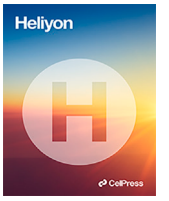
Heliyon 10 (2024) e29593

Contents lists available at ScienceDirect

Heliyon

journal homepage: www.cell.com/heliyon



Research article

Towards understanding the role of content-based and contextualized features in detecting abuse on Twitter

Kamal Hussain [a], Zafar Saeed [b], Rabeeh Abbasi [c,*], Muddassar Sindhu [c], Akmal Khattak [c], Sachi Arafat [d], Ali Daud [e], Mubashar Mushtaq [f]

[a] *Instituto Superior Técnico, Universidade de Lisboa, Portugal*
[b] *Dipartimento di Informatica, Università degli Studi di Bari, Bari, Italy*
[c] *Department of Computer Science, Quaid-i-Azam University, Islamabad, Pakistan*
[d] *Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah, Saudi Arabia*
[e] *Faculty of Resilience, Rabdan Academy, Abu Dhabi, United Arab Emirates*
[f] *Department of Computer Science, Forman Christian College, Lahore, Pakistan*

ARTICLE INFO

Dataset link: https://github.com/zeerakw/hatespeech

Keywords:
Abuse
Context
Machine learning
Social media
Twitter

ABSTRACT

This paper presents a novel approach for detecting abuse on Twitter. Abusive posts have become a major problem for social media platforms like Twitter. It is important to identify abuse to mitigate its potential harm. Many researchers have proposed methods to detect abuse on Twitter. However, most of the existing approaches for detecting abuse look only at the content of the abusive tweet in isolation and do not consider its contextual information, particularly the tweets posted before the abusive tweet. In this paper, we propose a new method for detecting abuse that uses contextual information from the tweets that precede and follow the abusive tweet. We hypothesize that this contextual information can be used to better understand the intent of the abusive tweet and to identify abuse that content-based methods would otherwise miss. We performed extensive experiments to identify the best combination of features and machine learning algorithms to detect abuse on Twitter. We test eight different machine learning classifiers on content- and context-based features for the experiments. The proposed method is compared with existing abuse detection methods and achieves an absolute improvement of around 7%. The best results are obtained by combining the content and context-based features. The highest accuracy of the proposed method is 86%, whereas the existing methods used for comparison have highest accuracy of 79.2%.

1. Introduction

Social media platforms allow users to share content and converse with each other [1]. However, with the increasing use of social media, negative content has become more common, such as bullying, trolling, harassment, and abuse. These issues can be challenging to address automatically, as they often depend on input from real users. Abusive language is prevalent on Twitter, and it can take many forms, such as racism, sexism, insults, profanity, vulgarity, and threats of violence [2]. Abuse is not only a personal

* Corresponding author.
E-mail addresses: k.hussain@cs.qau.edu.pk (K. Hussain), zafar.saeed@uniba.it (Z. Saeed), rabbasi@qau.edu.pk (R. Abbasi), masindhu@qau.edu.pk (M. Sindhu), akhattak@qau.edu.pk (A. Khattak), sachiarafat@gmail.com (S. Arafat), alimsdb@gmail.com (A. Daud), mubasharmushtaq@fccollege.edu.pk (M. Mushtaq).

https://doi.org/10.1016/j.heliyon.2024.e29593
Received 2 October 2023; Received in revised form 3 April 2024; Accepted 10 April 2024

Available online 16 April 2024

Table 1
Summary of keyword-based approaches.

Reference	Keyword-based Features
[15]	Continuous-bag of words
[18]	Pattern, Skip-gram
[15,18]	Ngram, Word2Vec
[18–20]	Sentiment, Lexicon, Polarity

detriment to the individual who is abused, but it can also have a negative impact on community relations. Abusive posts have become a major problem for social media platforms like Twitter. Thus, automatic detection of abuse is crucial to prevent real-life (physical or psychological) effects on individuals and communities.

Some social media platforms have reporting mechanisms in place, but these mechanisms can be ineffective, as they rely on users to report abuse. Additionally, users may be hesitant to report abuse for fear of retaliation. There is a need for more effective methods to detect and address abusive language on social media. One promising approach is to use machine learning to automatically identify abusive language. Machine learning algorithms can be trained on large datasets of labeled data to learn the patterns of abusive language. Once trained, these algorithms can be used to scan social media posts for signs of abuse.

The performance of such methods is limited relative to those that use contextual data beyond the content, such as user information, metadata, and other statistics. In user information, most of the existing studies have used the descriptions of user profiles and information about followees and followers [3]. Most of the existing studies have used hashtags, words, and mentions in the context of tweets. The context features in existing studies include tweet meta-data, user features, and network-based features [4–7]. The context features are more effective when the features help in identifying patterns.

In the literature, it is necessary to find the optimal combination of features from both content and context. In this paper, we develop several contextual features pertaining to user information and past tweets (within a sliding time window) that work to identify abusive tweets that are otherwise not identifiable by content alone. There is a marked performance improvement, particularly in accuracy, when content-based features from existing literature are combined with context-based features. The proposed combination of features is novel and, to the best of our knowledge, has not been examined in other works.

The main contributions of this research are:

- Novel features and their combinations for detecting abusive tweets.
- Detailed contextual feature engineering including content-based, user-based, and window-based feature sets.
- Evaluating the performance of feature sets individually, as well as with multiple combinations, for abusive tweet detection.
- Measurement of the performance of various classification algorithms and ensemble methods using the best-selected features.
- Comparing the results with state-of-the-art methods used for detecting abusive tweets.

The remainder of the paper is organized as follows: Section 2 discusses related work. Section 3 explains the proposed methodology and the feature extraction process. Section 4 elaborates on experimental setup and the dataset used for the experiments. Section 5 presents the results and compares the proposed method with state-of-the-art methods. Finally, Section 6 concludes the paper and provides direction for future work.

2. Related work

In this section, we discuss related work on abuse detection on Twitter. Like any other social media platform, Twitter suffers from various types of negative content, including abuse, cyberbully, trolling, hateful speech and fake news, among many others [5,4,8,9]. The severity of negativity has attracted researchers to identify negative content and users posting such content. This section discusses various approaches used to detect abusive content on Twitter.

2.1. Keyword-based approaches

Abuse identification on Twitter is usually performed based on the content of the tweets or the user profile. Detecting abusive tweets mainly includes keyword-based, content or tweets-based, and context-based approaches [10,11,2,3,12]. Recently, researchers have successfully used deep learning approaches [10,3,13,14] for solving a variety of problems.

Keyword or lexicon-based approaches employ word lists or dictionaries [15,16]. They compare every word with a lexicon that contains the sentiment or polarity associated with expressions [17–19]. In the sentiment approach, each word is given a score for each sentiment [18,19]. Tweets with overall negative sentiment are highlighted as potential abuse. Table 1 summarizes the linguistic features used in different researches.

2.2. Content-based approaches

Beyond the lexicon approach, the linguistic [18] and n-gram [18,19] approaches from the NLP techniques, and word embedding approach which is mostly combined with deep learning algorithms [15,18,21] are used for the identification of abusive words. Researchers in this area have focused on content-based detection to address the shortcomings of lexicon-based methods.

Table 2
Summary of content-based approaches.

Reference	Content-based Features
[2,10]	Linguistic, Syntactic
[10]	Semantic, Words length/average, Punctuations, Spelling errors
[10,22]	Syntactic (relation), Parts of Speech (POS) Tags
[2,10,11,27,28]	N-gram(Word): range(1 to 4)
[2,10,27,28]	N-gram (Character): range(3 to 8)
[2,10,22,28]	TF-IDF
[11]	Comment2vec
[27]	Word2vec
[2,28]	GLOVE
[11,28]	FastText

Abusive content-based approaches for Twitter incorporate different features. These features have been extracted using natural language processing (NLP) techniques that include linguistic features [2], syntactic features [5], semantic features [10], statistical features [22,2], and language models [23].

The linguistic features contain several subcategories to use as features such as text length, average number of words, number of punctuation marks, repeated words, grammar, and spelling mistakes [2]. Syntactic features check the relationship of the words [2,24] and part of speech (POS) tags [2]. These features make tuples of words and capture the dependency relation between words [5]. Semantic features check the similarity between words using different methods [15,24], such as cosine similarity [15] and edit distance [24]. In this case, statistical features are used to transform the data into vectors as required for their classification. These features include n-grams [25], $TF\text{-}IDF$ [2] and word embedding [4]. N-gram has been used at the word level [25,3] and the character level [26,2]. At the word level, unigram, bigram, trigram, and even tetragram have also been used [25,26]. The character-level n-grams have been used mostly in the range of 2-8 grams [26]. $TF\text{-}IDF$ has also been used with n-grams to normalize the weight of each word [2,22]. Content-based features are summarized in Table 2.

2.3. Context-based approaches

Context-based approaches have focused on features that exploit relations between tweet features [5,25], user features [8,3], and network features [29]. User features are based on user profile information, whereas network features are usually obtained from users' connectivity in their social networks. For instance, the following and followship network of user [29,8,3].

The features of the tweet are categorized into hashtags, mentions, URLs, retweets, favorites, replies, words, average tweet words, syllables, and sentiment score [5,25,8]. Most of the tweet features consist of numeric count values such as the number of retweets and favorites. Besides, textual features such as words [30], hashtags [31], and inappropriate words [5,8] are also used as tweet features.

The user features constitute characteristics of the user profile. These features have been used with the combination of user profile information, such as followers and follower counts, favorites, lists, account age, geographical location, profile URL, profile image, user description, and account verification [32,31,3,8,33,34].

Some of the user features consist of count values such as the number of followers and number of followee [4]. Other include textual features such as screen name [30], user profile description [35,5] and geo-location [31].

In user features, the relationships have been checked to find the ratio, difference, and similarity between the connected users [36].

The network features are extracted from the tweet, and the user features pertain to the connections and relations between the users or other entities. These connections are taken as contextual features in several studies [30]. The features of the network estimate the connectivity and similarity between users using social network analysis methods with centrality measures [4,5,37]. For example, the number of followers and the number of followees connections, user retweet connections, and connections between followers and followees [4,8,38,3].

Finally, the sequential features comprise latent information of time. For example, tweets posted by a user in the past one hour. These features are based on the temporal sequence of multiple tweets. Such latent temporal information plays a key role in discovering useful patterns to exploit the data. Many existing studies [39–42] use sequential features to detect unusual and anomalous behavior from temporally interlinked data. The temporal context is usually derived from sliding windows moving across the data stream. In this study, we have also used the sliding windows to explore the sequential feature. Besides, we have intuitively introduced the temporal history of a tweet by incorporating the number of tweets posted previously within a specific time interval. Context-based features are summarized in Table 3.

2.4. Abuse detection methods

Various machine learning methods, including deep learning, supervised, semi-supervised, unsupervised, and regression, have been used to detect Twitter abuse using selections of the features discussed in the previous sections. Moreover, social network analysis and graph-based features have also been used. Specific examples include linear classification [22,2], ensemble methods [26,22], clustering [5,2], similarity [15,38], neighborhood [8,43], convolutional, and recurrent neural networks [2,44].

Table 3
Summary of contextual features.

Reference	Context Used
[3]	Profile images/description/location/URLs
[31,4,8,30]	Followees, Followers, Graph-based features
[38]	Hashtags, Mentions, Retweets, Replies, Verified Account, Favorites, Listed, Account Age, Graph-based features

Table 4
Summary of the existing datasets.

Reference	Details	Size	Labels	Used By
[22]	Abusive language and hate speech	24,802 Tweets	Hateful, Offensive and Normal	[27,4,11]
[31]	User behavior (cyber bullying)	9,484 Tweets, 1303 Users	Aggressive, bullying, spam and normal	[4]
[61]	Abusive language and hate speech	16,914 Tweets, 1236 Users	Sexism, Racism and Normal	[4,27,11,29]
[62]	Sarcasm	61,075 Tweets	Sarcasm and non-sarcasm	[4]
[63]	Manually labeled harassment tweets	35,000 Tweets	Harassing and non-harassing	[64,4]

Table 5
Summary of the existing studies categorized into various approaches.

Reference	Approach
[15,18–20]	Keyword-based Approaches
[2,10,22,11,27,28]	Content-based Approaches
[3,31,4,8,30,38]	Context-based Approaches
[5,43,45–48,54,55]	Other Approaches

A probabilistic clustering method along with fuzzy classification is used to detect hate speech [45] on Twitter. Other approaches use deep learning and pre-trained embeddings to identify and visualize hate on Twitter [46] and Facebook [46,47]. In addition to Twitter and English, researchers have also focused on other social media platforms and languages to detect hate speech. For example, [48] detect hate speech from Instagram comments posted in the Turkish language.

In addition to traditional machine learning algorithms, meta-heuristic algorithms like Ant Colony Optimization [49], Ant Lion Optimization [50], Moth Flame Optimization [51], Social Spider Optimization [52], and Tunicate Swarm Algorithm [53] have also been used for detecting hate speech [54,55]. Use of other recent meta-heuristics like Red Deer Algorithm [56] and [57] need to be explored for detecting hate speech. Optimization algorithms can also be explored for their potential use in detecting abusive tweets [58–60].

The proliferation of such approaches meant the creation of appropriate datasets from platforms such as Twitter, Facebook, Flickr, Yahoo, and Blog websites [28,10,19]. For Twitter, existing datasets include those that have been extracted using its API [5,4,38,15,3], some of which are labeled and not all of them publicly available. Those that are labeled and publicly available include [61], [22], Temporal dataset [10], WWW15 dataset [10], and cyberbullying [5]. Out of these, we conduct our experiments on the most frequently used dataset for abuse and hate speech detection published by [61]. Different datasets available in the literature are listed in Table 4.

Table 5 summarizes the literature and different approaches used for detecting abuse and hate-speech on Twitter.

None of the above mentioned methods derive context from retrospective temporal data to detect abuse on Twitter. The existing literature also lacks in exploring the effect of various types of features on the performance of detection methods. The method proposed in this research covers this gap and shows that a combination of context and content can be helpful in detecting abuse on Twitter.

3. Methodology

This section presents our proposed methodology for Twitter abuse detection. Furthermore, it also describes feature engineering and supervised learning methods that include model parameters for experiments. The workflow for the proposed approach is visually presented in Fig. 1. The tweets are crawled from Twitter for the dataset provided by [61]. The dataset is then cleaned in preprocessing steps (see Section 4). The feature extractor finds content-based (see Section 3.2) and context-based (see Section 3.3) features from the tweets. A total of nine supervised learning methods (see Section 3.4) are used for the experiments. The performance is evaluated using stratified k-fold cross-validation.

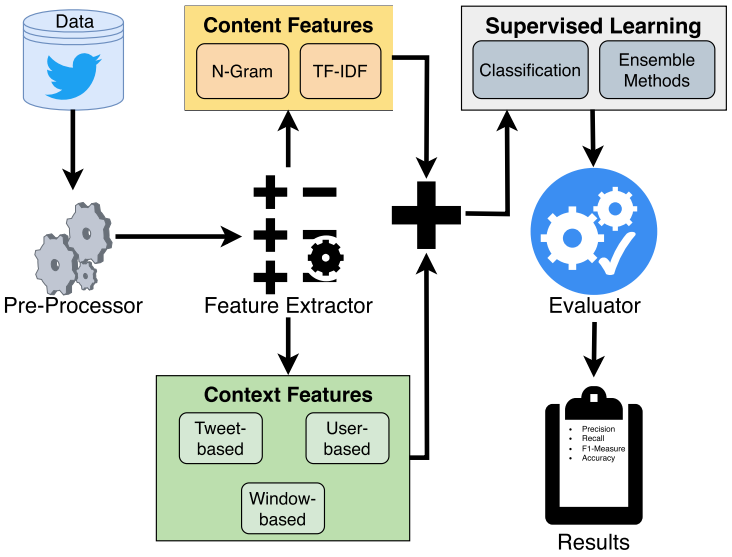


Fig. 1. Workflow of the proposed approach.

3.1. Features design

The following sections describe content-based and context-based features extracted from tweet, user, and retrospective windows. Features are designed based on contexts and contents. Let $D = \{\tau_1, \tau_2, \ldots, \tau_m\}$ be the set of tweets available in the dataset. $W = \{w_1, w_2, \ldots, w_k\}$ be a set of distinct words extracted from tweets contained in the dataset D. $U = \{u_1, u_2, \ldots, u_n\}$ be the users who posted at least one tweet.

3.2. Content-based feature

Natural Language Processing (NLP) is the simplest way to detect abusive language. NLP techniques can directly process tweet content involving textual attributes to detect abusive tweets. Tweets' content is represented in the Bag-of-Word (BoW) model with statistical features to capture the notion of abusive words, which are helpful in detecting abusive language.

A set of words $W = \{w_1, w_2, \ldots, w_k\}$ is extracted from the text of tweets to form a vocabulary. NLP pre-processing techniques and linguistic modules are applied to the text of tweets to get word types or unigrams. Statistical features are further used to make them machine-understandable. Term Frequency–Inverse Document Frequency $TF\text{-}IDF$ [65] as shown in Eq. (1), is used to compute the weight for each uni-gram feature.

$$TF - IDF = tf_{w_i,\tau_j} \times idf_{w_i} \tag{1}$$

The *tf* gives importance to frequently occurring terms. We have used max normalization for *tf* with the smoothing term set to 0.5, as shown in Eq. (2).

$$tf_{w_i,\tau_j} = \frac{1}{2} + \frac{1}{2} \times \frac{count_{w_i,\tau_j}}{\max\limits_{1 \le i \le |W|} \{count_{w_i,D}\}} \tag{2}$$

where $count_{w_i,\tau_j}$ is the number of times word w_i occurs in tweet τ_j and $count_{w_i,D}$ is the word frequency in complete dataset. The IDF gives importance to rare terms and estimated as shown in Eq. (3).

$$idf_{w_i} = log \frac{N}{df_{w_i}} \tag{3}$$

where N is the total number of tweets in corpus D and df_{w_i} is the number of tweets in which word w_i occurs at least once.

3.3. Contextual features

This section discusses the contextual features for each tweet $\tau_i \in D$ consisting of a 5-tuple as shown in Eq. (4).

$$\tau_i = (t, \overline{W}, \Gamma, \Pi, \Lambda) \tag{4}$$

Table 6
List of contextual features used in the proposed approach.

Context	Features	Description
Tweet	Retweets	Number of retweets
	Like	Number of likes on a tweet.
	Mentions	List of mentions in a tweet.
	Hashtags	List of hashtags in a tweet
	URLs	List of URLs in a tweet
User	Follower	Number of users followed to a user
	Followees	Number of users followed by a user
	Like	Numbers of tweets a user has liked
	List	Number of lists a user has created
	Description	The user's profile description
Window-Based	Previous Tweets	Takes past tweets which is based on sliding window
	Temporal Tweets	Takes past tweets which is based on temporal window

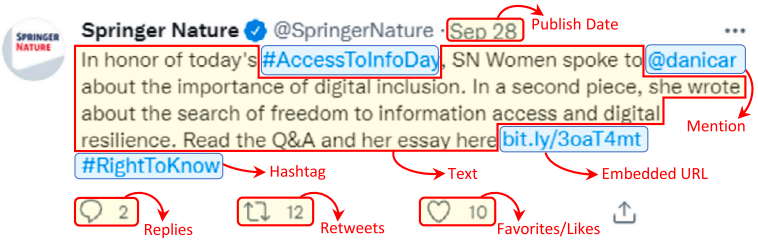


Fig. 2. Example of tweet features.

The 5-tuple derives the complete context for each tweet, where t is posting time of tweet τ_i, $\overline{W} \subset W$ denotes the set of words in a tweet, Γ represents the combination of tweets characteristics (see Section 3.3.1), Π represents the user characteristics (see Section 3.3.2) and Λ denotes the window-based characteristic (see Section 3.3.3).

These features are useful in identifying abusive tweets more accurately and efficiently. The important features can be seen in Table 6. Features such as retweets, favorites, mentions, hashtags, and URLs extracted from tweets are considered tweet features. Features such as followers, followees, likes, lists, and description are considered user-based features. Historical tweets using sliding and temporal windows are defined as window-based features that are investigated in combination with tweet and user-based features.

3.3.1. Tweet features

Tweet features Γ are the attributes expressing tweet information. Γ contains retweet count, like count, replies count, a list of mentions, a list of hashtags, and a list of URLs. These features support the content to detect abusive tweets. These features are based on the popularity of the tweets. Each Γ corresponding to the tweet $\tau_i \in D$ is a 5-tuple defined in Eq. (5).

$$\Gamma_{\tau_i} = (\#rt, \#lk, \#mn, \#ht, \#url) \tag{5}$$

where $\#rt$ represents the number of times a tweet is retweeted, $\#lk$ represents the number of times a tweet is liked, $\#mn$ represents the number of users mention in a tweet, $\#ht$ represents the number of hashtags in a tweet, and $\#url$ the number of URLs in tweet τ_i. These tweet features can be seen in Fig. 2. After a careful preliminary investigation, only important and contributing features are selected in the proposed approach.

3.3.2. User features

User features describe the various characteristics of a user over. The features Π corresponding to the tweet $\tau_i \in D$ are defined in Eq. (6). Π consists of a 5-tuple, consisting of the number of followers, number of followees, number of lists, number of likes, and profile description of user u_i who posted tweet τ_i.

$$\Pi_{\tau_i} = (\#fo, \#fe, \#lt, \#lk, \omega) \tag{6}$$

The attributes of a user are extracted to measure user popularity. An example Twitter user handle marked with visible attributes is shown in Fig. 3. The popularity of a user shows active connectivity within the social network. The user description, defined as $\omega = tf \times idf$, is computed statistically as discussed in Section 3.2. This feature is also used in content features.

3.3.3. Window features

Two types of windows are designed to extract historical context. The first type is sliding window P^{τ_i} that consider k number of historical tweets of user u_i who posted tweet τ_i. The second type is temporal window S^{τ_i} that accounts Δt as time interval and considers all historical tweets during Δt concerning tweet τ_i posted by user u_i to derive the context from retrospective stream.

Fig. 3. Examples of user features.

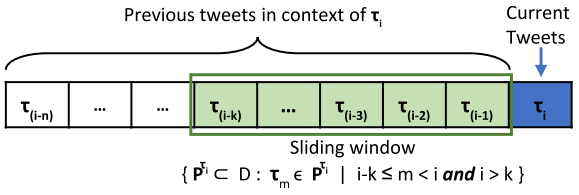


Fig. 4. Historical tweet collection based on sliding window of size k.

These features contain the past tweets for each current tweet τ_i. The features based on past tweets effectively detect current abusive tweets due to retrospective information. A user may have an inherent tendency to post tweets with abusive content. Therefore, window-based features containing historical tweets of a user serve as a context to classify the current tweets as either abusive or normal. The classification methods perform poorly if current tweets are confusing or ambiguous. This ambiguity of tweets can be removed with the consideration of past tweets. The tweets are considered abusive if the past tweets within the sliding P^{τ_i} and temporal window S^{τ_i} are also abusive. In addition to this, the contents of abusive tweets within the specific window frame are mostly similar. The uni-gram features based on content are computed using $TF\text{-}IDF$ measure shown in Eq. (1) and described in Section 3.2.

Sliding window

Sliding window P^{τ_i} contains a set of tweets collected with a temporal context to classify the current tweet τ_i posted by user u_i. Fig. 4 shows the formal design of a sliding window of size k and formulated using Eq. (7).

$$\{P^{\tau_i} \subset D : \tau_m \in P^{\tau_i} \mid i-k \leq m < i \ \wedge \ i > k\} \tag{7}$$

where D is the tweet corpus, τ_m are the historical tweets in the sliding window P^{τ_i}, i and m represent the i^{th} and m^{th} indexes of tweets in the D, and k represent the size of sliding window P^{τ_i} for tweet τ_i.

In our experiments, we used different sliding window sizes to extract the historical tweets. The sliding window size is set to 5, 10, 15 and 20 to investigate the effectiveness and performance of the classification methods on different sizes of sliding windows. Algorithm 1 describes the extraction of historical tweets for a sliding window P^{τ_i}

Algorithm 1 Sliding Window Features.

Require: D, τ_i, and k ▷ tweet corpus in temporal order, current tweet and size of sliding window, respectively.

Ensure: P^{τ_i} and Λ

$P^{\tau_i} \leftarrow [\,]$ ▷ Set of tweets qualified for the sliding window
for each τ_m ***in*** D posted by user u_i prior to τ_i **do**
  $P^{\tau_i}.Add(\tau_m)$
  $k \leftarrow count + 1$
  if $k = 0$ **then**
    break loop
  end if
end for
Extract sliding-window tweet features from P^{τ_i} and merge into Λ

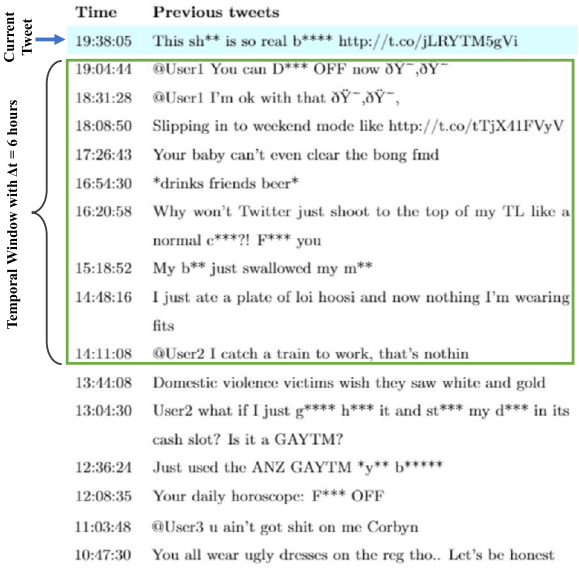


Fig. 5. An example of temporal windows with time interval $\Delta t = 6$ hours.

Temporal Window

Temporal window S^{τ_i} contains the historical tweets of user u_i using time interval Δt in context of τ_i and formally defined in Eq. (8).

$$\{S^{\tau_i} \subset D : \tau_m \in S^{\tau_i} \mid \pi_0(\tau_i) - \pi_0(\tau_{i-k}) \leq \Delta t\} \tag{8}$$

where $\pi_0(\tau_{i-k})$ and $\pi_0(\tau_i)$ are the posting date and time of the earliest tweet τ_{i-k} and current tweet τ_i covered in the time interval Δt for temporal window S^{τ_i}. Tweet features based on the past tweets of users within a specific time window are useful to model the users' abusive behavior. Three different intervals (i.e., 6, 12, and 48 hours) for Δt of the temporal window S^{τ_i} are considered in the experiments to test the effectiveness and performance of detection methods. The date and time of tweets are normalized into seconds to collect the historical tweets using a temporal window. After the tweet date and time normalization, the time difference δt between current tweet τ_i and past tweets of user u_i is computed. In our experiments, temporal window S^{τ_i} contains tweets posted prior to 6, 12, or 48 hours of the current tweet. This process can be seen in Algorithm 2. Fig. 5 shows an example of abusive tweets within the temporal window with $\Delta t = 6$.

Algorithm 2 Temporal Window Features.

Require: D, τ_i, and Δt
▷ tweet corpus in temporal order, current tweet and coverage of temporal window, respectively.

Ensure: S^{τ_i} and Λ

$P^{\tau_i} \leftarrow [\,]$ ▷ Set of tweets qualified for the sliding window
$S^{\tau_i} \leftarrow [\,]$

for each τ_m ***in*** D posted by user u_i prior to τ_i **do**

  $\delta t \leftarrow TimeDifference(\pi_0(\tau_i), \pi_0(\tau_m))$ ▷ calculating the temporal distance between τ_i and τ_m

  if $\delta t > \Delta t$ **then**
    break loop
  end if

  $S^{\tau_i}.Add(\tau_m)$ ▷ add previously posted tweet τ_m in temporal window
end for

Extract temporal-window tweet features from S^{τ_i} and merge into Λ

3.4. Supervised machine learning techniques

Supervised learning, Logistic Regression (LR) [66], Support Vector Machine (SVM) [67] and Naive Bayes (NB) [68] are used as classification models to detect abusive tweets. Based on empirical analysis, linear kernel SVM is used with margin maximization

Table 7
Class distribution of dataset on tweets and users.

Class	Tweets	Users
Sexism	3,383	613
Racism	1,972	9
None	11,559	614
Total	16,914	1236

parameter $C = 1$, and the regularization parameter C is used to avoid overfitting during training. NB classifier is used with additive smoothing parameter α, values of $\alpha \geq 0$ are used for avoiding overfitting and zero probabilities, the value of $\alpha = 0$ is called Laplace smoothing, whereas $\alpha \leq 1$ is called Lidstone smoothing. Embedding tweet text results in a sparse feature set; hence, it can cause the learning model to overfit due to the high variance. A higher value of α smoothens the probability estimates and reduces the impact of individual features. Therefore, we set the value of $\alpha = 1$. For the LR classifier, $L2$ regularization is used with error minimization parameter $C = 1$ to avoid overfitting. The ensemble methods, Random Forest (RF), AdaBoosting, Gradient Tree Boosting (GradBoost), XGBoosting, LightBoost and Bagging, are also used for the detailed comparison. The number of trees in RF and number of boosting stages in GradBoost methods are set to 35. Usually, a large number of trees increases the accuracy of the classifiers, however at the same time, they also increase the training time. Increasing the number of trees after a certain number (35 in our case) does not significantly improve the results. A large number of stages increases accuracy, and a small learning rate affects the smooth optimization of the objective function over gradient descent and prevents overshooting. Therefore, the number of boosting stages and learning rate are set as 100 and 0.1, respectively, for the AdaBoosting, GradBoosting, XGBoosting, LightBoost and Bagging_Classifier.

Several measures can be used to evaluate the performance of classifiers. This study uses precision, recall, f-measure, and accuracy for the performance evaluation and comparison of different methods on novel contextual features for abusive tweet detection. Additionally, stratified k-fold cross-validation is used to split the corpus in training and testing data.

4. Experimental setup

This section discusses the datasets, state-of-the-art baseline techniques for abusive tweet detection, and evaluation metrics. The existing dataset is explored in this section. The mechanism for data collection, filtration, and cross-validation method for training and testing are also explained. The proposed approach is compared with the baseline techniques introduced by [2] and [22] for abusive tweet detection. The baseline techniques are implemented and tested on the dataset used in this research work. The following sections also discuss the feature set and algorithm of these baseline techniques. The research work of [2] investigates the dynamics involved in offensive language and introduces a technique to detect hate speech automatically. Tweet contents with some additional features are used in their work. Statistical measure, known as $TF\text{-}IDF$, is used to compute the score for each feature extracted from tweet content. The additional features include sentiment score, word count, character count and syllable count. Different classifiers, including Linear-SVM, Naive Bayes, Logistic Regression, Random Forest, and Gradient Boosting are used to detect hateful speech. The research work of [22] uses tweet content. The word-level and character-level features were extracted from tweet content. The *importance score* from tweet content is computed using $TF\text{-}IDF$ term weighting scheme. The word-level features include uni, bi, tri, and tetra-gram. Character-level 3-8 gram features were also used. Supervised machine learning including Linear-SVM, Naive Bayes, Logistic Regression, Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN).

The focus of this research is on abusive tweet detection. With respect to our choice of the Twitter platform, we found the benchmark dataset in [61] has widely been used for detecting abusive language and is opted for this research as well. The dataset was constructed from a corpus of 136,052 tweets. It contains 16,914 annotated tweets posted by 1,239 unique and distinct users. The annotated tweets are categorized into three classes: sexism, racism and normal, as shown in Table 7. It contains 35.3% abusive tweets, 7.3% unique users, 20.0% sexism, 10.1% racism and 68.3% normal tweets [61]. The original dataset contained tweets with three labels. The dataset was modified according to Table 7 to obtain contextual information.

The data collection strategy is based on tweet information, user information, and past tweets. Tweet attributes are extracted from tweet information. Tweet attributes include user IDs, text, user, and entities as shown in Fig. 6. User profile attributes are retrieved through Twitter API for user information. The attributes include user id, screen name, location, user description, followees and followers as shown in Fig. 6.

Twitter API strictly follows the rate limit[1] for the data acquisition. Table 8 shows the number of tweets that are retrieved through Twitter API with the rate limit policy. Furthermore, user IDs were used to retrieve the users past tweets. The rate limit of Twitter API restricts access within the past 7-15 days only. But, when a user's tweets are too far back, the restricted access on the retrospective stream would not collect that data. We required the historical tweets of each user to define contextual features; therefore, it was necessary to collect past tweets that were way beyond the access limit of publicly available Twitter API. We used the Twitter-Scraper to solve this problem, as it can access all previous tweets if they are publicly accessible and reachable.

The benchmark acquired from [61] has three class labels. The tweet distribution concerning class labels is shown in Table 8. To map the dataset onto our problem domain, the sexism and racism classes are merged to a single *Abusive* class as both sexism and racism represent abuse. The class *Normal* is considered as *Non-abusive* class.

[1] http://web.archive.org/web/20210327063011/ https://developer.twitter.com/en/docs/rate-limits Latest access date April 12, 2024.

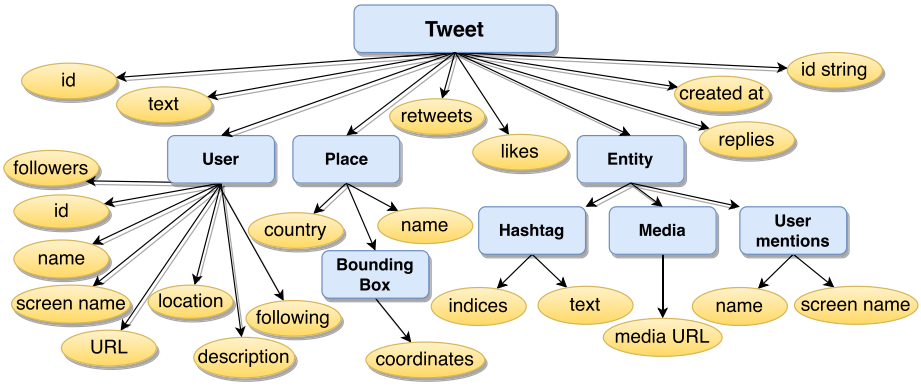


Fig. 6. Tweets attributes or information.

Table 8
The dataset collected using the benchmark [61].

Class	Tweets	Users
Sexism	2699	203
Racism	261	1
Normal	8069	463
Total	11029	667

Table 9
Filtered dataset with respect to Abusive and Non-abusive classes.

Class	Tweets	Users
Abusive	2960	204
Non-abusive	4452	458
Total	7412	662

Besides, the dataset was inherently class imbalanced with 27% and 73% of the tweets belonging to *Abusive* and *Non-abusive* classes, respectively. We filtered the data by dropping five users along with their tweets. The dropped users account for nearly 45% of the non-abusive tweets, contributing to a significant class imbalance in the data. The crawled accessible data contained 11,029 tweets. However, with the filtration process, the tweets set is reduced to 7,412 with 40% and 60% tweets belonging to *abusive* and *Non-abusive* classes, respectively. The tweet distribution in the final dataset is shown in Table 9. After filtration, the data contained 7,412 tweets posted by 662 unique users. 204 distinct users posted a total of 2960 abusive tweets. Non-abusive tweets account for 4452 in total posted by 458 unique users as shown in Table 9.

Tweets are then pre-processed by converting tweets to lowercase, replacing tokens into base words using Porter stemmer[2] and wordnet lemmatizer.[3] In the pre-processing, common and stop words, special characters, symbols, punctuation and emojis are also removed from the tweets.

The data split technique to distribute the dataset into testing and training data is called cross-validation in machine learning. Different strategies include the holdout method, k-fold cross-validation and Stratified k-fold cross-validation. Instead of a widely used k-fold cross-validation, stratified k-fold cross-validation is used due to class imbalance in the dataset. It takes all labels in every fold iteration and performs better than k-fold cross-validation. We have used 10-fold stratified cross-validation for evaluation.

During classification, the results are obtained in the form of correct classifications, i.e., True Positives (TP) True Negatives (TN), and false classifications, i.e., False Positives (FP) and False Negatives (FN). These classifications are used in the evaluation measures Precision (Eq. (9)), Recall (Eq. (10)), F1-measure (Eq. (11)), and Accuracy (Eq. (12)) for statistically analyzing different results obtained in this research.

$$Precision = \frac{TP}{TP + FP} \tag{9}$$

$$Recall = \frac{TP}{TP + FN} \tag{10}$$

[2] https://web.archive.org/web/20190627112454/ http://snowball.tartarus.org/algorithms/porter/stemmer.html Lasted access date April 12, 2024.

[3] https://web.archive.org/web/20190711193825/ https://www.machinelearningplus.com/nlp/lemmatization-examples-python/ Lasted access date April 12, 2024.

$$F1 - Measure = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{11}$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{12}$$

5. Results and discussion

In this section, we evaluate the performance of our proposed method on a dataset of Twitter tweets and compare them with existing methods for detecting abuse on Twitter. Each contextual feature set is compared with all other individual features. In the comparative analysis, the combination of feature sets producing the best results is selected for the proposed method. The proposed method is then compared with the state-of-the-art research work of [22] and [2]. A comparison of the feature set of the proposed approach is drawn with the content-based approach for the evaluation. This section analyzes feature sets and methods in terms of precision, recall, f1 measure and accuracy.

Recent studies [10,18,37,69] show that the research on abusive language detection can be categorized as keyword, content and context-based detection. This research work has explored content-based features with three different contextual features containing historical tweets, users, sliding, and temporal window-based information. Different combinations of contextual features are explored. The results for each contextual feature are compared with one another. Furthermore, the contextual features are combined with content-based features to examine the impact of different combinations. The combination of feature sets that give the best results is selected to compare with the state-of-the-art techniques.

Content-based approaches are often used to identify abuse using the textual content of tweets. This research investigates the combination of contextual features based on various attributes of tweet, user, and time-window. First, the contextual features based on historical tweets, users, temporal, and sliding windows are analyzed independently. The performance in terms of precision, recall, f1-measure and accuracy are measured for these contextual features along with the content-based features.

The contextual features, extracted from historical tweets include mention, hashtag, URL, retweet, and favorite counts. The user-based contextual features comprise user information and include follower, following, favorite, like counts, and user descriptions. The sliding window is based on the last 5, 10, 15 and 20 tweets posted prior to the current reference tweet. The temporal window is based on the previous tweets posted within the last 6, 12 and 48 hours prior to the current reference tweet.

Table 10 shows the performance of different feature sets for the problem of abusive language detection. For the content-based feature set, the highest precision, recall, f1 measure and accuracy observed are 80%, 78%, 76% and 77%, respectively. For tweet-based contextual features, the best precision achieved is 78% and for the contextual features extracted from user profile, the best precision is 88% as shown in Table 10. Thus, a decrease of 2% and an increase of 8% in precision is observed for tweet and user profile-based contextual features, respectively. Similarly, comparing user profile-based features with content-based features shows improvement of 9%, 8% and 8% in recall, f1-measure and accuracy, respectively. This improvement is quite substantial. Tweet-based features show no improvement in recall, whereas f1-measure and accuracy were improved by only 1%. This shows that the contextual features based on user profile information distinguishe abusive tweets better.

Ensemble methods known as GradBoost, XGBoost, LightGBM and classifier naive Bayesian achieve the highest precision of 80% for the identification of abusive tweets based on the textual content of tweets. The Naive Bayesian classifier achieves the highest recall, f1-score and accuracy for abusive tweets classification based on content as shown in Table 10. For tweet-based contextual features, SVM outperforms all other classifiers. Ensemble method XGBoost achieved highest precision, recall, f1-score and accuracy of 88%, 85%, 84% and 85%, respectively, for user profile-based contextual features. The highest recall was also achieved by SVM, ADABoost and GradBoost, as shown in Table 10. Similarly, the highest f1-score for user profile-based contextual features was also shared by SVM, naive Bayesian, logistic regression, random forest, ADABoost and GradBoost. The highest accuracy of 85% was also achieved by GradBoost, as shown in Table 10.

The sliding window intervals k for historical tweets are set as 5, 10, 15 and 20 for the experiments. The classification accuracy is measured for each classifier against every interval, as shown in Fig. 7.

The average for each sliding window is computed which are 79.48% (for $k = 5$), 79.51% (for $k = 10$), 79.61% (for $k = 15$) and 79.41% (for $k = 20$). The highest accuracy of 81.6% was observed for the naïve Bayes classifier as shown in Fig. 7. Since the sliding window with $k = 15$ has the highest average accuracy, we select $k = 15$ for the sliding window size to further explore the impact of the sliding window when combined with the different feature sets. Temporal window is different from sliding window. Here, instead of having a fixed number of past tweets, the temporal coverage (Δt) of past 6, 12, and 48 hours is considered for collecting historical tweets for each tweet. The classifiers result for each temporal window is shown in Fig. 8.

The average accuracy for each temporal window (6 hours, 12 hours and 48 hours) is 78.8%, 78.3%, and 79.3%, respectively. Temporal window with $\Delta t = 48$ outperforms other temporal windows of 6 and 12 hours. The best temporal window, which in our case is 2 days, is chosen for further experimentation. The highest accuracy achieved is 82% when logistic regression is used, as shown in Fig. 8.

The results of each classifier for the content, tweet, and user-based features are shown in Table 10. Furthermore, based on their performances, we have selected the best classifiers for each feature set. The best results for each feature set are compared to exhibit their compound usefulness with certain classifiers. Fig. 9 shows that the results for user-based features outperform all other features using Gradient Boost algorithm. However, in general, it can be seen from Fig. 9 that contextual features achieve higher accuracy than content-based features.

The discussion above paves the way to explore the combination of features. The performance of detection method based on combination of contextual and content features is measured. This combination was our proposed feature set. It can be seen in

Table 10
Performance of classifiers on content and context (tweet and user information) based feature set.

Classifiers		Precision	Recall	F1-Score	Accuracy
Content-based Approach					
Individual Classifiers	SVM	78%	76%	75%	76.20%
	NB	**80%**	**78%**	**76%**	**77%**
	LR	78%	76%	74%	75.60%
Ensemble Methods	RF	75%	75%	73%	74.60%
	ADABoost	79%	74%	71%	74.30%
	GradBoost	**80%**	75%	72%	74.60%
	Bagging	71%	72%	71%	71.50%
	XGBoost	**80%**	74%	71%	74%
	LightGBM	**80%**	74%	71%	73.40%
Tweet Features					
Individual Classifiers	SVM	**78%**	**78%**	**77%**	**78.0%**
	NB	75%	75%	75%	75.4%
	LR	77%	77%	76%	77.0%
Ensemble Methods	RF	76%	77%	76%	76.5%
	ADABoost	74%	74%	74%	74.5%
	GradBoost	76%	76%	75%	76.0%
	Bagging	73%	74%	73%	73.5%
	XGBoost	76%	76%	75%	76.0%
	LightGBM	77%	77%	76%	77.0%
User Features					
Individual Classifiers	SVM	86%	**85%**	**84%**	84.6%
	NB	86%	84%	**84%**	84.4%
	LR	85%	84%	**84%**	84.0%
Ensemble Methods	RF	85%	84%	**84%**	84.0%
	ADABoost	87%	**85%**	**84%**	84.7%
	GradBoost	87%	**85%**	**84%**	**85.0%**
	Bagging	84%	83%	82%	82.8%
	XGBoost	**88%**	**85%**	**84%**	**85.0%**
	LightGBM	86%	84%	83%	84.2%

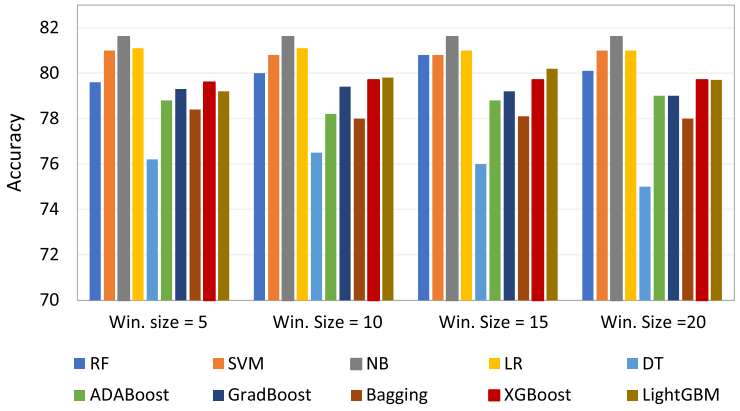


Fig. 7. Performance of classifiers against sliding windows with size $k = 5$, 10, 15, and 20.

Table 11 that the feature set based on the combination of context and content-based features outperforms all other individual features in terms of precision, recall, f1-score and accuracy. Fig. 10 shows that the proposed method outperforms all other state-of-the-art techniques. The highest accuracy achieved for the proposed method is 86% when logistic regression was used, whereas the highest precision and f1-score of 88% and 85%, respectively, is achieved when naive Bayesian classifier is used. Similarly, the highest recall of 86% is recorded with logistic regression.

The proposed method is compared with two baseline methods [22] and [2]. Both baseline methods achieved an accuracy of 79.2% and 75.1%. The proposed method based on content, user, and time-window features outperformed all other state of the art methods. This improvement in accuracy varies from 9% to 10%, which is quite significant, as shown in Fig. 10. Table 12 shows the detailed results of the baseline methods.

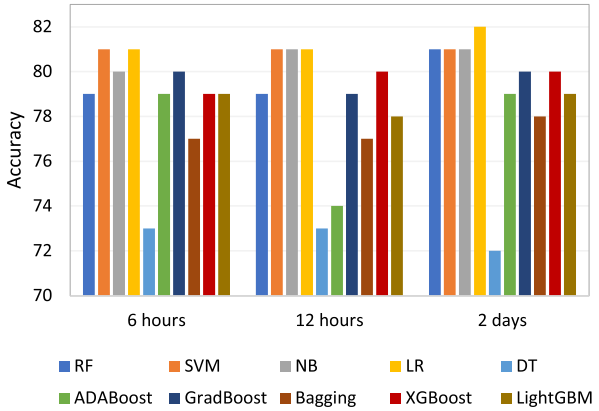


Fig. 8. Performance of classifiers against temporal windows with time interval $\Delta t = 6$, 12, and 48 hours.

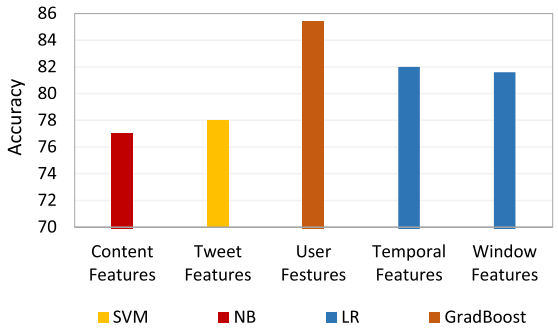


Fig. 9. Comparison of the content-based approach and individual contextual features.

Table 11
Performance comparison of the proposed approach combining the contextual features with content-based approach.

	Classifiers	Prec-ision	Recall	F1-Score	Accu-racy
Individual Classifiers	SVM	85%	84%	84%	84%
	NB	**88%**	85%	**85%**	85.3%
	LR	**88%**	**86%**	**85%**	**86%**
Ensemble Methods	RF	86%	85%	**85%**	85%
	AdaBoost	85%	84%	83%	84.0%
	GradientBoost	**88%**	85%	**85%**	85.3%
	Bagging	81%	81%	81%	81.4%
	XGBoost	**88%**	85%	**85%**	85.4%
	LightGBM	86%	84%	84%	85%

Table 12
Performance (Accuracy) comparison of the baseline techniques used for abuse detection.

Methods	Lee et al., [2] (Word-level)	Lee et al., [2] (Character-level)	Davidson et al., [22]
SVM	68.70%	75.10%	69%
NB	64.90%	65.60%	74.80%
LR	64.50%	70.80%	**75.90%**
DT			67.90%
RF	69.10%	70.60%	74.70%
GradBoost	63.10%	65.30%	
CNN	77.30%	**75.60%**	
RNN	**79.20%**	56.30%	

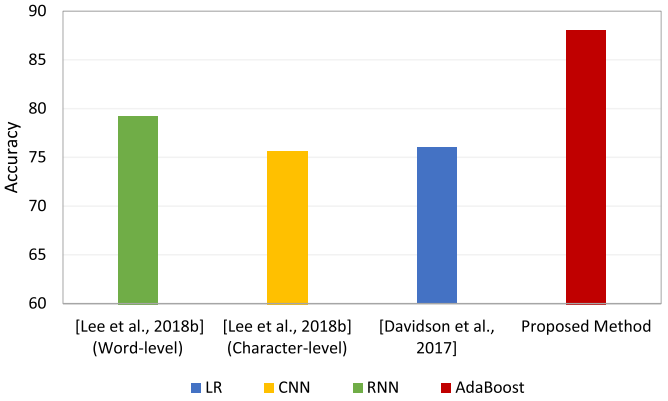


Fig. 10. Comparison of the proposed approach with state-of-the-art.

The contextual features performed better and are more helpful than content-based features in distinguishing abusive tweets with high accuracy. The reason for this can be the use of ambiguous words and sentences, and sometimes malicious users misuse other accounts. Therefore, this situation demands the use of contextual features to detect abusive tweets more accurately. The important features are extracted from all individual feature sets through evaluation measures. Combining these significant user, tweet and window-based features is analyzed in this research work. Most of the contextual features refer to popularity. For instance, followees, following, retweets and other aspects are often considered to be popularity. User-based features outperform other features because all user attributes capture the notion of popularity except user descriptions. The attributes of user features distinguish abusive and non-abusive in a better way. The users posting abusive tweets often have an insignificant number of followers and lists. Tweet-based features combine features based on tweet content and popularity. Therefore, tweet-based features often lag behind user-based features in terms of performance when identifying abusive tweets. The pattern of abusive and non-abusive tweets is not recognized using tweet-based features. One of the reasons for the negative impact of tweet-based features on performance can be the problem of overfitting because some features, which include mentions, hashtags, and URLs, can result in overfitting. In window-based features, the combination of features is extracted based on the tweets posted prior to current tweets in a dataset. These features can further boost the detection of abusive tweets if all of the tweets posted prior to current tweets are accessible. Unfortunately, this is not the case currently. Therefore, a few past accessible tweets are extracted. The contextual features are combined more effectively and accurately. Contextual features improve the performance of abusive language identification in terms of accuracy.

6. Conclusions and future work

This paper aimed to detect abusive language using context and content-based features. Previous studies use three different approaches word-, content- and context-based detection. These techniques have some limitations. Word-based detection suffers from the inclusion of new words that are not available in a word list or dictionary. Another limitation of the word-based approach is the set of ambiguous words in the dataset leading to inaccurate detection of abusive tweets. Contextual features including tweet, user, and network features have been recently to detect abusive language. The pattern and distribution of features play a crucial role in abuse detection.

We deduced an optimal combination of features from the categories of tweet information, user information, and past tweets, with variations of sliding window approaches. Supervise machine learning algorithms are used and evaluated to show the effectiveness of the proposed method. The algorithms include individual classifiers and ensembles. We compared our proposed methods with two baseline methods, showing that our proposed method outperforms the state-of-the-art methods.

The bottleneck of the proposed window-based methods is the unavailability of the past and deleted tweets. The results of window-based feature set were affected. The performance of our proposed approach could have shown more improvement if the missing past tweets were accessible. There are some potential future directions to explore this area further and gain insight. Images and videos associated with tweets can be used by deriving textual representation. Firstly, we could use images and videos associated with tweets by deriving textual representations for them. Content of referring web pages and URLs can be helpful. There is a need for research to predict abuse instead of detecting it. Target abuse detection towards a community is another area to explore. Multi-lingual abuse detection is another dimension that needs attention. Similarly, cyberbullying and hate speech to instigate violence against a community are some other potential future directions.

Funding

The authors declare that no funds, grants, or other support were received during the preparation of this manuscript.

Ethics approval

Review and/or approval by an ethics committee was not needed for this study because publicly available dataset was used and no experiments involving humans were conducted.

CRediT authorship contribution statement

Kamal Hussain: Writing – original draft, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Zafar Saeed:** Writing – review & editing, Methodology, Formal analysis. **Rabeeh Abbasi:** Writing – review & editing, Writing – original draft, Validation, Supervision, Methodology, Investigation, Formal analysis, Conceptualization. **Muddassar Sindhu:** Writing – review & editing, Formal analysis. **Akmal Khattak:** Writing – review & editing, Investigation, Conceptualization. **Sachi Arafat:** Writing – review & editing. **Ali Daud:** Writing – review & editing, Methodology, Conceptualization. **Mubashar Mushtaq:** Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Rabeeh Abbasi is an *Associate Editor for Heliyon (Society and Politics section)* and was not involved in the editorial review or the decision to publish this article.

Data availability

Data used in this research can be downloaded from the online repository: https://github.com/zeerakw/hatespeech.

Acknowledgements

The author Zafar Saeed is funded by the project FAIR - Future AI Research (PE00000013), Spoke 6 - Symbiotic AI, under the NRRP MUR program funded by NextGenerationEU.

References

[1] Z. Saeed, R.A. Abbasi, O. Maqbool, A. Sadaf, I. Razzak, A. Daud, N.R. Aljohani, G. Xu, What's happening around the world? A survey and framework on event detection techniques on Twitter, J. Grid Comput. 17 (2) (2019) 279–312.
[2] Y. Lee, S. Yoon, K. Jung, Comparative studies of detecting abusive language on Twitter, in: Proceedings of the 2nd Workshop on Abusive Language Online (ALW2), Brussels, Belgium, Association for Computational Linguistics, Oct. 2018, pp. 101–106.
[3] E. Fehn Unsvåg, B. Gambäck, The effects of user features on Twitter hate speech detection, in: Proceedings of the 2nd Workshop on Abusive Language Online (ALW2), Brussels, Belgium, Association for Computational Linguistics, Oct. 2018, pp. 75–85.
[4] A. Founta, D. Chatzakou, N. Kourtellis, J. Blackburn, A. Vakali, I. Leontiadis, A unified deep learning architecture for abuse detection, CoRR, arXiv:1802.00385 [abs], 2018.
[5] D. Chatzakou, N. Kourtellis, J. Blackburn, E.D. Cristofaro, G. Stringhini, A. Vakali, Measuring #gamergate: a tale of hate, sexism, and bullying, CoRR, arXiv: 1702.07784 [abs], 2017.
[6] M.A. Masood, R.A. Abbasi, N. Wee Keong, Context-aware sliding window for sentiment classification, IEEE Access 8 (2020) 4870–4884.
[7] A. Said, T.D. Bowman, R.A. Abbasi, N.R. Aljohani, S.U. Hassan, R. Nawaz, Mining network-level properties of Twitter altmetrics data, Scientometrics 120 (1) (2019) 217–235.
[8] M.H. Ribeiro, P.H. Calais, Y.A. Santos, V.A.F. Almeida, W.M. Jr., "Like sheep among wolves": characterizing hateful users on Twitter, CoRR, arXiv:1801.00317 [abs], 2018.
[9] O. Warke, J.M. Jose, J. Breitsohl, Utilising Twitter Metadata for Hate Classification, Lecture Notes in Computer Science, 2023, pp. 676–684.
[10] C. Nobata, J. Tetreault, A. Thomas, Y. Mehdad, Y. Chang, Abusive language detection in online user content, in: Proceedings of the 25th International Conference on World Wide Web, WWW '16, Republic and Canton of Geneva, CHE, in: International World Wide Web Conferences Steering Committee, 2016, pp. 145–153.
[11] J.H. Park, P. Fung, One-step and two-step classification for abusive language detection on Twitter, CoRR, arXiv:1706.01206 [abs], 2017.
[12] W. Yin, A. Zubiaga, Hidden behind the obvious: misleading keywords and implicitly abusive language on social media, Online Soc. Netw. Media 30 (2022) 100210.
[13] M.K. Hayat, A. Daud, A.A. Alshdadi, A. Banjar, R.A. Abbasi, Y. Bao, H. Dawood, Towards deep learning prospects: insights for social media analytics, IEEE Access 7 (2019) 36958–36979.
[14] C. Zhan, X. Zhang, J. Yuan, X. Chen, A.M. Fathollahi-Fard, C. Wang, J. Wu, G. Tian, A hybrid approach for low-carbon transportation system analysis: integrating critic-dematel and deep learning features, Int. J. Environ. Sci. Technol. 21 (Jan 2024) 791–804.
[15] H.S. Lee, H.R. Lee, J.U. Park, Y.S. Han, An abusive text detection system based on enhanced abusive and non-abusive word lists, Decis. Support Syst. 113 (December 2017) (2018) 22–31.
[16] E.W. Pamungkas, V. Basile, V. Patti, Investigating the role of swear words in abusive language detection tasks, Lang. Resour. Eval. 57 (Feb 2022) 155–188.
[17] A. Banjar, Z. Ahmed, A. Daud, R.A. Abbasi, H. Dawood, Aspect-based sentiment analysis for polarity estimation of customer reviews on Twitter, Comput. Mater. Continua 67 (2) (2021) 2203–2225.
[18] M. Wiegand, J. Ruppenhofer, A. Schmidt, C. Greenberg, Inducing a lexicon of abusive words – a feature-based approach, in: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), New Orleans, Louisiana, Association for Computational Linguistics, June 2018, pp. 1046–1056.
[19] N.D. Gitari, Z. Zuping, H. Damien, J. Long, A lexicon-based approach for hate speech detection, Int. J. Multimed. Ubiquitous Eng. 10 (4) (2015) 215–230.
[20] S. Choudhury, J.G. Breslin, User sentiment detection: a YouTube use case, in: The 21st National Conference on Artificial Intelligence and Cognitive Science (AICS 2010), 2010.

[21] G. del Valle-Cano, L. Quijano-Sánchez, F. Liberatore, J. Gómez, Socialhaterbert: a dichotomous approach for automatically detecting hate speech on Twitter through textual analysis and user profiles, Expert Syst. Appl. 216 (2023) 119446.
[22] T. Davidson, D. Warmsley, M.W. Macy, I. Weber, Automated hate speech detection and the problem of offensive language, CoRR, arXiv:1703.04009 [abs], 2017.
[23] J.C. Aguerri, L. Molnar, F. Miró-Llinares, Old crimes reported in new bottles: the disclosure of child sexual abuse on Twitter through the case #metooinceste, Soc. Netw. Anal. Min. 13 (Feb 2023).
[24] Z. Waseem, T. Davidson, D. Warmsley, I. Weber, Understanding abuse: a typology of abusive language detection subtasks, in: Proceedings of the First Workshop on Abusive Language Online, Vancouver, BC, Canada, Association for Computational Linguistics, Aug. 2017, pp. 78–84.
[25] H. Watanabe, M. Bouazizi, T. Ohtsuki, Hate speech on Twitter: a pragmatic approach to collect hateful and offensive expressions and perform hate speech detection, IEEE Access 6 (c) (2018) 13825–13835.
[26] A. Sharma, A. Nandan, R. Ralhan, An investigation of supervised learning methods for authorship attribution in short hinglish texts using char & word n-grams, CoRR, arXiv:1812.10281 [abs], 2018.
[27] A. Gaydhani, V. Doma, S. Kendre, L. Bhagwat, Detecting hate speech and offensive language on Twitter using machine learning: an n-gram and TFIDF based approach, CoRR, arXiv:1809.08651 [abs], 2018.
[28] E. Fehn Unsvåg, B. Gambäck, The effects of user features on Twitter hate speech detection, in: Proceedings of the 2nd Workshop on Abusive Language Online (ALW2), Brussels, Belgium, Association for Computational Linguistics, Oct. 2018, pp. 75–85.
[29] G.K. Pitsilis, H. Ramampiaro, H. Langseth, Detecting offensive language in tweets using deep learning, CoRR, arXiv:1801.04433 [abs], 2018.
[30] N. Cécillon, V. Labatut, R. Dufour, G. Linarès, Abusive language detection in online conversations by combining content- and graph-based features, Front. Big Data 2 (2019) 8.
[31] D. Chatzakou, N. Kourtellis, J. Blackburn, E.D. Cristofaro, G. Stringhini, A. Vakali, Hate is not binary: studying abusive behavior of #gamergate on Twitter, CoRR, arXiv:1705.03345 [abs], 2017.
[32] M. Casavantes, M.E. Aragón, L.C. González, M. Montes-y Gómez, Leveraging posts' and authors' metadata to spot several forms of abusive comments in Twitter, J. Intell. Inf. Syst. (Feb 2023).
[33] S. Tuarob, M. Satravisut, P. Sangtunchai, S. Nunthavanich, T. Noraset, Falcon: detecting and classifying abusive language in social networks using context features and unlabeled data, Inf. Process. Manag. 60 (4) (2023) 103381.
[34] R. Song, F. Giunchiglia, Q. Shen, N. Li, H. Xu, Improving abusive language detection with online interaction network, Inf. Process. Manag. 59 (5) (2022) 103009.
[35] P. Mathur, R. Shah, R. Sawhney, D. Mahata, Detecting offensive tweets in Hindi-English code-switched language, in: Proceedings of the Sixth International Workshop on Natural Language Processing for Social Media, Melbourne, Australia, Association for Computational Linguistics, July 2018, pp. 18–26.
[36] A. García-Recuero, Discouraging abusive behavior in privacy-preserving online social networking applications, in: Proceedings of the 25th International Conference Companion on World Wide Web, WWW '16 Companion, Republic and Canton of Geneva, CHE, in: International World Wide Web Conferences Steering Committee, 2016, pp. 305–309.
[37] N. Tahmasbi, E. Rastegari, A socio-contextual approach in automated detection of public cyberbullying on Twitter, Trans. Soc. Comput. 1 (Dec. 2018).
[38] Á. García-Recuero, A. Morawin, G. Tyson, Trollslayer: crowdsourcing and characterization of abusive birds in Twitter, CoRR, arXiv:1812.06156 [abs], 2018.
[39] Q. Wang, J. She, T. Song, Y. Tong, L. Chen, K. Xu, Adjustable time-window-based event detection on Twitter, in: Web-Age Information Management - 17th International Conference, WAIM 2016, Nanchang, China, June 3-5, 2016, Proceedings, Part II, 2016, pp. 265–278.
[40] Z. Saeed, R.A. Abbasi, M.I. Razzak, G. Xu, Event detection in Twitter stream using weighted dynamic heartbeat graph approach [application notes], IEEE Comput. Intell. Mag. 14 (3) (2019) 29–38.
[41] Z. Saeed, R.A. Abbasi, I. Razzak, O. Maqbool, A. Sadaf, G. Xu, Enhanced heartbeat graph for emerging event detection on Twitter using time series networks, Expert Syst. Appl. 136 (2019) 115–132.
[42] Y. Wang, C. Goutte, Detecting changes in Twitter streams using temporal clusters of hashtags, in: Proceedings of the Events and Stories in the News Workshop, Vancouver, Canada, Association for Computational Linguistics, Aug. 2017, pp. 10–14.
[43] J. Chen, S. Yan, K.C. Wong, Verbal aggression detection on Twitter comments: convolutional neural network for short-text sentiment analysis, Neural Comput. Appl. 0 (2018) 1–10.
[44] S. Zimmerman, U. Kruschwitz, C. Fox, Improving hate speech detection with deep learning ensembles, in: Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018), Miyazaki, Japan, European Language Resources Association (ELRA), May 2018.
[45] F.E. Ayo, O. Folorunso, F.T. Ibharalu, I.A. Osinuga, A. Abayomi-Alli, A probabilistic clustering model for hate speech classification in Twitter, Expert Syst. Appl. 173 (2021) 114762.
[46] S. Modha, P. Majumder, T. Mandl, C. Mandalia, Detecting and visualizing hate speech in social media: a cyber watchdog for surveillance, Expert Syst. Appl. 161 (2020) 113725.
[47] F.M.P. del Arco, M.D. Molina-González, L.A. Ureña-López, M.T. Martín-Valdivia, Comparing pre-trained language models for Spanish hate speech detection, Expert Syst. Appl. 166 (2021) 114120.
[48] H. Karayiğit, Çiğdem İnan Acı, A. Akdağlı, Detecting abusive Instagram comments in Turkish using convolutional neural network and machine learning methods, Expert Syst. Appl. 174 (2021) 114802.
[49] M. Dorigo, M. Birattari, T. Stutzle, Ant colony optimization, IEEE Comput. Intell. Mag. 1 (4) (2006) 28–39.
[50] S. Mirjalili, The ant lion optimizer, Adv. Eng. Softw. 83 (2015) 80–98.
[51] S. Mirjalili, Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm, Knowl.-Based Syst. 89 (2015) 228–249.
[52] A. Luque-Chang, E. Cuevas, F. Fausto, D. Zaldívar, M. Pérez, Social spider optimization algorithm: modifications, applications, and perspectives, Math. Probl. Eng. 2018 (Dec 2018) 6843923.
[53] S. Kaur, L.K. Awasthi, A. Sangal, G. Dhiman, Tunicate swarm algorithm: a new bio-inspired based metaheuristic paradigm for global optimization, Eng. Appl. Artif. Intell. 90 (2020) 103541.
[54] C. Baydogan, B. Alatas, Metaheuristic ant lion and moth flame optimization-based novel approach for automatic detection of hate speech in online social networks, IEEE Access 9 (2021) 110047–110062.
[55] S. Gite, S. Patil, D. Dharrao, M. Yadav, S. Basak, A. Rajendran, K. Kotecha, Textual feature extraction using ant colony optimization for hate speech classification, Big Data Cogn. Comput. 7 (1) (2023).
[56] A.M. Fathollahi-Fard, M. Hajiaghaei-Keshteli, R. Tavakkoli-Moghaddam, Red deer algorithm (rda): a new nature-inspired meta-heuristic, Soft Comput. 24 (Oct 2020) 14637–14665.
[57] A.M. Fathollahi-Fard, M. Hajiaghaei-Keshteli, R. Tavakkoli-Moghaddam, The social engineering optimizer (seo), Eng. Appl. Artif. Intell. 72 (2018) 267–293.
[58] Y. Fu, X. Ma, K. Gao, Z. Li, H. Dong, Multi-objective home health care routing and scheduling with sharing service via a problem-specific knowledge-based artificial bee colony algorithm, IEEE Trans. Intell. Transp. Syst. 25 (2) (2024) 1706–1719.
[59] Y. Fu, M. Zhou, X. Guo, L. Qi, Scheduling dual-objective stochastic hybrid flow shop with deteriorating jobs via bi-population evolutionary algorithm, IEEE Trans. Syst. Man Cybern. Syst. 50 (12) (2020) 5037–5048.
[60] Z. Zhang, M. Zhao, H. Wang, Z. Cui, W. Zhang, An efficient interval many-objective evolutionary algorithm for cloud task scheduling problem under uncertainty, Inf. Sci. 583 (2022) 56–72.
[61] Z. Waseem, D. Hovy, Hateful symbols or hateful people? Predictive features for hate speech detection on Twitter, in: Proceedings of the NAACL Student Research Workshop, San Diego, California, Association for Computational Linguistics, June 2016, pp. 88–93.

[62] A. Rajadesingan, R. Zafarani, H. Liu, Sarcasm detection on Twitter: a behavioral modeling approach, in: Proceedings of the Eighth ACM International Conference on Web Search and Data Mining, WSDM '15, New York, NY, USA, Association for Computing Machinery, 2015, pp. 97–106.
[63] J. Golbeck, Z. Ashktorab, R.O. Banjo, A. Berlinger, S. Bhagwan, C. Buntain, P. Cheakalos, A.A. Geller, Q. Gergory, R.K. Gnanasekaran, R.R. Gunasekaran, K.M. Hoffman, J. Hottle, V. Jienjitlert, S. Khare, R. Lau, M.J. Martindale, S. Naik, H.L. Nixon, P. Ramachandran, K.M. Rogers, L. Rogers, M.S. Sarin, G. Shahane, J. Thanki, P. Vengataraman, Z. Wan, D.M. Wu, A large labeled corpus for online harassment research, in: Proceedings of the 2017 ACM on Web Science Conference, WebSci '17, New York, NY, USA, Association for Computing Machinery, 2017, pp. 229–233.
[64] R. Kshirsagar, T. Cukuvac, K.R. McKeown, S. McGregor, Predictive embeddings for hate speech detection on Twitter, CoRR, arXiv:1809.10644 [abs], 2018.
[65] G. Salton, C. Buckley, Term-weighting approaches in automatic text retrieval, Inf. Process. Manag. 24 (5) (1988) 513–523.
[66] S. Symeonidis, D. Effrosynidis, A. Arampatzis, A comparative evaluation of pre-processing techniques and their interactions for Twitter sentiment analysis, Expert Syst. Appl. 110 (2018) 298–310.
[67] T. Joachims, Text categorization with support vector machines: learning with many relevant features, in: C. Nédellec, C. Rouveirol (Eds.), Machine Learning: ECML-98, Springer Berlin Heidelberg, Berlin, Heidelberg, 1998, pp. 137–142.
[68] S. Xu, Bayesian naïve Bayes classifiers to text classification, J. Inf. Sci. 44 (1) (2018) 48–59.
[69] T. Chakrabarty, K. Gupta, Context-aware attention for understanding Twitter abuse, CoRR, arXiv:1809.08726 [abs], 2018.