International Journal of Data Science and Analytics (2025) 20:4489–4500
https://doi.org/10.1007/s41060-025-00730-1

REGULAR PAPER

Enhancing online toxicity detection on gaming networks: a novel embeddings-based valence lexicon approach

Heba Ismail[1] · **Ashraf Khalil**[1] · **Ahmed Jasmy**[2]

Received: 10 October 2024 / Accepted: 29 January 2025 / Published online: 19 February 2025

Abstract
Online toxicity and violent speech on gaming networks pose significant threats to societal well-being, particularly among adolescents, and are linked to severe consequences such as suicide. This highlights an urgent need for effective toxicity detection methods tailored to these platforms. Traditional rule-based approaches are inherently limited, and the performance of predictive models in detecting online toxicity is critically dependent on the quality and representativeness of their training data. However, the distinct linguistic characteristics of discourse on gaming networks present unique challenges in curating representative training samples using existing valence lexicons, often resulting in suboptimal detection accuracy. In this study, we propose a novel framework for online toxicity detection on gaming platforms that addresses these linguistic challenges. Our approach introduces an extended embeddings-based valence lexicon that can be customized to any gaming platform. In this study, we specifically target Twitch. Through a comprehensive comparative evaluation against state-of-the-art lexicon-based techniques, the proposed method demonstrates superior performance. Notably, agreement analysis reveals a moderate alignment between human annotators and the proposed method, with a kappa score of 0.619. Experimental results further underscore the efficacy of our approach, showing an average performance improvement of 31.47% across LSTM, GRU, and CNN architectures compared to all baseline methods. The findings of this study have significant implications for improving platform-specific toxicity detection, enhancing the safety and inclusivity of gaming environments. The framework can be adapted to other online platforms, offering a scalable solution to address toxicity and protect vulnerable populations.

Keywords Online toxicity detection · Toxicity prediction · Implicit valence · Word embeddings · Deep learning

1 Introduction

The global cybergaming industry is expected to reach 3.22 billion players by 2024, indicating substantial growth [1]. This expansion coincides with a concerning rise in cyberbullying incidents, violent speech, and online toxicity in general, especially among adolescents. A 2020 survey conducted in the USA found that 68% of multiplayer gamers reported experiencing harassment and abuse [2], which inflicts significant harm, including mental distress, social isolation, and potential physical harm, on its victims [3]. To mitigate this, gaming platforms employ community guidelines and moderation tools, often leveraging predictive models trained on samples of potentially offensive content, to enhance user safety [4]. However, existing online toxicity detection training samples predominantly rely on explicit valence indicators defined in valence lexicons [5–7], disregarding the prevalence of implicit toxicity characterized by sarcasm, slogans, emojis, and special characters on gaming platforms [8]. This accentuates more representative training samples capturing both explicit and implicit indicators of online toxicity. The emotional orientation of text, inferred from its context, has garnered significant interest due to its profound implications across domains such as well-being [9–11], health [12], and education [13, 14], where it can offer valuable insights. Similarly, in the field of online toxicity detection, grasping the

✉ Heba Ismail
heba.ismail@zu.ac.ae

Ashraf Khalil
ashraf.khalil@zu.ac.ae

Ahmed Jasmy
ahmad.jasmy@adu.ac.ae

[1] College of Technological Innovations, Zayed University, Abu Dhabi, UAE

[2] Department of Computer Science and Information Technology, College of Engineering, Abu Dhabi University, Abu Dhabi, UAE

valence orientation of text as inferred from its context holds paramount importance. Twitch is one of the most popular gaming streaming platforms with up to 30 million daily active users [15]. Therefore, this study focuses on analyzing Twitch chat messages and comments for online toxicity detection. However, the proposed framework can be used on any chatting platform.

Existing studies on online toxicity detection rely on either manually annotated training samples [16], or annotations based on explicit rules and linguistic markers defined in valence lexicons [5–7]. While manual annotation effectively captures implicit toxicity indicators, it is resource-intensive and time-consuming, resulting in smaller and less representative training samples for toxicity detection. Consequently, models trained on manually annotated toxicity samples often exhibit low predictive accuracies [16]. In contrast, the existing lexicon-based approaches do not account for the wide linguistic variety on gaming platforms and are limited to explicit toxicity indicators neglecting implicit indicators and valence orientation of the text. Consequently, training samples annotated using existing valence lexicons lack generality and produce less accurate detection models.

This study proposes a novel framework for detecting implicit toxicity on gaming platforms by semantically enriching existing valence lexicons with word embeddings curated from the target platform. This approach generates representative and up-to-date training samples that capture domain-specific valence nuances. Leveraging these word embeddings enhances the model's ability to detect toxicity by capturing both semantic and lexical properties of the text, including variable valence orientations. Furthermore, curating embeddings from the target platform strengthens domain-specific knowledge, leading to more accurate detection.

For validation, the proposed valence embedding-based method is compared against several baseline methods, including TextBlob [17], valence aware dictionary and sentiment reasoner (VADER) [18], and extended VADER. We utilize a corpus of Twitch streaming chat logs accessible via Harvard Dataverse [19]. To ensure linguistic diversity, we curated a dataset consisting of 2,162 Twitch chat logs sourced from 42 streamers over a span of two months, encompassing approximately 7.5k comments. Additionally, agreement analysis is conducted by two human annotators to validate the reliability of the produced valence labels using the proposed valence embeddings-based annotation method various deep learning (DL) models, such as LSTM, GRU, and CN N, are trained on the training samples annotated using the proposed method, as well as the baseline methods, utilizing different language feature vector representations such as term frequency-inverse document frequency (TF-IDF), GloVe, and Word2Vec word embeddings. Performance enhancement metrics are calculated to demonstrate the improvement in performance achieved using the proposed annotation method.

The rest of this paper is structured as follows: Section two discusses and analyses the literature related to online toxicity detection. Section three introduces the proposed valence embeddings-based annotation method for online toxicity detection. Finally, section four discusses the evaluation experiments including the environment and experimental setup and results.

2 Related work

In this section, several research studies that addressed toxicity detection from text on gaming platforms are reviewed. A summary of the reviewed literature is presented in Table 1 and compared for the datasets used, annotation method, dataset content, classification model, and performance results.

Given the diverse variability of online toxicity orientations and the dominance of implicit valence markers on gaming platforms, some studies relied on manual annotation of training datasets to accurately capture the valance nuances. For instance, Vo et al. [16] manually annotated 17,561 comments from the League of Legends (LoL) gaming forum and 17,112 comments from the World of Warcraft (WoW) gaming forum. Surprisingly, only 207 comments from the LoL dataset (1.19% of the dataset) and 137 comments from the WoW dataset (0.81% of the dataset) were labeled as toxic. As a result, the annotated training datasets for toxicity detection turned out to be highly imbalanced with a minimum representation of toxic samples. Consequently, the prediction accuracies of toxic comments were very low with an F1-score ranging between 58.65% and 82.69% for LoL dataset and between 61.20 and 83.86% for WoW dataset. Despite being accurately labeled and publicly available for toxicity and toxicity detection research, the LoL and WoW datasets present less representative training samples given the minority of toxic samples in both datasets. Besides being time- and resource-consuming, manually annotated training datasets are subjective to the annotator's bias making them less generalizable.

On the other hand, the majority of research addressing toxicity detection on gaming platforms, focused on distant supervisions utilizing different types of lexicons or rules. For instance, Murnion et al. [5] focused on the World of Tanks (WOT) in-game chat and defined rules to classify in-game chat into eight broad classes (i.e., IsAbusive, IsPositive, IsNegative, HasBadLanguage, IsRacist, NoobRelated, SpecificTarget, FilteredText), each with a predefined toxicity score. Based on the content of in-game chat messages, each message may be assigned one or multiple classes, resulting in a final toxicity score that aggregates all related class scores. Murnion et al. employed a combined approach, utilizing manual and rule-based methods, to annotate 126,091 in-game chat messages extracted from WOT servers. Naïve

Table 1 Summary of studies addressing online toxicity detection on gaming platforms

Study	Dataset		Annotation method	Extracted Content		Toxicity detection method	Performance	
	Platform	Availability		Type of content	Count of hate samples			
Murnion et al. [5]	World of Tanks (WOT)	NA	Rule-based and Manual	In-game Chat	N/A	Microsoft Azure Sentiment Analyzer	NA	
Vo et al. [16]	League of Legends (LoL) World of Warcraft (WoW)	✓	Manual	Gaming Forum Comments	LoL: 207 (1.19%) WoW: 13 (0.81%)	Toxic-BERT Logistic Regression SVM Text-CNN GRU	oL	F1-score 58.65%—82.69%
							oW	F1-score 61.20—83.86%
Cornel et al. [20],	MMORPG (ragnarok)	NA	Scoring System with Explicit Bullying Terms	In-game chat	20,355	CNN	F1-score 99%	
Kobs et al. [7]	Twitch	✓	Lexicon-based (Emoji, VADER, and self-labeled emote) sentiments	Chats	404(25%)	Distribution based classifier Average based classifier CNN based classifier	F1-score 60.5%—62.7%	

automatic classification using SQL rules annotated 121,091 messages, while 5000 messages were manually annotated. These annotated messages were further analyzed using Microsoft Azure sentiment analysis. The researchers conducted several assessments to compare manual annotation with rule-based annotation, as well as rule-based annotation with Azure sentiment analysis. The findings confirmed the poor accuracy of toxic text classification. One limitation of this annotation approach is the subjectivity of the rules used to define toxicity cases, which rely solely on explicit linguistic markers within the content of in-game chat. In contrast, Kobs et al. [7] proposed the utilization of two unsupervised lexicon-based approaches, heavily reliant on the information encoded in Twitch emotes, alongside a weakly supervised neural network classifier trained on the outputs of these lexicon-based methods. This novel approach enhances the model's generalization toward out-of-vocabulary words through the integration of domain-specific embeddings. The augmentation of VADER lexicons with emotes and emojis has demonstrated efficacy in enhancing classification performance on Twitch comments. Despite providing more representative training samples, these methods yielded classification performance that is deemed insufficient. More recently, Cornel et al. [20], developed a learning model for toxicity detection in Filipino online games chat logs using a convolutional neural network (CNN). The researchers performed labeling using a system with previously defined toxic terms and annotated a total of 525,598 chat lines of the game Ragnarok chat's log. As a result, 20,355 lines were labeled as bullying and 499,718 phrases as not-bullying. CNN model yielded classification accuracies reaching 99%. Despite impressive prediction accuracies, the accuracy of labels through the proposed system was not validated in the study leading to uncertainty about the validity of the prediction results.

As presented in this literature review, existing approaches for toxicity detection on gaming platforms generally fail to capture the implicit valence indicators that are common on gaming platforms resulting in a low representation of the diverse valence orientations present on these platforms. Consequently, this study attempts to improve on the work presented by other researchers by extending valence lexicons with valence embeddings to allow for more comprehensive coverage of toxic language and nuances on gaming platforms.

3 Proposed framework

In this study, we propose a novel framework for implicit online toxicity detection. To address the lack of representative valence lexicons for online toxicity detection on gaming platforms, we propose a valence embeddings-based lexicon that captures not only the explicit valence indicators in

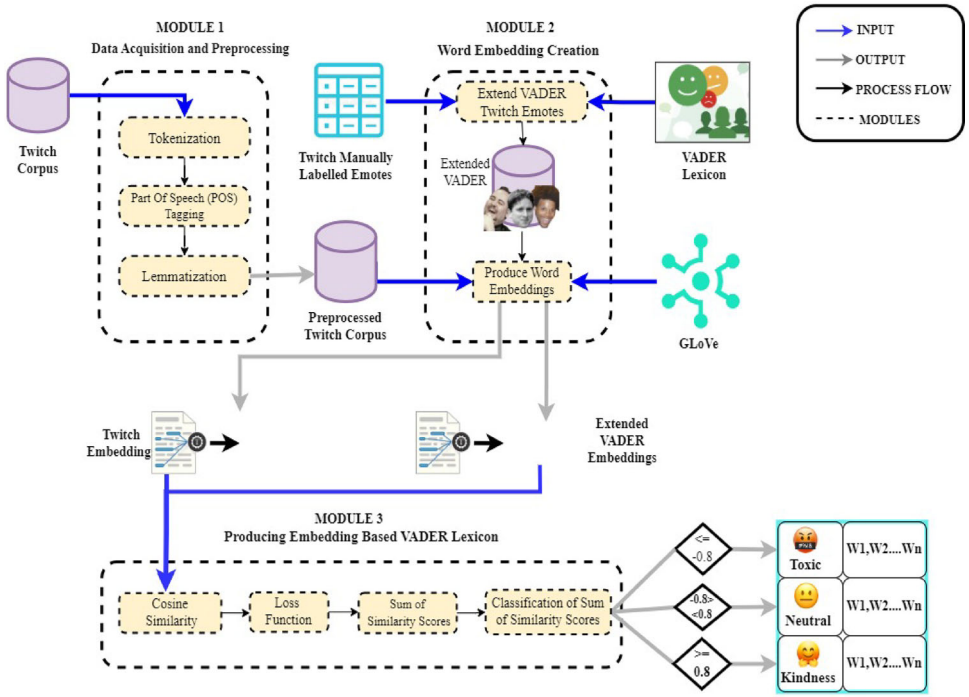

Fig. 1 The proposed embeddings-based lexicon for online toxicity detection

the text but also valence nuances and variable valence orientations. The proposed valence embeddings-based lexicon produces representative training samples for online toxicity detection with extensive coverage of dynamic valence. Figure 1 depicts the components of the proposed online toxicity detection framework. In subsequent sections, we explain in detail the workings of each module.

While Twitch was chosen as the emphasis due to its popularity in the gaming community and its distinctive linguistic traits, the suggested toxicity embeddings-based lexicon is easily adaptable to other platforms. This versatility results from the proposed framework's modular design, specifically in the embedding generation and annotation processes. By generating domain-specific embeddings and toxicity indicators for every platform, the proposed embeddings-based lexicon can successfully capture the complex nature of toxicity across several social media or gaming platforms. Discord and YouTube Gaming allow for informal and emoji-rich conversation. Platform-specific linguistic cues, such as Discord's custom emotes or YouTube's comment slang, might be incorporated into the embedding layer to increase the framework's application. Future research could build on this work by undertaking cross-platform tests to confirm the framework's robustness and performance in different scenarios.

3.1 Module 1: data acquisition and preprocessing

In this study, we focus on Twitch as a target platform; however, the proposed framework can be tailored to any other gaming platform. Twitch, boasting up to 30 million daily active users [14], stands as one of the most prominent gaming streaming platforms. For data acquisition of this study, we utilize a corpus of Twitch streaming chat logs available on Harvard Dataverse [18].

In this module, the raw Twitch chat corpus undergoes preprocessing using various natural language processing (NLP) techniques, including tokenization, part-of-speech (POS) tagging, and lemmatization. Given the corpus's informal nature—rich in emojis, special characters, emotes, and other linguistic markers—light processing is applied to retain features like capitalization, repetitions, and spelling errors, which may convey sentiments critical to toxicity detection. The preprocessing steps include: (1) tokenizing the text into manageable units, (2) applying POS tagging to identify word categories essential for lexicon-based valence analysis, and (3) lemmatizing words to their base forms. The processed data are then passed to the embedding creation module, Module 2.

3.2 Module 2: word embedding creation

In this module, we begin with the VADER lexicon [18], as a foundation. First, we extend this lexicon by incorporating emotes frequently used on Twitch, particularly those associated with toxic nuances. Later in Module 3 of this framework, we further enhance its semantic representation by integrating Twitch-specific valence embeddings. This two-step extension process aims to create a more comprehensive and representative lexicon for detecting online toxicity

on Twitch. Subsequently, we detail the methods for incorporating both emotes and embeddings into the VADER lexicon.

3.2.1 VEDAR and Extended VEDAR

VADER, a rule-based sentiment lexicon, assigns polarity scores to words, ranging from -1 (most violent) to 1 (most positive), based on predefined rules and context. VADER's scoring involves assessing word polarity and adjusting scores with sentiment intensity modifiers, including semantic orientation, punctuation, and word case [18]. The final compound score (f^v) aggregates these adjusted scores, as illustrated in Eq. 1.

$$f^v = \frac{\sum sentiment\ scores}{\left(\sum sentiment\ scores\right)^2 + \alpha}, \tag{1}$$

where $sentimentscores$ refer to the positive, violent, and neutral scores and α is the scale factor.

Twitch chat incorporates unique elements such as emotes and emojis, which can convey significant sentiment but also complicate toxicity detection. Emotes, often used in toxic contexts, add a layer of complexity. To address this, extended VADER incorporates a broader lexicon, including Twitch-specific emotes and slang. This enhancement improves the handling of contextual valence shifts and captures dynamic meanings of emotes, which may vary depending on their use [21]. For example, emotes like 'kappa' and 'TriHard' can indicate sarcasm or racial discrimination, affecting the sentiment analysis. For instance, the emote 'kappa,' which portrays the face of one of Twitch's employees, is utilized in sarcasm. The face emotes of famous streamer 'TriHard' are most commonly used to allude to Black people.

In this study, we extend the VADER lexicon by integrating emotes commonly used on Twitch, such as 'kappa,' 'TriHard,' and 'PogChamp,' into the sentiment analysis framework. Table 2 provides examples of these emotes, their names, and their meanings. Each emote is replaced by its name, and valence scores are assigned and aggregated, resulting in a final compound score (f^v) ranging from -4 (violent) to 4 (positive). The score is normalized and thresholded for final classification, with thresholds determined experimentally.

3.2.2 Twitch-specific valence word embeddings

Word embeddings, which represent words as low-dimensional vectors, capture semantic and lexical properties, including implicit valence. GloV [22] provides pre-trained embeddings using co-occurrence statistics. In this module, we use pre-trained GloVe 6B embeddings to generate word embeddings of the extended VADER lexicon as well as of the preprocessed Twitch corpus. These embeddings, generated for both preprocessed Twitch corpus and the extended VADER lexicon, are utilized in Module 3 to produce the embeddings-based valence Lexicon.

3.3 Module 3: producing Twitch-specific embedding-based valence lexicon

In this module, the embeddings generated from the previous module are matched using the cosine similarity (S_C) as depicted in Eq. 2. Cosine similarity calculates the inner product space of both embedding vectors, which produces a score that ranges from -1 to 1, where -1 indicates a negative correlation, 1 indicates a positive correlation, and 0 indicates no correlation [23]. In the context of embedding similarity, words with nearly similar meanings produce word space vectors that are more parallel, with a cosine similarity near 1.

$$S_C(x,\ y) = \frac{x \cdot y}{||x||||y||} \tag{2}$$

A loss function $loss_v(t)$ is then implied to assign a higher score to the context's valence label rather than the other possible labels. Equation 3 describes the objective function of the proposed extended valence lexicon embeddings. This implementation is based on the ternary sentiment-specific word embeddings and adapted to valence analysis instead. The objective is to assign a higher score to the context's valence label rather than the other possible labels.

$$\begin{aligned} loss_v(t) = &\max\left(0,\ m - f_c^v(t) + f_{i1}^v(t)\right) \\ &+ \max\left(0,\ m - f_c^v(t) + f_{i2}^v(t)\right), \end{aligned} \tag{3}$$

where t is the context window, m is the margin, f_c^v is the VADER valence score of the currently labeled word, and $f_{i1}^v(t)$ and $f_{i2}^v(t)$ are the valence scores of the other two classes.

The similarity scores are then summed up and thresholded to produce the negative (toxicity), neutral, and positive (kindness) annotation labels. The summation of similarity scores ($\sum$S) is then thresholded to produce a final classification. Similar to the lexicon-based annotation methods, the thresholds are set experimentally and validated on random samples from the dataset.

The algorithmic steps to produce Twitch-specfic embeddings-based valence lexicon are illustrated in Fig. 2. The proposed algorithm consists of four sub-procedures, each covering a vital step in the lexicon development process. The algorithm receives inputs from the raw Twitch corpus (T_c), the VEDAR lexicon, Twitch-specific emotes, and GloVe embeddings. The algorithm's output is the Twitch-specific-embeddings-based toxicity lexicon, which can recognize poisonous, neutral, and kind messages using contextual and semantic analysis.

Table 2 Sample Twitch emotes and their meanings

Emote visual representation	Emote name	Meaning
	Kappa	Used to express an eye-roll or relay sarcasm as a response to a streamer
	TriHard	Initially intended to express hype or excitement. However, it is often used for racial discrimination
	LUL	Stands for "Lame Uncomfortable Laugh", but is used for expressing laughter, even "Laughing Out Loud" at times
	PogChamp	Expressing excitement or surprise, mostly by viewers
	MonkaS	Used to express anxiety or stress by viewers or streamers undergoing intense or stressful gameplay
	Jebaited	Used when gamers are baiting for tips of information, and can also be used when a streamer is trolled into an obvious joke
	5Head	Expressing that either someone is intelligent or is pretending to be intelligent. It can be used for acknowledging a highly skilled play or mocking a poor one

Algorithm 1 Generate Twitch-Specific Toxicity Lexicon Using Embeddings

Require: Twitch Corpus T_c, VEDAR, Twitch Emotes, GloVe
Ensure: Twitch-Specific-Embeddings-Based Toxicity Lexicon
procedure GENERATETWITCHSPECIFICTOXICITYLEXICONUSINGEMBEDDINGS(T_c, VEDAR, Twitch Emotes, GloVe)

▷ Sub Procedure 1: Twitch Corpus Processing

Input: Twitch Raw Corpus T_c
Clean and preprocess (Lemmatize, POS Tagging, Lemmatize) T_c
Output: Clean Corpus C_c

▷ Sub Procedure 2: VEDAR Extension

Input: VEDAR, Manually Annotated Emotes, and GloVe
Extend VEDAR using inputs
Output: Extended VEDAR Embeddings V_{emb}

▷ Sub Procedure 3: Twitch Embeddings

Input: Clean Corpus C_c and GloVe
Generate embeddings for Twitch
Output: Twitch Embeddings T_{emb}

▷ Sub Procedure 4: Twitch-Specific Embedding-Based Lexicon

Input: Extended VEDAR Embeddings V_{emb} , Twitch Embeddings T_{emb}
Generate similarity scores for embeddings
Sum up similarity scores for classification
if similarity score ≤ -0.8 **then**
Classify as Toxic
else if $-0.8 <$ similarity score < 0.8 **then**
Classify as Neutral
else
Classify as Kindness
end if
Output: Twitch-Specific Toxicity Lexicon
end procedure

Fig. 2 Pseudocode of the proposed framework

The first sub-procedure, *Twitch Corpus Processing*, preprocesses the raw Twitch corpus (T_c) to produce a clean version (C_c). The preprocessing includes tokenization, part-of-speech (POS) tagging, and lemmatization. Next, in *VEDAR Extension*, the algorithm combines the existing VEDAR lexicon with manually annotated emotes and GloVe embeddings to generate extended VEDAR embeddings (V_{emb}). The third sub-procedure, *Twitch Embeddings*, uses the clean Twitch corpus (C_c) and GloVe embeddings to generate platform-specific word embeddings (T_{emb}) capturing the unique linguistic features of Twitch. Finally, in *Twitch-Specific Embedding-Based Lexicon*, similarity scores between V_{emb} and T_{emb} are computed to enrich the lexicon with Twitch-specific linguistic indicators. Content with a similarity score below -0.8 is labeled as toxic, between -0.8 and 0.80 as neutral, and above 0.8 as kind, resulting in the final Twitch-specific embeddings-based valence lexicon.

4 Experiments and results

This section presents the experimental settings used to validate the proposed framework and discuss the results.

4.1 Dataset

A dataset of 2162 Twitch chat logs by 42 streamers was collected for a duration of two months [19]. An approximation of 7.5k comments was analyzed. For each of the comments, emotes are replaced by their corresponding names, i.e., kappa, LUL, TriHard, etc. Upon annotation with the four methods, the distribution of negative, neutral, and positive labels varied based on the lexicons. Table 3 shows the distribution of labels for each of the annotation methods.

Table 3 Sample distribution of the annotation methods

Annotation method	Negative	Neutral	Positive
TextBlob	1379	4065	2064
VADER	1146	5122	1240
Extended VADER	1443	3853	2212
Valene embeddings-based Lexicon	1037	653	5817

4.2 Baseline

We ran the Twitch corpus through the three baseline lexicons: TextBlob, VADER, and extended VADER.

As presented in the literature review section, manual annotation has several disadvantages in general and particularly in the context of online toxicity detection. Therefore, distant supervision using valence lexicons is a commonly used approach in this field [7, 24, 25]. We ran the Twitch corpus through the three baseline lexicon-based annotation methods, namely TextBlob, VADER, and extended VADER.

In TextBlob [17], polarity and subjectivity are used for sentiment analysis. Polarity is a measure of the sentiment expressed, which determines how positive, negative, or neutral the chat message is. Polarity (**P**) values range from -1 to 1, where -1 indicates strong negativity, 1 indicates strong positivity, and 0 indicates neutrality. In this work, we rather utilize TextBlob for valence analysis; where negative ($P < 0$) refers to an expression of cyber toxicity, positive ($P > 0$) refers to an expression of kindness, and zero ($P = 0$) refers to a neutral expression. Polarity values are then thresholded to produce the annotation labels (i.e., toxicity, neutral, kindness). On the other hand, subjectivity is a measure of how subjective or objective an expression is. Subjectivity is expressed as a value between 0 and 1, where 0 indicates high objectivity, while 1 indicates high subjectivity. For our baseline annotation, we use the polarity annotator from TextBlob. Unlike TextBlob, VADER handles the explicit linguistic features of informal language, slang, and sarcasm better.

4.3 Feature representation

Feature representations are critically important for enhancing predictive performance. For validation purposes, we experiment with three feature representations, namely term frequency—inverse document frequency (TF-IDF), GloVe, and Word2Vec word embeddings.

TF-IDF [26] is a measure of the importance and commonality of terms in a particular corpus. For the TF-IDF representations, an inverse document frequency (IDF) dictionary is created containing the frequently occurring words and their IDF values. Then, the term frequency (TF) dictionary is formulated containing the TF values for the corresponding words in each document.

Word embeddings [27] are state-of-the-art feature extraction and representation methods. Unlike the statistical-based feature representation, word embeddings are more capable of capturing and encoding semantic relationships between words. Moreover, word embeddings handle out-of-vocabulary (OOV) words, which are not present in the corpus. In contrast, TF-IDF can only handle words in the corpus.

Both Word2Vec and GloVe embeddings are used in this study for context-rich, feature representation. While GloVe uses a co-occurrence matrix to capture the relationships between words, Word2Vec produces embeddings differently. Word2Vec [28] generates vector representations through a shallow neural network that learns the embeddings by predicting the probability of a word given its context. Table 4 summarizes the configurations of both word embedding models used in this study.

4.4 Classification models

Four DL classifiers are used for the evaluation, namely LSTM, GRU, and CNN. Each of these classifiers is trained on samples annotated using the three baseline methods and the proposed method explained in Sect. 3. We experiment with several architectural and training hyperparameters and evaluate the performance to reach the optimal configuration that produces the highest prediction results.

For the LSTM, we deploy a network configuration of one embedding layer followed by an LSTM layer with 128 units, and a dense layer with 3 units. Similarly, for the GRU model, we deploy a network configuration of one embedding layer followed by a GRU layer with 128 units and a dense layer with 3 units. However, we deploy a CNN with a network configuration of one embedding layer followed by a sequence of two convolutional layers with ReLU nonlinear activation and max pooling.

LSTM, GRU, and CNN models are trained for 80 epochs with a 0.01 learning rate and dropout rates of 0.5, 0.5, and 0.4, respectively.. All predictive models are trained using the categorical cross-entropy loss function and optimized with stochastic gradient descent (SGD) algorithm.

4.5 Experimental setup

To carry out the experiments, we utilized the Google Colab Pro + , which provided us with ample computational resources as follows:

- RAM: 52GB
- GPU: Nvidia V100

Table 4 Word embedding language model configurations for feature representation

Annotation language model	Hyperparameter			
	Embedding dimension	Window size	Number of filters	Vocabulary size
GloVe	50	NA	100	400,000
Word2Vec	500	3	NA	13,541

The Nvidia V100 GPU was employed to handle the computational demands of deep learning architectures and extended valence word embeddings, enabling efficient training and hyperparameter tuning.

The dataset was split into training and testing sets using an 80:20 ratio to evaluate the performance of the models. This data split was applied to all annotated datasets.

4.6 Manual validation of the proposed method using agreement analysis

To validate the reliability of the proposed valence embeddings-based lexicon annotation method, two human annotations conducted agreement analysis. The two human annotators assigned one of three labels to each sample based on its valence orientation as follows:

- Neutral (0): This label is assigned to samples that do not exhibit any valence. These samples do not convey any form of toxicityor kindness and are considered neutral.
- Positive (1): This label is assigned to samples that exhibit expressions of kindness. These samples contain words or phrases that express positivity, support, encouragement, or empathy.
- Negative (2): This label is assigned to samples that convey negative valence or engage in toxic behavior. These samples include offensive or abusive language, targeted insults, denigration, doxing, flaming, trolling, threats, impersonation, hate speech, or any form of harassment.

After manual annotation, Cohen's kappa agreement analysis is conducted, which considers both observed and chance agreements. Cohen's kappa coefficient (κ) is calculated as shown in Eq. 4.

$$\kappa = \frac{P_o - P_e}{1 - P_e}, \tag{4}$$

where P_o is the observed agreement and P_e is the expected agreement by chance.

The agreement analysis is conducted on the fully annotated dataset. Table 5 shows the contingency table of the two annotators' judgment. The annotations produce a kappa of 0.619 with 95% confidence and a standard error of 0.009. This result indicates a substantial agreement between both annotators.

4.7 Online toxicity detection results

The performance of the classifiers—LSTM, GRU, and CNN—was evaluated using accuracy, precision, recall, and F1-score, with the best results highlighted in Table 6. These metrics provide a comprehensive assessment of model performance:

- **Accuracy** measures the proportion of correctly predicted instances out of the total instances.
- **Precision** indicates the percentage of positive predictions that are actually correct.
- **Recall** quantifies the model's ability to identify all true positive instances.
- **F1-score** is the harmonic mean of precision and recall, offering a balanced measure when dealing with imbalanced datasets.

Experimental results revealed significant variation in online toxicity detection performance depending on whether the models were trained on baseline or proposed annotation methods. To quantify improvements, relative performance improvement was calculated using Eq. (4):

$$Performance\ Improvement = \frac{New\ Performance - Old\ Performance}{Old\ Performance} * 100\% \tag{4}$$

where *New Performance* indicates the prediction performance of the model on the training datasets generated using the proposed embedding-based lexicon. The *Old Performance* indicates the prediction performance of the model on training datasets generated using the three baseline annotation methods (e.g., TextBlob, VADER, Extended VADER).

The classifiers were tested with three feature representations: TF-IDF, Word2Vec, and GloVe, explained in Sect. 4.3. Training samples generated from baseline annotation methods were compared with those produced using the proposed embeddings-based annotation framework.

Table 5 Cohen's kappa agreement analysis contingency table

	Neutral	Positive	Negative	Total
Neutral	**4239**	47	311	4597
Positive	71	**71**	10	152
Negative	807	20	**1681**	2508
Total	5117	138	2002	7257

Table 6 Online toxicity detection results for LSTM, GRU, and CNN

Annotation method	Feature representation	Model											
		LSTM				GRU				CNN			
		Acc	Prc	Rec	F1	Acc	Prc	Rec	F1	Acc	Prc	Rec	F1
TextBlob	TF-IDF	0.42	0.47	0.43	0.41	0.60	0.62	0.58	0.61	0.66	0.69	0.64	0.67
	Word2Vec	0.70	0.74	0.70	0.71	0.75	0.75	0.75	0.75	0.71	0.71	0.71	0.68
	GloVe	0.45	0.43	0.45	0.43	0.63	0.65	0.63	0.63	0.77	0.78	0.77	0.77
VADER	TF-IDF	0.63	0.62	0.63	0.61	0.66	0.67	0.66	0.66	0.53	0.57	0.51	0.52
	Word2Vec	0.75	0.74	0.75	0.73	0.73	0.72	0.73	0.73	0.69	0.65	0.69	0.64
	GloVe	0.73	0.70	0.73	0.69	0.71	0.68	0.71	0.65	0.78	0.77	0.78	0.76
Extended VADER	TF-IDF	0.57	0.59	0.55	0.55	0.55	0.55	0.54	0.55	0.62	0.64	0.61	0.60
	Word2Vec	0.64	0.71	0.64	0.63	0.75	0.75	0.75	0.74	0.74	0.76	0.74	0.73
	GloVe	0.60	0.61	0.60	0.57	0.60	0.61	0.60	0.56	0.70	0.70	0.70	0.69
The proposed valence word embeddings lexicon	TF-IDF	0.79	0.81	0.78	0.80	0.77	0.79	0.78	0.77	0.83	0.84	0.84	0.84
	Word2Vec	0.84	0.84	0.84	0.84	**0.86**	**0.85**	**0.86**	**0.85**	0.86	0.86	0.86	0.86
	GloVe	**0.88**	**0.87**	**0.88**	**0.87**	**0.86**	**0.85**	**0.86**	**0.85**	**0.90**	**0.89**	**0.90**	**0.89**

* (Acc): accuracy, (Prc): precision, (Rec): recall, (F1): F1-score

The results in Table 6 highlight the substantial performance improvements achieved by the proposed valence word embeddings lexicon, particularly when paired with GloVe as the feature representation. Across all models—LSTM, GRU, and CNN—the proposed method consistently outperformed baseline approaches, including TextBlob and VADER. Baseline F1-scores for these methods, across all feature representations (i.e.,TF-IDF, Word2Vec, and GloVe), ranged between 41 and 77%, demonstrating their limitations in effectively capturing nuanced and implicit toxicity markers commonly found in gaming discourse.

In contrast, the proposed framework achieved superior performance across all models and metrics. CNN, the best-performing model, achieved 90% accuracy, 89% precision, 90% recall, and 89% F1-score with GloVe, marking a substantial improvement over the highest baseline results for CNN, which included 77% accuracy and 77% F1-score with VADER and GloVe. Similarly, GRU recorded 86% accuracy, 85% precision, 86% recall, and 85% F1-score with GloVe, while LSTM achieved 88% accuracy, 87% precision, 88% recall, and 87% F1-score, further emphasizing the robustness of the proposed framework across models. These results indicate consistent gains across all models, with F1-score improvements ranging from 12 to 48% compared to baseline methods.

The choice of feature representation significantly influenced performance. GloVe consistently outperformed TF-IDF and Word2Vec across all methods. For example, with the proposed framework, F1-scores using GloVe reached 89% (CNN), 85% (GRU), and 87% (LSTM), compared to 84%, 78%, and 80% with TF-IDF and 86%, 85%, and 84% with Word2Vec for the same models. These results highlight the superior capability of GloVe, combined with the proposed framework, to capture semantic and contextual nuances in textual data, making it particularly effective for detecting the implicit toxicity present in gaming discourse.

Baseline methods, including TextBlob, VADER, and extended VADER, demonstrated improved performance with advanced feature representations like Word2Vec and GloVe but still fell short of the results achieved when combined with the proposed framework. For example, TextBlob with GloVe achieved its best F1-scores of 45%, 63%, and 77% for LSTM, GRU, and CNN, respectively, while VADER with GloVe reached 69%, 65%, and 76% for the same models. Similarly, extended VADER with Word2Vec yielded F1-scores of 63%, 74%, and 73%, showing incremental improvements

over traditional configurations. However, even with these enhancements, the baseline methods failed to consistently capture the nuanced and implicit toxicity markers in gaming discourse. In comparison, the proposed framework, when paired with GloVe, achieved significantly higher F1-scores of 87%, 85%, and 89% for LSTM, GRU, and CNN, respectively. These results highlight that while advanced feature vectors enhance baseline methods, they are insufficient on their own without the semantic depth and contextual nuance introduced by the proposed valence word embeddings lexicon.

The proposed framework also demonstrated robustness in handling the challenges of minor class imbalance. High recall scores, ranging from 86% for GRU to 90% for CNN, show the framework's ability to effectively identify toxic instances, minimizing false negatives. Additionally, precision values (ranging from 85% for GRU to 89% for CNN) ensure reliability in predictions by reducing false positives. This balance across metrics validates the framework's suitability for real-world applications, where accurately detecting minority classes is critical for fairness and effectiveness.

5 Discussion

In contrast to the baseline methods, the proposed embeddings-based method has proven effective in producing more representative and semantically rich training samples in the context of toxicity detection. This is evident by the reported prediction accuracies in Table 6. GRU, LSTM, and CNN models achieved high prediction accuracies up to 86%, 88%, and 90%, respectively, when trained on samples annotated using the proposed method compared to the prediction accuracies achieved when trained on samples produced using the baseline methods. This improvement is observed across all types of feature vectors used (i.e., TF-IDF, Word2Vec, and Glove) with increased enhancement observed with Glove feature vectors. Moreover, the consistency of the evaluation metrics' scores (i.e., accuracy, precision, recall, and F1-score) indicates good fitting of the models to the training data. This further supports the generalizability of the proposed method.

The performance improvement is calculated using Eq. 4 based on the accuracy metric. For instance, considering LSTM model, to calculate the percentage of prediction enhancement achieved using the proposed method (i.e., *New Performance*) over TextBlob as the baseline (i.e., *Old Performance*) to produce training samples with TF-IDF as feature vector we calculate $(NewPerformance - OldPerformance)/(OldPerformance) * 100\%$ such that (0.79–0.42)/0.42*100 = 88.1%. Similarly, the performance improvement is calculated for all models considering all annotation methods and all feature vector representations. Table 7 shows that the proposed method achieves major performance enhancement across all models and feature vector representations in terms of prediction accuracy. For instance, for LSTM, accuracy enhancement ranges between 12% and 95.56%. For GRU, accuracy enhancement ranges between 14.67% and 43.33%. For CNN, accuracy enhancement ranges between 15.38% and 56.6%. Overall, the experimental results showcase the substantial performance improvement achieved by incorporating extended valence word embeddings across the tested DL models. On average, the proposed method yields a classification enhancement of 31.47% for LSTM, GRU, and CNN, which is a significant improvement. These findings demonstrate the importance of considering contextual valence in toxicity detection, which plays a pivotal role in combatting toxic content. The success of the proposed method across diverse DL models underscores its robustness and generalization, making it a promising approach for addressing the challenges in online toxicity detection.

GloVe embeddings outperformed all other representations due to its ability to recognize both global and local context in textual data. Unlike TF-IDF, which is based only on term frequency and lacks semantic comprehension, GloVe depicts words as dense vectors formed from co-occurrence patterns in a huge corpus, allowing it to encode meaningful word associations [22]. Additionally, Word2Vec predicts word probabilities based on local context windows, while GloVe integrates a global statistical perspective via a co-occurrence matrix, making it better suitable for capturing complex linguistic patterns. This capacity to integrate larger context is especially useful in casual, emoji-rich contexts like Twitch. These factors explain why GloVe consistently outperformed other feature representations, delivering higher accuracy, precision, recall, and F1-scores in the study.

5.1 Bias and diversity considerations

The proposed framework acknowledges the potential biases inherent in toxicity detection, particularly within culturally and linguistically diverse gaming communities. To mitigate such biases, the framework employs a domain-specific lexicon enriched with platform-relevant embeddings that capture diverse linguistic markers, including emotes and slang commonly used in gaming discourse. This approach ensures that the model recognizes implicit indicators of toxicity beyond explicit lexical features.

Additionally, the training data were curated from multiple Twitch streamers across varying contexts to enhance representativeness. This diverse dataset contributes to the framework's robustness in handling different linguistic nuances. The manual agreement analysis conducted as part of the validation process further reinforced the reliability of the

Table 7 Performance improvement achieved by the proposed method against the baseline method considering the model's accuracy metric

Baseline annotation method	Feature representation	DL models		
		LSTM (%)	GRU (%)	CNN (%)
TextBlob	TF-IDF	88.1	28.33	25.76
	Word2Vec	20	14.67	20.85
	Glove	95.56	36.51	16.88
VADER	TF-IDF	25.4	16.67	56.6
	Word2Vec	12	17.81	24.64
	Glove	20.55	20.85	15.38
Extended VADER	TF-IDF	38.6	40	33.87
	Word2Vec	31.25	14.67	16.22
	Glove	46.67	43.33	28.57

annotations, achieving a kappa score of 0.619, indicative of substantial alignment between annotators.

While the current implementation focuses on English-language data, future extensions of this work will explore multilingual toxicity lexicons to address linguistic and cultural variations comprehensively.

6 Conclusion and future directions

This study tackled the challenge of online toxicity detection in gaming communities, focusing on Twitch, by introducing a novel framework that extends valence lexicons with valence word embeddings. The proposed method achieved significant improvements, with an average performance enhancement of 31.47% across LSTM, GRU, and CNN models, and accuracy gains ranging from 12% to 95.56% (LSTM), 14.67% to 43.33% (GRU), and 15.38% to 56.6% (CNN) compared to rule-baed and sentiment-based baseline methods. Agreement analysis further validated the method's reliability, achieving a kappa score of 0.619.

The proposed methodology currently focuses on toxicity identification in English-language textual data. Future research could explore the inclusion of multilingual toxicity lexicons and embeddings to improve the framework's applicability across languages. Additionally, while the framework is tested on Twitch streaming data, its design is adaptable to any other platform with textual chatter content. Therefore, a possible future extension can consider testing the proposed framework on other gaming or social media platforms.

Overall, this research contributes to the ongoing efforts to combat toxicty in online gaming communities, offering insights and methodologies that can enhance user safety and well-being in digital environments.

Acknowledgements The authors would like to thank the CSIT students who took part in the manual annotation and agreement analysis work.

Author Contribution H.I. contributed to conceptualization, methodology, software, validation, formal analysis, investigation, resources, data curation, writing–original draft, writing–review and and editing, visualization, project administration, funding acquisition. A.K. contributed to resources, writing–review and editing, fund acquisition. A.J. contributed to writing–review and editing, visualization.

Funding This work is funded by ORSP.

Availability of data and material No datasets were generated or analyzed during the current study.

Declarations

Ethics approval and consent to participate NA.

Conflict of interest The authors declare no competing interests.

References

1. M. Kaiser, "Gaming Statistics – 2023," Truelist.
2. A.-D. League, "Free to Play? Hate, Harassment and Positive Social Experience in Online Games 2020," 2020.
3. N. A. Khan, S. N. Brohi, and N. Zaman, "Ten Deadly Cyber Security Threats Amid COVID-19 Pandemic," *TechRxiv Powered by IEEE*, no. May, pp. 1–6, Oct. 2020, https://doi.org/10.36227/TECHRXIV.12278792.V1.

4. Twitch, "Chat Tools," Twitch.
5. Murnion, S., Buchanan, W.J., Smales, A., Russell, G.: Machine learning and semantic analysis of in-game chat for cyberbullying. Comput. Secur. **76**, 197–213 (2018). https://doi.org/10.1016/j.cose.2018.02.016
6. J. O. Atoum, "Cyberbullying Detection through Sentiment Analysis," in *Proceedings - 2020 International Conference on Computational Science and Computational Intelligence, CSCI 2020*, Institute of Electrical and Electronics Engineers Inc., Dec. 2020, pp. 292–297. https://doi.org/10.1109/CSCI51800.2020.00056.
7. Kobs, K., et al.: Emote-controlled. ACM Trans. Soc. Comput. **3**(2), 1–34 (2020). https://doi.org/10.1145/3365523
8. Kim, J., Wohn, D.Y., Cha, M.: Understanding and identifying the use of emotes in toxic chat on Twitch. Online Soc. Networks Media **27**, 100180 (2022). https://doi.org/10.1016/j.osnem.2021.100180
9. Ismail, H., Serhani, M.A., Hussien, N., Elabyad, R., Navaz, A.: Public wellbeing analytics framework using social media chatter data. Soc. Netw. Anal. Min. **12**(1), 1–17 (2022). https://doi.org/10.1007/S13278-022-00987-5/FIGURES/10
10. G. K. P, A. A. V. S, J. P. V, A. Paul, and A. Nayyar, "A context-sensitive multi-tier deep learning framework for multimodal sentiment analysis," *Multimed. Tools Appl.*, vol. 83, no. 18, pp. 54249–54278, May 2024, https://doi.org/10.1007/S11042-023-17601-1/FIGURES/8.
11. Jothi, P., Antran, V.A.: A unified framework for analyzing textual context and intent in social media. ACM Trans. Intell. Syst. Technol. (2024). https://doi.org/10.1145/3682064
12. Ismail, H., Hussein, N., Elabyad, R., Abdelhalim, S., Elhadef, M.: Aspect-based classification of vaccine misinformation: a spatiotemporal analysis using Twitter chatter. BMC Public Health **23**(1), 1–14 (2023). https://doi.org/10.1186/S12889-023-16067-Y/FIGURES/9
13. Ismail, H.M., Belkhouche, B., Harous, S.: Framework for Personalized Content Recommendations to Support Informal Learning in Massively Diverse Information Wikis. IEEE Access **7**, 172752–172773 (2019). https://doi.org/10.1109/ACCESS.2019.2956284
14. J. P. V. and A. A. V. S., "A novel socio-pragmatic framework for sentiment analysis in Dravidian–English code-switched texts," *Knowledge-Based Syst.*, vol. 300, p. 112248, Sep. 2024, https://doi.org/10.1016/J.KNOSYS.2024.112248.
15. J. Wise, "Twitch statistics 2022: How many people use twitch?," Earthweb.
16. H. H. P. Vo, H. Trung Tran, and S. T. Luu, "Automatically Detecting Cyberbullying Comments on Online Game Forums," in *Proceedings - 2021 RIVF International Conference on Computing and Communication Technologies, RIVF 2021*, Institute of Electrical and Electronics Engineers Inc., Jun. 2021. https://doi.org/10.1109/RIVF51545.2021.9642116.
17. S. Loria, "TextBlob: Simplified Text Processing," TextBlob 0.16.0 documentation.
18. C. J. Hutto and E. Gilbert, "VADER: A parsimonious rule-based model for sentiment analysis of social media text," in *Proceedings of the 8th International Conference on Weblogs and Social Media, ICWSM 2014*, The AAAI Press, May 2014, pp. 216–225. https://doi.org/10.1609/icwsm.v8i1.14550.
19. J. Kim, "Twitch.tv Chat Log Data - Harvard Dataverse," Harvard Dataverse, V1.
20. J. A. Cornel *et al.*, "Cyberbullying Detection for Online Games Chat Logs using Deep Learning," in *2019 IEEE 11th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management, HNICEM 2019*, Institute of Electrical and Electronics Engineers Inc., Nov. 2019. https://doi.org/10.1109/HNICEM48295.2019.9072811.
21. L. Goodman, "75 Most Popular Twitch Emotes! - Meaning & Origin," 2023.
22. J. Pennington, R. Socher, and C. D. Manning, "GloVe: Global vectors for word representation," in *EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, Association for Computational Linguistics (ACL), 2014, pp. 1532–1543. https://doi.org/10.3115/v1/d14-1162.
23. J. Han, M. Kamber, and J. Pei, "Getting to Know Your Data," in *Data Mining*, Elsevier, 2012, pp. 39–82. https://doi.org/10.1016/b978-0-12-381479-1.00002-2.
24. Kini, P.M., Keni, A., Deepika, K.V., Divya, C.H.: Cyber-Bullying Detection using Machine Learning Algorithms. Int. Res. J. Eng. Technol. **8**(8), 1966–1972 (2020)
25. M. Fortunatus, P. Anthony, and S. Charters, "Combining textual features to detect cyberbullying in social media posts," in *Procedia Computer Science*, 2020, pp. 612–621. https://doi.org/10.1016/j.procs.2020.08.063.
26. Y. Hamdaoui, "TF(Term Frequency)-IDF(Inverse Document Frequency) from scratch in python . | by Yassine Hamdaoui | Towards Data Science."
27. R. Khandelwal, "Word Embeddings for NLP. Understanding word embeddings and their… | by Renu Khandelwal | Towards Data Science," Towards Data Science.
28. TensorFlow, "word2vec | TensorFlow Core," www.tensorflow.org.