	**0.53**	**0.51**	**0.47**	**0.55**	0.51	**0.54**
	ELECTRA + CNN	0.50	0.50	0.43	0.53	0.61	0.51
	ELECTRA + MLP	0.51	0.50	0.42	0.54	**0.62**	0.50
Testing—TSA							
Glove	Glove + CNN	0.66	0.84	0.71	0.94	0.88	0.90
	Glove + MLP	0.73	0.78	0.75	0.94	0.93	0.94
Transformers	Small BERT + CNN	0.85	0.86	0.86	0.96	0.96	0.96
	Small BERT + MLP	0.86	**0.88**	0.87	**0.97**	0.96	0.96
	ELECTRA + CNN	0.88	0.87	**0.88**	**0.97**	**0.97**	**0.97**
	ELECTRA + MLP	**0.91**	0.85	**0.88**	**0.97**	**0.97**	**0.97**

For each dataset, the best performance per column within each metric is boldfaced

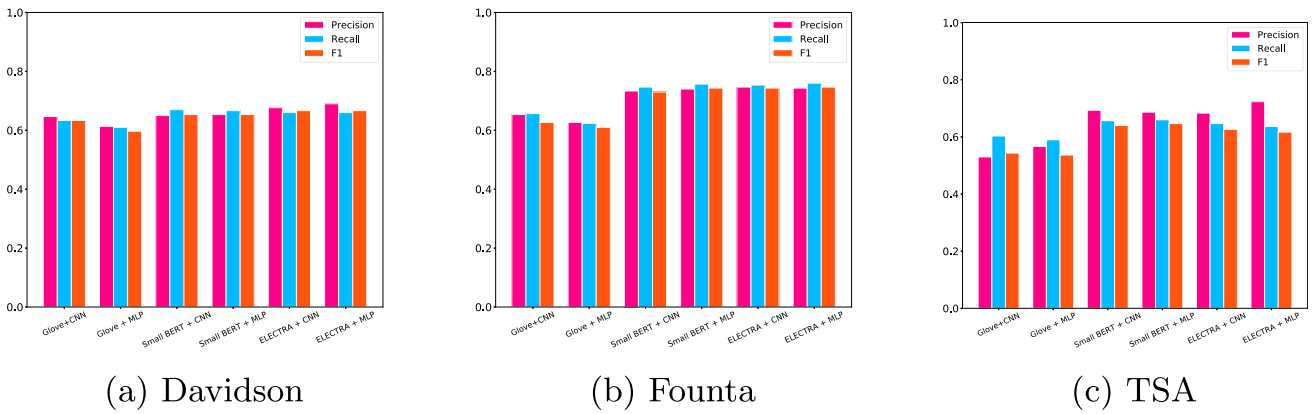

Fig. 6 Averaged detection performance across three cross-domain scenarios

MLP models. Glove-based models obtain comparative performance to the transformer-based models in this case.

When the testing is performed on TSA dataset, the results get much better across the models, as they are trained on the TSA training data. For the macro average measure, the highest performance is shared between Small BERT and ELECTRA with F_1 score ranging between 0.86 and 0.88. Glove provides the best performance with an F_1 score of 0.75 in combination with MLP. For the weighted average measure, the highest performance is also achieved by transformer embeddings-based models, with the averaged F_1 score between 0.96 and 0.97. Glove models come next with the F_1 score of 0.90 and 0.94 with CNN and MLP, respectively.

Averaged results across three scenarios. As shown by the averaged cross-domain experimental results of each method in Fig. 6, the transformer-based methods outperform other methods including Glove+MLP and Glove+CNN in terms of average generalization ability, among which ELECTRA + MLP yields the best performance in three metrics on most of the datasets. This further demonstrates that the transformer-based models can generalize better across different domains.

Also, the contextualized embeddings depending on the surrounding words generated by the transformer-based models (e.g., BERT, ELECTRA) are more effective than the static embeddings generated by Glove in domain generalization.

Overall observations. The transformer-based methods generally perform better than the Glove-based methods. The transformers enable the downstream detectors to obtain good performance in detecting non-hatred tweets across all three settings (as indicated by the performance in weighted average F_1 score) and the impressive performance on detecting hatred tweets in some scenarios. It is interesting that the cross-domain performance is not reversible in our results. For example, transformer-based methods work well when transferring knowledge from the Founta data to the Davidson data in Table 5, but they work poorly in the reverse case in Table 4. This may be because the Founta data is much larger than the Davidson data, containing knowledge of a broader hate speech scope (i.e., hate and abusive tweets) than Davidson (i.e., hate class only). As a result, the Founta data may contain hate speech knowledge relevant to that of Davidson, but the hatred tweets in the Davidson data may be not generalizable to that in Founta. This explains the different result patterns on Davidson and Founta in Tables 4 and 5.

TSA explicitly focuses on racist and sexism tweets, which is different from the other two datasets. As a result, it is difficult to adapt relevant knowledge from the other two datasets to the hate speech detection on TSA, as shown in Tables 4 and 5. Further, the models including the transformer-based methods, also fail to adapt relevant knowledge from TSA to Davidson and Founta, indicating nearly no shared knowledge of hatred tweets between TSA and the other two datasets.

5 Conclusions and discussions

This paper presents a large-scale empirical evaluation of 14 shallow and deep models for hate speech detection on three commonly-used benchmarks of different data characteristics. This is to provide important insights into their detection effectiveness, computational efficiency, capability in using pre-trained models, and domain generalizability for their deployment in real-world applications. Our conclusions based on the empirical results above are as following.

- *Detection effectiveness*. As shown in Table 2, the combination of BERT, ELECTRA, and Al-BERT and neural network-based classifiers perform consistently better than the other methods on the three benchmarks, especially in macro F_1 score. It is interesting that TF-IDF-enabled shallow classifiers (e.g., TF-IDF + XGB) can outperform Glove-enabled deep classifiers on most datasets.
- *Computational efficiency*. In terms of computational cost, transformers-enabled classifiers are more costly than the other models, as illustrated in Fig. 4. AI-BERT is the most time-consuming model among the transformers, followed by the BERT model. Small BERT is the most efficient transformer model in our task. Deep classifiers like Bi-LSTM are also computationally costly. When considering both effectiveness and efficiency, ELECTRA-based MLP models seem to be the most practical method, which achieves among the best classification performance while being sufficiently computationally efficient.
- *Capability in using pre-trained models*. As transformers are pre-trained on significantly larger corpora than Glove, they learn embedding space with richer semantics, empowering significantly better detection performance. This can be observed by the performance difference of having the same MLP-based classifier trained upon respective TF-IDF, Glove, and transformer-based representations, as shown in Table 2. However, it does not mean that the larger the pre-trained models are, the better performance the model would obtain. For example, the ELECTRA transformer can normally perform better than, or on par with, the larger transformer BERT on a number of cases in Table 1.
- *Domain generalizability*. Benefiting from the large-scale pre-training, 5 and 6, both Glove and transformers learn generalizable pre-trained embeddings, as shown in Tables 4. Again, due to the larger corpora used and the greater model capacity, the transformers-enabled classifiers gain stronger cross-domain generalization ability than the same Glove-based classifiers. It is interesting that the cross-domain performance may be not reversible due to the difference in the data size and the types of hatred classes covered in the source and target domains.

Although the transformer-based hate speech detectors show promising performance, they are still weak in terms of macro F_1 performance (e.g., the best performer gains a macro F_1 of 0.68 on the Founta dataset and 0.90 on the TSA dataset), indicating the great difficulty in achieving high precision and recall rates of identifying the minority hatred tweets from the massive tweets. Another major challenge lies in domain generalization ability is that it is difficult to collect a dataset that well covers all possible properties of hate speech in social media, which may lead to a domain difference between the source data and the target data. Thus, it is important for detectors to have a good domain generalization ability in practice. However, as shown in our results, there is still a large gap between the same-domain performance and the cross-domain performance. These two challenges present some important opportunities to further promote the development and deployment of hate speech detection techniques.

In addition, Large Language Models (LLMs) currently plays a significant role in the detection of hate speeches due to their ability to understand, process, and generate human-like texts. Although LLMs can significantly enhance the detection of hate speeches, e.g., by understanding context, handling multiple languages, and reducing bias. LLMs may struggle to detect such speech without explicit labels and examples. Also, LLMs may not keep up with latest concepts/topics/facts unless they are regularly updated and fine-tuned. Given the limitations of LLMs and the complexity of hate speech detection, it becomes crucial to synthesize LLMs and specialized hate speech detection datasets/models to develop more accurate hate speech detection.

References

1. Agrawal, S., Awekar, A.: Deep learning for detecting cyberbullying across multiple social media platforms. In: European Conference on Information Retrieval, pp. 141–153. Springer (2018)
2. Aizawa, A.: An information-theoretic perspective of tf-idf measures. Inf. Process. Manag. **39**(1), 45–65 (2003)
3. Arango, A., Pérez, J., Poblete, B.: Hate speech detection is not as easy as you may think: a closer look at model validation. In: Proceedings of the 42nd International ACM Sigir Conference on Research and Development in Information Retrieval, pp. 45–54 (2019)
4. Arco Plaza-del, F.M., Molina-González, M.D., Urena-López, L.A., Martín-Valdivia, M.T.: Comparing pre-trained language models for Spanish hate speech detection. Expert Syst. Appl. **166**, 114120 (2021)
5. Awal, M.R., Cao, R., Lee, R.K.W., Mitrovic., S.: Angrybert: joint learning target and emotion for hate speech detection. arXiv Preprint at arXiv:2103.11800 (2021)
6. Ayo, F.E., Folorunso, O., Ibharalu, F.T., Osinuga, I.A., Abayomi-Alli, A.: A probabilistic clustering model for hate speech classification in twitter. Expert Syst. Appl. **173**, 114762 (2021)
7. Badjatiya, P., Gupta, S., Gupta, M., Varma, V.: Deep learning for hate speech detection in tweets. In: Proceedings of the 26th International Conference on World Wide Web Companion, pp. 759–760 (2017)
8. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. arXiv Preprint at arXiv:1409.0473 (2014)
9. Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching word vectors with subword information. Trans. Assoc. Comput. Linguist. **5**, 135–146 (2017)
10. Burnap, P., Williams, M.L.: Cyber hate speech on twitter: an application of machine classification and statistical modeling for policy and decision making. Policy Internet **7**(2), 223–242 (2015)
11. Jose Camacho-Collados and Mohammad Taher Pilehvar: From word to sense embeddings: a survey on vector representations of meaning. J. Artif. Intell. Res. **63**, 743–788 (2018)
12. Cao R., Lee, R.K.W.: Hategan: adversarial generative-based data augmentation for hate speech detection. In: Proceedings of the 28th International Conference on Computational Linguistics, pp. 6327–6338 (2020)
13. Chen, Y., Zhou, Y., Zhu, S., Xu, H.: Detecting offensive language in social media to protect adolescent online safety. In: 2012 International Conference on Privacy, Security, Risk and Trust and 2012 International Confernece on Social Computing, pp. 71–80. IEEE (2012)
14. Chicco, D., Jurman, G.: The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. BMC Genom. **21**(1), 1–13 (2020)
15. Clark, K., Luong, M.-T., Le, Q.V., Manning, C.D.: Electra: pre-training text encoders as discriminators rather than generators. arXiv Preprint at arXiv:2003.10555 (2020)
16. Corazza, M., Menini, S., Cabrio, E., Tonelli, S., Villata, S.: A multilingual evaluation for online hate speech detection. ACM Trans. Internet Technol. (TOIT) **20**(2), 1–22 (2020)
17. Davidson, T., Warmsley, D., Macy, M., Weber, I.: Automated hate speech detection and the problem of offensive language. In: Proceedings of the International AAAI Conference on Web and Social Media, vol. 11 (2017)
18. Del Vigna12, F., Cimino23, A., Dell'Orletta, F., Petrocchi, M., Tesconi, M.: Hate me, hate me not: Hate speech detection on facebook. In: Proceedings of the First Italian Conference on Cybersecurity (ITASEC17), pp. 86–95 (2017)
19. Devlin, J., Chang, M.-W., Lee, K., Toutanova, K.: Bert: pre-training of deep bidirectional transformers for language understanding. arXiv Preprint at arXiv:1810.04805 (2018)
20. Fortuna, P., Nunes, S.: A survey on automatic detection of hate speech in text. ACM Comput. Surv. (CSUR) **51**(4), 1–30 (2018)
21. Founta, A.M., Djouvas, C., Chatzakou, D., Leontiadis, I., Blackburn, J., Stringhini, G., Vakali, A., Sirivianos, M., Kourtellis, N.: Large scale crowdsourcing and characterization of twitter abusive behavior. In: Twelfth International AAAI Conference on Web and Social Media (2018)
22. Gala, J., Gandhi, D., Mehta, J., Talat, Z.: A federated approach for hate speech detection. In: Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics, pp. 3248–3259 (2023)
23. Gambäck, B., Sikdar, U.K .: Using convolutional neural networks to classify hate-speech. In: Proceedings of the First Workshop on Abusive Language Online, pp. 85–90 (2017)
24. Gitari, N.D., Zuping, Z., Damien, H., Long, J.: A lexicon-based approach for hate speech detection. Int. J. Multimed. Ubiquitous Eng. **10**(4), 215–230 (2015)
25. Greevy, E., Smeaton, A.F.: Classifying racist texts using a support vector machine. In: Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 468–469 (2004)
26. Ibrohim, M.O., Budi, I.: Multi-label hate speech and abusive language detection in indonesian twitter. In: Proceedings of the Third Workshop on Abusive Language Online, pp. 46–57 (2019)
27. Jha, A., Mamidi, R.: When does a compliment become sexist? analysis and classification of ambivalent sexism using twitter data. In: Proceedings of the Second Workshop on NLP and Computational Social Science, pp. 7–16 (2017)
28. Joulin, A., Grave, E., Bojanowski, P., Douze, M., Jégou, H., Mikolov, T.: Fasttext.zip: Compressing text classification models. arXiv Preprint at arXiv:1612.03651 (2016)
29. Kapil, P., Ekbal, A.: A deep neural network based multi-task learning approach to hate speech detection. Knowl. Based Syst. **210**, 106458 (2020)
30. Kim, J., Jin, S., Park, S., Park, S., Han, K.: Label-aware hard negative sampling strategies with momentum contrastive learning for implicit hate speech detection. arXiv Preprint at arXiv:2406.07886 (2024)
31. Kwok, I., Wang, Y.: Locate the hate: detecting tweets against blacks. Proc. AAAI Conf. Artif. Intell. **27**, 1621–1622 (2013)
32. Le, Q., Mikolov, T.: Distributed representations of sentences and documents. In: International Conference on Machine Learning, pp. 1188–1196. PMLR (2014)
33. Lee, J., Lim, T., Lee, H., Jo, B., Kim, Y., Yoon, H., Han, S.C.: K-mhas: a multi-label hate speech detection dataset in korean online news comment. arXiv Preprint at arXiv:2208.10684 (2022)

34. MacAvaney, S., Yao, H.-R., Yang, E., Russell, K., Goharian, N., Frieder, O.: Hate speech detection: challenges and solutions. PloS One **14**(8), e0221152 (2019)
35. Mathew, B., Saha, P., Yimam, S.M., Biemann, C., Goyal, P., Mukherjee, A.: Hatexplain: a benchmark dataset for explainable hate speech detection. arXiv Preprint at arXiv:2012.10289 (2020)
36. Mehdad, Y., Tetreault, J.: Do characters abuse more than words? In: Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue, pp. 299–303 (2016)
37. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in Neural Information Processing Systems, pp. 3111–3119 (2013)
38. Modha, S., Majumder, P., Mandl, T., Mandalia, C.: Detecting and visualizing hate speech in social media: a cyber watchdog for surveillance. Expert Syst. Appl. **161**, 113725 (2020)
39. Mollas, I., Chrysopoulou, Z., Karlos, S., Tsoumakas, G.: Ethos: a multi-label hate speech detection dataset. Complex Intell. Syst. **8**(6), 4663–4678 (2022)
40. Mossie, Z., Wang, J.-H.: Vulnerable community identification using hate speech detection on social media. Inf. Process. Manag. **57**(3), 102087 (2020)
41. Mozafari, M., Farahbakhsh, R., Crespi, N.: A bert-based transfer learning approach for hate speech detection in online social media. In: International Conference on Complex Networks and Their Applications, pp. 928–940. Springer (2019)
42. Ni R., Cao, H.: Sentiment analysis based on gloVe and LSTM-GRU. In: 2020 39th Chinese Control Conference (CCC), pp. 7492–7497. IEEE (2020)
43. Nirmal, A., Bhattacharjee, A., Sheth, P., Liu, H.: Towards interpretable hate speech detection using large language model-extracted rationales. arXiv Preprint at arXiv:2403.12403 (2024)
44. Nobata, C., Tetreault, J., Thomas, A., Y., Mehdad, Chang, Y.: Abusive language detection in online user content. In: Proceedings of the 25th International Conference on World Wide Web, pp. 145–153 (2016)
45. Pamungkas, E.W., Basile, V., Patti, V.: Misogyny detection in twitter: a multilingual and cross-domain study. Inf. Process. Manag. **57**(6), 102360 (2020)
46. Park, J.H., Fung, P.: One-step and two-step classification for abusive language detection on twitter. arXiv Preprint at arXiv:1706.01206 (2017)
47. Pennington, J., Socher, R., Manning, C.D.: Glove: global vectors for word representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1532–1543 (2014)
48. Poletto, F., Basile, V., Sanguinetti, M., Bosco, C., Patti, V.: Resources and benchmark corpora for hate speech detection: a systematic review. Lang. Resour. Eval. **55**, 477–523 (2021)
49. Poulston, A., Waseem, Z., Stevenson, M.: Using tf-idf n-gram and word embedding cluster ensembles for author profiling: notebook for pan at clef 2017. In: CEUR Workshop Proceedings, vol. 1866. CEUR (2017)
50. Rini, R., Utami, E., Hartanto, A.D.: Systematic literature review of hate speech detection with text mining. In: 2020 2nd International Conference on Cybernetics and Intelligent System (ICORIS), pp. 1–6. IEEE (2020)
51. Sayar Ghosh Roy, Ujwal Narayan, Tathagata Raha, Zubair Abid, and Vasudeva Varma. Leveraging multilingual transformers for hate speech detection. arXiv preprint arXiv:2101.03207, 2021
52. Saroj, A., Mundotiya, R.K., Pal, S.: IRLab@ IITBHU at HASOC 2019: traditional machine learning for hate speech and offensive content identification. In: FIRE (Working Notes), pp. 308–314 (2019)
53. Schmidt, A., Wiegand, M.: 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 (2017)
54. Sharma, Y., Agrawal, G., Jain, P., Kumar, T.: Vector representation of words for sentiment analysis using glove. In: 2017 International Conference on Intelligent Communication and Computational Techniques (ICCT), pp. 279–284. IEEE (2017)
55. Zhong, T., Wenqiang, L., Li, Y., Zhao, W., Li, S.: Several alternative term weighting methods for text representation and classification. Knowl. Based Syst. **207**, 106399 (2020)
56. Tay, Y., Dehghani, M., Bahri, D., Metzler, D.: Efficient transformers: a survey. arXiv Preprint at arXiv:2009.06732 (2020)
57. Turc, I., Chang, M.-W., Lee, K., Toutanova, K.: Well-read students learn better: the impact of student initialization on knowledge distillation, vol. 13. arXiv Preprint at arXiv:1908.08962 (2019)
58. Vijayaraghavan, P., Larochelle, H., Roy, D.: Interpretable multi-modal hate speech detection. arXiv Preprint at arXiv:2103.01616 (2021)
59. Warner, W., Hirschberg, J.: Detecting hate speech on the world wide web. In: Proceedings of the Second Workshop on Language in Social Media, pp. 19–26 (2012)
60. Waseem, Z., Hovy, D.: Hateful symbols or hateful people? Predictive features for hate speech detection on twitter. In: Proceedings of the NAACL Student Research Workshop, pp. 88–93 (2016)
61. Xiang, G., Fan, B., Wang, L., Hong, J., Rose, C.: Detecting offensive tweets via topical feature discovery over a large scale twitter corpus. In: Proceedings of the 21st ACM International Conference on Information and Knowledge Management, pp. 1980–1984 (2012)
62. Yin, W., Zubiaga, A.: Towards generalisable hate speech detection: a review on obstacles and solutions. PeerJ Comput. Sci. **7**, e598 (2021)
63. Zhang, Z., Robinson, D., Tepper, J.: Detecting hate speech on twitter using a convolution-gru based deep neural network. In: European Semantic Web Conference, pp. 745–760. Springer (2018)
64. Zhong, H., Li, H., Squicciarini, A., Rajtmajer, S., Griffin, C., Miller, D., Caragea, C.: Content-driven detection of cyberbullying on the instagram social network. IJCAI Int. Joint Conf. Artif. Intell. **2016**, 3952–3958 (2016)