Information Systems Frontiers
https://doi.org/10.1007/s10796-023-10446-x

Comparing Machine Learning and Deep Learning Techniques for Text Analytics: Detecting the Severity of Hate Comments Online

Alaa Marshan[1] · Farah Nasreen Mohamed Nizar[2] · Athina Ioannou[3] · Konstantina Spanaki[4]

Accepted: 1 November 2023

Abstract
Social media platforms have become an increasingly popular tool for individuals to share their thoughts and opinions with other people. However, very often people tend to misuse social media posting abusive comments. Abusive and harassing behaviours can have adverse effects on people's lives. This study takes a novel approach to combat harassment in online platforms by detecting the severity of abusive comments, that has not been investigated before. The study compares the performance of machine learning models such as Naïve Bayes, Random Forest, and Support Vector Machine, with deep learning models such as Convolutional Neural Network (CNN) and Bi-directional Long Short-Term Memory (Bi-LSTM). Moreover, in this work we investigate the effect of text pre-processing on the performance of the machine and deep learning models, the feature set for the abusive comments was made using unigrams and bigrams for the machine learning models and word embeddings for the deep learning models. The comparison of the models' performances showed that the Random Forest with bigrams achieved the best overall performance with an accuracy of (0.94), a precision of (0.91), a recall of (0.94), and an F1 score of (0.92). The study develops an efficient model to detect severity of abusive language in online platforms, offering important implications both to theory and practice.

Keywords Machine learning · Deep learning · Hate speech · Social media · Text pre-processing · Text representation · Text analytics

1 Introduction

The introduction of social media has significantly affected people's lives, allowing them to publicly share their opinions and beliefs about various issues spanning areas such as politics, economics, health and social issues (Meske & Bunde, 2022). As a result, social media has been deemed as one of the most prominent contributors to the freedom of speech principle (Putri et al., 2020). However, recently freedom of speech has been abused on many occasions, bringing negative consequences both to the individuals themselves as well as others who are being abused (Putri et al., 2020). Formally defined as online harassment, online abuse refers to verbal and/or graphical abuse towards others on an online platform (Karatsalos & Panagiotakis, 2020). Online harassment can adversely affect people's lives as people being targeted by others feel physical and mental suffering (Modha et al., 2020a).

According to Matamoros-Fernández and Farkas, (2021, p. 205), 'sociality is continuously transformed by the interplay of humans and technology'. As such, social media platforms, although characterised as communication infrastructures that are quite open and decentralised, where users can widely share their opinions, participate, and develop new networks (e.g., activism); they have also amplified several forms of abuse, such as digital hate speech, racism, and online discrimination (Kim et al., 2022; Matamoros-Fernández & Farkas, 2021). The spread of digital hate speech through social media has evidently contributed to the reshaping of "racist dynamics through their affordances, policies, algorithms and corporate decisions" (Matamoros-Fernández & Farkas, 2021, 206).

✉ Alaa Marshan
a.marshan@surrey.ac.uk

1 Department of Computer Science, University of Surrey, Guildford, UK

2 Department of Computer Science, Brunel University, London, UK

3 Surrey Business School, University of Surrey, Guildford, UK

4 Audencia Business School, Nantes, France

Published online: 24 November 2023

Evidence from previous studies demonstrates the widespread reshaping of structural oppression across several social media platforms, such as Facebook, Instagram, TikTok and YouTube (Matamoros-Fernández & Farkas, 2021). Likewise, one of the most popular social media platforms that exhibits the majority of cases of online harassment is Twitter (Oriola & Kotze, 2020). There has been a rise in the propagation of harassment on Twitter under the guise of freedom of speech, even though Twitter's terms of service prohibit such behaviours (Oriola & Kotze, 2020). While Twitter does have a mechanism to detect tweets that hinder harassment towards an individual, such harassment detection tools fail to detect such contexts due to the sheer volume of data that is being generated (Modha et al., 2020).

Identifying the prevalence of online abuse as well as being able to measure and detect such phenomena is crucial in understanding the negative sheer influence of social media platforms in various aspects of society; causing harm on individuals, exacerbating social divisions, and eroding trust among people (Vidgen et al., 2019). Social media addiction, exhaustion and fake news sharing constitute only some of the negative user experiences in using social media platforms (Jabeen et al., 2023). Such adverse, and often detrimental, consequences of social media platforms contribute to growing body of research focusing on known as the *dark sides of social media* (DoSM) (Jabeen et al., 2023). Scholars in the IS field have become increasingly concerned with the investigation of the adverse effects of social media, and more specifically with online hate speech phenomena; the latter being an area that has been characterised as quite vital and challenging to research as well as significantly understudied (Kim et al., 2022; Matamoros-fernández, 2021; Tontodimamma et al., 2021). Addressing these calls for further research, the present study contributes to the IS discipline by investigating the *dark side of social media* and more specifically the spread of such forms of abuse (i.e., online hate speech), offering concrete solutions in detecting such phenomena ultimately aiming to counteract online abusive behaviours.

Recently, several studies have focused on developing techniques towards detecting rumours (Singh et al., 2022), abusive comments and hate speech online, classifying such comments into predefined categories such as racism, sexism, or toxic comments (MacAvaney et al., 2019). Most of the existing research has focused on comparing the performance of various Machine Learning algorithms and their ability to detect such comments (Muneer & Fati, 2020; Talpur & O'Sullivan, 2020). While other studies have conducted investigations comparing the different Deep Learning algorithms used to identify and classify abusive comments (Chen et al., 2018; Lee et al., 2019; Lynn et al., 2019). Surprisingly, there is scarcity of research focusing on detecting the severity of comments causing online abuse in online platforms. Moreover, the examination of the impact of text pre-processing on the performance of the developed models has not yet been investigated within the context of hate speech severity detection (Alam & Yao, 2019; Keerthi Kumar & Harish, 2018). Intending to address these challenges, the present study aims to answer the following research question:

> What are the most effective machine and/or deep learning algorithms to successfully detect the severity of abusive comments online?

In answering the above, this paper is structured as follows. First, a theoretical background of the work related to hate speech detection and the different approaches and algorithms used to achieve this is discussed. The following section clarifies the research methodology, namely CRoss Industry Standard Process for Data mining (CRISP-DM), that is followed to pre-process the data and develop and validate the different machine learning and deep learning models used to detect the severity of hate speech. Third, the results of each developed model are discussed in detail and compared to each other. Finally, the conclusion section concludes this work and offers some suggestions for future research.

2 Theoretical Background

2.1 Hate Speech in Online Platforms

The growth of Internet 2.0 has had a significant impact on people's lives (Muneer & Fati, 2020). Social media facilitate information exchange, allowing users to consume and share information with others (Kiilu et al., 2018), where users share their opinions online, popular and unpopular ones, vastly increasing the volume of user-generated content (Lynn et al., 2019). This high volume of content contributes to uncontrolled conversations on social media as there are too many users (Ibrohim & Budi, 2018). The increase in the use of social media has introduced a dark side as well, resulting in the misuse of social media platforms (Novalita et al., 2019). Evidence suggests that social media can nurture heated discussions, which often can result in the use of offensive and insulting language thus manifesting into abusive behaviours (Tontodimamma et al., 2021). According to (Castaño-Pulgarín et al., 2021), hate speech can be defined as "[…] the use of violent, aggressive or offensive language, focused on a specific group of people who share a common property, which can be religion, race, gender or sex or political affiliation through the use of Internet and Social Networks […]" (pg. 1). Moreover, (MacAvaney et al., 2019) highlights the difficulties related to subtle language

differences between different types of hate speech such as hate, insult and threat.

Past research argues that there is a wide range of available tools to deal with abusive behaviours in platforms including Twitter and Facebook (Au et al., 2021); such as comment moderation tools on Facebook. These mechanisms, however, prove to be inadequate to manage online abuse (Founta et al., 2019). In their study, (Nascimento et al., 2022) argue for the importance of investigating the systems' robustness to deal with biases towards identity terms including gender, race and religion. The widespread abuse in social media causes significant emotional trauma and damage, desensitizing individuals (Ibrohim & Budi, 2018; Kim et al., 2022). Furthermore, online abusive comments can also affect the potential sales of companies' services. For example, in 2013, Facebook faced severe criticism because it hosted pages that promoted hate against women. As a result, many Facebook users along with large corporations threatened to remove their ads from the platform (Nobata et al., 2016). Another instance has been Pepsi, Walmart, and AT&T removing their ads from YouTube after the ads were played within videos promoting extremist views (Lynn et al., 2019).

Apart from social media, Internet 2.0 has also changed news delivery, allowing users to read news online through websites or news apps (Desrul & Romadhony, 2019). One of the most significant changes brought by Internet 2.0 to the digital realm is the comment feature. The comment feature allows for interaction between the author and the reader. In their study, (Desrul & Romadhony, 2019) argue that the primary purpose of a comment section in online platforms is to allow readers to share their comments and provide constructive criticism. However, not all comments that are posted are constructive. Very often, there are comments expressed in abusive, intimidating, and hateful speech. Online abuse makes individuals feel embarrassed, humiliated, and even insulted by anonymous users on the internet (Kim et al., 2022; Talpur & O'Sullivan, 2020). The abuse online in the comment sections of news platforms can result in negative feedback from other users as well. As a result, news websites can lose traffic coming to their website over time (Awal et al., 2018). However, there are initiatives to tackle these issues. For instance, pre-published moderation is carried out by human moderators for BBC news online. Human moderation of all content is a costly and time-consuming process. In their study, (Chen et al., 2018) highlight that human moderation lacks scalability because of the sheer volume of user-generated content produced daily. Facebook and YouTube deploy simple word filters to moderate abusive content online (Chen et al., 2018), however such as word filters are only capable of moderating semantic abuse.

An alternative to pre-published moderation would be post-published moderation which depends on crowdsourcing mechanisms such as reporting systems used on Twitter or moderators' decisions on platforms such as Reddit. The disadvantage of post-published moderation is that the damage has already occurred as the comments have been published and already publicly available to a public audience (Chen et al., 2018). Overall, correctly identifying abusive speech is a critical issue and priority both for large social media companies and every company that allows user-generated content (Gambäck & Sikdar, 2017). Thus, it is increasingly essential that a mechanism is developed able to assess the severity of abusive content online automatically to tackle abuse in online platforms.

2.2 Traditional Machine Learning Methods

Past research has adopted a wide variety of machine learning methods to detect hate speech in online platforms (Singh et al., 2022; Tontodimamma et al., 2021). Considering the task of hate speech detection, supervised learning is most used type of machine learning (Malmasi & Zampieri, 2017; Meske & Bunde, 2022). Supervised learning algorithms include Support Victor Machine (SVM), Naïve Bayes (NB), K-Nearest Neighbour (KNN), Random Forest (RF), and Logistic Regression (LR). The use of these algorithms in the field of hate speech detection is covered in the following sub-section.

2.2.1 Support Vector Machine (SVM)

Previous studies have shown SVM to be very effective for abuse detection online while other studies found contradictory results. More specifically, in their study investigating cyberbullying, a specific form of abuse and harassment in online platforms, (Muneer & Fati, 2020) found that SVM was less effective in detecting abusive content on Twitter when the dataset was large, comparing to other ML methods; while SVM performed better when the dataset was smaller. The study found that SVM achieved the lowest accuracy and precision, however, it had the best recall compared to the other classifiers. Furthermore, in their study (Chen et al., 2018) used n-grams, a sequence of n-words in a sentence that captures the context in which the words are used together, and word vectors based on word embeddings, to detect abusive content using SVM. The study found that SVM performed poorly when the original class distribution was more balanced, while once oversampling techniques were applied, the SVM performance improved drastically. Other studies found similar results, deploying SVM to detect abusive text using CountVectorizer and tfidVectorizer for feature extraction showing that SVM performed very poorly before oversampling, but the performance drastically increased after oversampling (Eshan & Hasan, 2017).

Evidence suggests that SVM classifiers perform well when trained on n-grams. Previous studies have shown that using

the SVM poly kernel classifier with an n-gram of 5 yields the best accuracy in classifying abusive text (Noviantho Isa & Ashianti, 2017). The high performance of the SVM classifier can be attributed to the data being used in the study being non-linear separable, implying that a linear line cannot separate the data within the dataset. Other studies have also used Linear SVM and training it on different features: character n-grams, skip-grams, and word n-grams (Malmasi & Zampieri, 2017). Skip-gram models can predict the context word for the given target word (Mikolov et al., 2013) while character n-grams breakdown a word at the character level and the word n-grams breakdown sentences at the word level (Lecluze et al., 2013). Evidence suggests that the character 4-g model with the linear SVM classifier achieves the best performance (Malmasi & Zampieri, 2017).

2.2.2 Naïve Bayes

Past research has deployed Naïve Bayes classifiers aiming to detect abusive comments online. In their study, (Ibrohim & Budi, 2018) used Naïve Bayes, Random Forest and SVM as well word n-grams and character n-grams. The results of the study showed that the Naïve Bayes classifier produced the best results with the word unigram and word bigram, while any combination of a word unigram and a word n-gram yielded better results. Moreover, (Kiilu et al., 2018) used n-grams with the Naïve Bayes classifier aiming to detect hate tweets, showing that the detection of hate tweets was most effective when using bigrams. While in their study, (Awal et al., 2018) used a different approach to detect abusive comments in discussion threads within tweets. To train the classifier, they use tokenization to split the text into smaller tokens prepared into a bag-of-words (BoW) vector. Using this method resulted in a relatively high accuracy for detecting abusive tweets. Other studies have also used the BoW method to train the Naïve Bayes classifier (Özel et al., 2017). Compared to other methods, such as SVM, decision trees, and KNearest Neighbours (kNN), the Naïve Bayes classifier achieved the best accuracy in detecting abusive text (Özel et al., 2017). Also, feature selection using Mutual Information (MI), measuring statistical independence between random variables, has been implemented attempting to detect hate comments online (Desrul & Romadhony, 2019). Results showed that the Naïve Bayes classifier with MI performed relatively poorly as the attributes in MI do not have variations in every class; thus, the Naïve Bayes classifier was not able to detect the abusive text accurately (Desrul & Romadhony, 2019).

2.2.3 K-Nearest Neighbour, Random Forest, and Logistic Regression

Past research has investigated online abusive speech, by comparing the performance of several techniques. More specifically, the performance of Logistic Regression, Random Forest, SVM, Naïve Bayes, Decision Trees, and KNN in the detection of toxic comments has been empirically demonstrated; measuring the performance of the models by calculating the loss with specific loss metrics, namely, Hamming Loss and Log Loss (Rahul et al., 2020). The Hamming Loss is the fraction of the wrong labels to the total number of labels (Wu & Zhu, 2020), and Log Loss indicates how close the probability of the predicted value is to the actual value (Rezaeinia et al., 2019). Results showed that Logistic Regression had the lowest Hamming Loss value highest accuracy among the two classifiers, deeming it as the best classifier to identify toxic comments in this scenario (Rahul et al., 2020). Similar machine learning classifiers have been adopted in other studies as well. (Talpur & O'Sullivan, 2020) implemented Naïve Bayes, kNN, Decision Trees, Random Forest, and SVM, developing a technique to combine Synthetic Minority Oversampling Technique (SMOTE) with the Pointwise Mutual Information (PMI) technique. Using this technique, the study showed that the Random Forest classifier was the best in detecting the severity of cyberbullying tweets (e.g., low, medium, high and non-cyber bullying) (Talpur & O'Sullivan, 2020). SMOTE has been also adopted to overcome the class imbalance while detecting online cyberbullying, demonstrating that the Random Forest classifier achieved the best results (Al-Garadi et al., 2016).

2.3 Deep Learning Methods

There is a small body of research adopting deep learning methods in the detection of hate speech in online digital platforms. (Chen et al., 2018) compared the performance of a Convolutional Neural Network (CNN) and a Recurrent Neural Network (RNN) to detect abusive content on social media websites. The models were trained on n-grams and word vectors. For the CNN model, the Rectified Linear Unit (ReLU) was used as the activation function, along with multiple filters where each feature had 100 feature maps. The RNN model was a one-layer bi-Long Short-Term Memory (Bi-LSTM) model. The model was used with the hidden layer's size set to 100 (Chen et al., 2018). Interestingly, the results showed that the models yield worse results when oversampling is used to deal with imbalanced class distribution. Oversampling limited the ability of the deep learning models to detect abusive content (Chen et al., 2018). (Pitsilis et al., 2018) also implemented an ensemble of LSTM-based classifiers to detect hate speech. The researchers found an improvement in the model's performance when an ensemble was used instead of a single classifier. It was also found that the ensemble schemes could outperform any NLP-based approach in abusive language detection.

(Lynn et al., 2019) implemented two bi-directional models with different recurrent layers: Bi-LSTM and Bi-Gated

Recurrent Unit (Bi-GRU) and compared their performance with the traditional machine learning models. The Bi-LSTM network consisted of three bidirectional layers containing 50 LSTM cells and used the tangent hyperbolic and hard sigmoid activation functions (Lynn et al., 2019). Similarly, the Bi-GRU network had the same structure, but there were only two bidirectional layers instead of three bidirectional layers. The results showed that the deep learning methods outperformed the traditional machine learning classifiers. (Lee et al., 2019) explored CNN, RNN, and their variant models using a pre-trained Global Vector (GloVe) representation for word-level features. GloVe is an unsupervised method for obtaining vector representations for word (Pennington et al., 2014). They additionally explored Latent Topic Clustering LTC), which extracts latent topic information from the hidden states of RNN, which is then used to classify the text data (Lee et al., 2019). The results found that RNN with LTC exhibited the best performance. However, the results were not too significant compared to the baseline models used (Naïve Bayes, Logistic Regression, SVM, Random Forests and Gradient Boosted Trees) and the attention-added model.

(Gambäck & Sikdar, 2017) used CNN to help classify hate speech into four categories: racism, sexism, racism and sexism, and non-hate speech. The neural network was trained on four different feature embeddings, namely, character 4-g, word vectors based on semantic information built using word2vec, randomly generated word vectors, and word vectors combined with n-grams to see which feature embedding could best help with a hate-speech classification system. The results showed that the model based on the word2vec embeddings achieved the best results in the classification. (Georgakopoulos et al., 2018) used word embedding to train the CNN model, which showed high accuracy in detecting toxic text. (Marwa et al., 2018) compared CNN, LSTM, and Bi-LSTM with the traditional machine learning methods. In their study, the word embedding models used for the neural networks were word2vec, GloVe, and Sentiment Specific Word Embeddings (SSWE). The study showed that the deep learning models did a better job detecting online harassment tweets. A similar study compared deep learning models with the traditional machine learning models by (Badjatiya et al., 2017). The deep learning models used were CNN and LSTM. The study had similar results to (Marwa et al., 2018), where the deep learning methods outperformed the more traditional methods in detecting hate speech.

Comparing the performance of CNN to LSTM, (Bashar et al., 2019) found that LSTM was sensitive to noise and therefore struggled to detect misogynistic tweets, thus concluding that CNN models are better at discovering larger patterns, as they are less affected by the noise and outperform LSTM. Additionally, (Park & Fung, 2017) used three CNN-based models to classify sexist and racist language. The CNN models used were CharCNN, WordCNN, and HybridCNN. The study compares one-step and two-step classification. The one-step approach is a multi-class classification for detecting sexist and racist language. The two-step approach combines two classifiers; one classifies racist language, and the other classifies racist and sexist comments (these comments were termed abusive comments). (Park & Fung, 2017) study shows that the two-step approach can potentially classify abusive language in large datasets correctly. For cyberbullying (abusive content) detection using CNN, (Al-Ajlan & Ykhlef, 2018) proposed the Optimised Twitter Cyberbullying Detection (henceforth. OCDD) approach. The OCDD approach eliminates the tiring task of feature selection/extraction. Instead, the OCDD approach replaced feature selection/extraction with word vectors that capture the semantics of words. While the OCDD approach has not been implemented for cyberbullying detection, (Al-Ajlan & Ykhlef, 2018) claim that it could yield significant results based on the OCDD's performance in other text mining contexts.

2.3.1 Traditional Machine Learning versus Deep Learning

Overall, it becomes apparent that the traditional machine learning methods, such as SVM and Naïve Bayes classifiers, perform well at detecting abusive text. Evidence suggests that such classifiers perform well when used with the n-gram feature extraction method. Also, Random Forest, KNN, and Logistic Regression models occasionally perform well, depending on the dataset used to train the model. In addition, Bi-LSTM and CNN models are popular deep learning techniques used in the detection of abusive text. Other popular techniques deployed to detect abusive speech include LSTM and deep learning algorithms with added attention layers. The deep learning models are trained using word embeddings, and the most popular word embeddings used are pre-trained word vectors such as GloVe or word2vec. While there is a growing body of studies focusing on the detection of abusive text in online platforms, there is a scarcity of research aiming to detect the severity of such abusive content. The implementation of techniques that can effectively detect the severity of hate speech online can help to further improve the filtering mechanisms in online platforms in the fight against online abusive behaviours and harassment.

3 Research Methodology

In the present study we have used a publicly available data set. The data was obtained from Kaggle's "Toxic Comment Classification Challenge" (Kaggle, 2021). It consists of 159,571 labelled comments classified into six categories: ‘toxic’, ‘severe toxic’, ‘obscene’, ‘threat’, ‘insult’, ‘identity hate’. Each category or label contains a binary value of 1

or 0; indicating whether the comment contains that specific category of harassment (1) or not (0).

In this study, Data Mining (DM) focused on exploring the available data and constructing algorithms that an organisation can apply to extract knowledge from a very large amount of data (Marbán et al., 2009). For this research, following the research conducted by (Marshan et al., 2021), the cross-industry process for data mining (CRISP-DM) methodology is adopted. CRISP-DM was introduced in 1999 and is a standard methodology for data mining applications in different domains such as healthcare, engineering, education, tourism, and warfare (Martinez-Plumed et al., 2019). This methodology is primarily divided into six steps: business understanding, data understanding, data preparation, modelling, evaluation and deployment, which are explained in the following sub-sections (Marbán et al., 2009). The R programming language was used to perform the technical steps of the CRISP-DM methodology.

4 Data Analysis and Results

4.1 Business and Data Understanding

The present study aimed to provide insights on the algorithms best deployed to classify the severity of abusive comments. This can further enhance the effectiveness of the existing harassment detection tools on different online platforms such as Twitter and Facebook.

The original dataset contains eight features (columns):

- **Id**: An alphanumeric value that uniquely identifies each observation in the dataset
- **comment_text:** The text data on which the hate speech detection will be run.
- **Six binary fields: toxic, severe_toxic, obscene, threat, insult, and identity_hate.** These fields contain 0's or 1's to indicate if the text contains any of the mentioned types of harassment (1) or not (0).

In addition to the existing variables, a new feature called **harassment_or_not** is created, which contains the sum of six harassment categories columns and reflects the severity of abuse in each comment. The values of the newly engineered feature, **harassment_or_not,** ranges from 0 to 6, where 0 indicates no abusive content in the text and 6 indicates the extremely abusive language in the comment. The column **harassment_or_not** was the primary target variable for the comparative study. Figure 1 shows sample observations in the toxic comment dataset.

For further exploration of the target variable, the **harassment_or_not** variable is converted to a factor (also known as a categorical variable) with seven categories ranging from 0 to 6. Once the data type is converted to factors, a histogram is plotted (see Fig. 2), which provides insight into the distribution of the classes.

The histogram shows that there is a class imbalance between the 7 classes. A proportion table (see Table 1) was created to understand the extent of the class imbalance further, which shows that around 89% of the text data falls under the class label 0. The other classes combined make up for the remaining 11%.

Next, to further explore the textual data stored in **comment_text**, a new variable called **TextLength** is generated, which stores the number of characters for every comment. The newly generated variable is used to overview the data within the **TextLength** column, see Table 2 for a summary statistics of the data: min value, 1st quartile, median, mean, 3rd quartile and the max value (Ross, 2021).

Interestingly, the result shows a massive jump in the text length between the third quartile and the maximum value. The gap indicates that the maximum value could be an outlier or that very few pieces of text have a very high text length (Kwak & Kim, 2017). In this case, the outlier may be the maximum value. The distribution of the length of the text is further explored by plotting a histogram to investigate relation between the number of comments and their length, see Fig. 3. The histogram shows that as the length of the text increases, the text count (number of comments) decreases. i.e., there is more significant proportion of short text compared to longer text, which can explain the skew between

id	comment_text	toxic	severe_toxic	obscene	threat	insult	identity_hate	harassment_or_not
0000997932d777bf	Why the edits made under my username Hardcore Metallica Fan	0	0	0	0	0	0	0
002264ea4d5f2887	recent appearence on the Tonight Show with Jay Leno? He looks	1	0	0	0	0	0	1
0036621e4c7e10b5	Would you both shut up, you don't run wikipedia, especially a stu	1	0	0	0	1	0	2
022509df20736807	I shit on your face ... fuck you	1	0	1	0	1	0	3
01166f26ee280e56	Gay ....The existence of CDVF is further proof that is a sad twat. He is also very ugly, and has a willy for a face.	1	0	1	0	1	1	4
057894cf4738a5d8	You are a gay homo. I hope you choke on your penis and die. I am	1	0	1	1	1	1	5
66f0a9006c188820	fuck you honkey, why you hatin' on blacks? You fucking pussy gee	1	1	1	1	1	1	6

Fig. 1 Sample observations in the toxic comment dataset

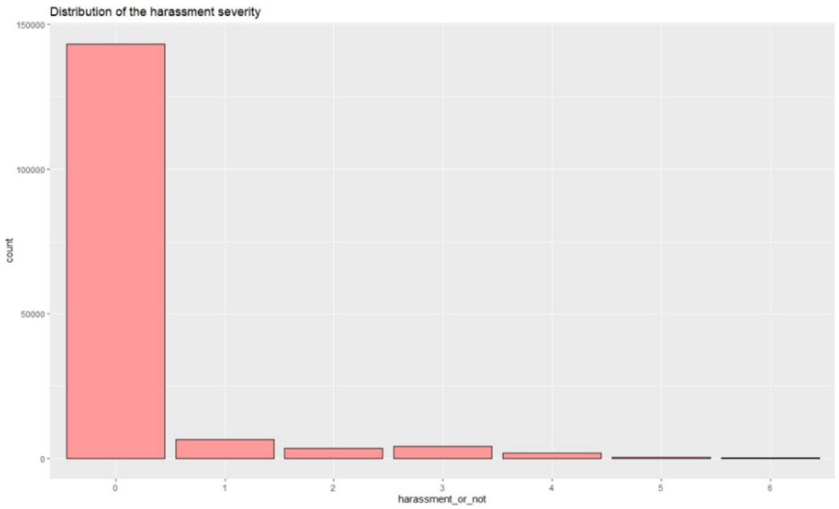

Fig. 2 Distribution of the harassment_or_not variable

Table 1 Distribution of the harassment_or_not variable

Class	0	1	2	3	4	5	6
Distribution	89.83	3.99	2.18	2.64	1.10	0.24	0.02

Table 2 Summary statistics of the TextLength variable

Min	1st Quartile	Median	Mean	3rd Quartile	Max
6.0	96.0	206.0	394.7	436.0	5895.0

the values in the third quartile of the text length and the maximum text length shown in Table 2.

Additionally, another histogram is plotted to visualize the distribution of the length of the text by class in **harassment_or_not** calculated variable, see Fig. 4, which shows similar distribution for each of the classes of harassment severity, i.e., a higher proportion of the texts have a shorter text length. This histogram is also very indicative of the class imbalance as indicated by the different colours.

Furthermore, a word cloud has been developed (see Fig. 5), a visual representation of the most frequently used words in the comments, after removing stop words, numbers, punctuation, words with more than two repeated letters, any single letters, and non-ASCII characters (DePaolo & Wilkinson, 2014); (Saif et al., 2014).

The resulting word cloud shows that most of the words within the text are non-abusive words. The word cloud results are somewhat expected, in accordance with the results from our analysis on Table 2, indicating that 89% of the textual data comprises non-abusive text.

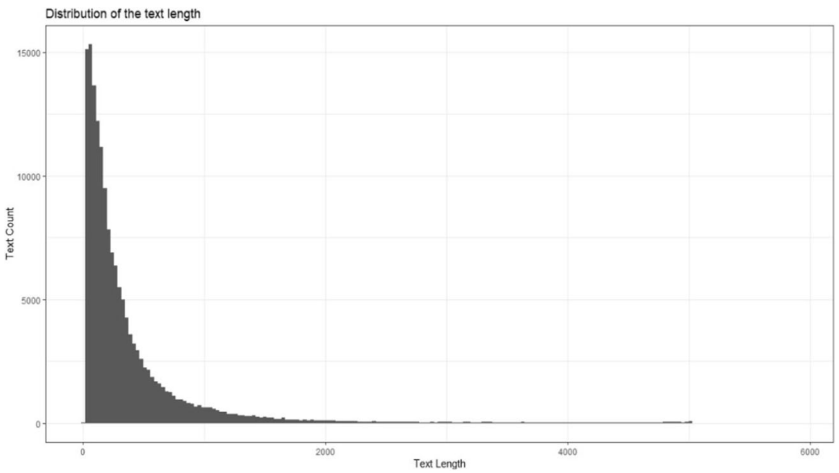

Fig. 3 The relation between Text Count and Text Length

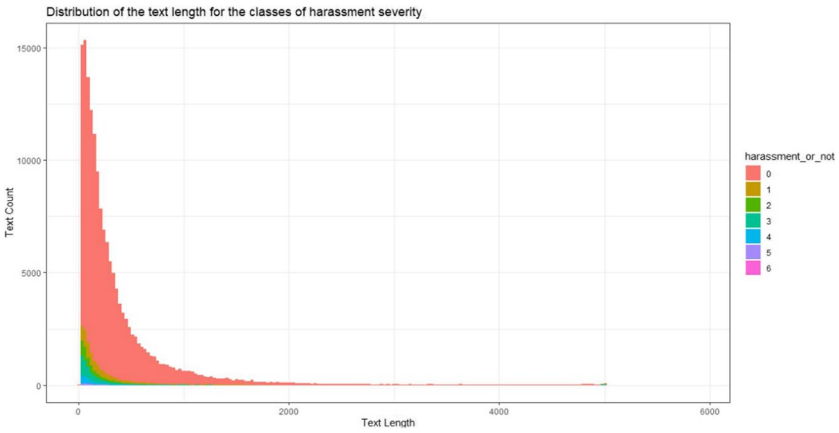

Fig. 4 Distribution of comments' lengths by harassment classes

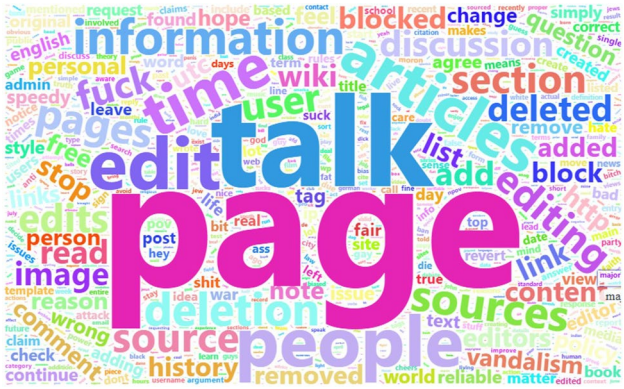

Fig. 5 Word cloud for most frequent words in the comments

4.2 Data Preparation

Two different approaches have been used to prepare the text data to detect the severity of abuse in the text (Georgakopoulos et al., 2018); one for the machine learning models: SVM, Naïve Bayes and Random Forest, and another method for the deep learning model Bi-LSTM and CNN.

4.3 Text Preparation for Machine Learning models

The text preparation for the machine learning models involved creating unigrams and bigrams. The lists of unigrams and bigrams were then filtered to remove stop words and non-words. Following, the count for each token (unigram or bigram) is summarised at a comment ID level, which resulted in extensive and sparse feature set because many of the tokens will have a majority of 0's across the many word features created. Next, the newly created unigrams and bigrams features are combined with the original dataset by joining them by using the comment ID as the key. Lastly, the modified dataset is split into training and testing datasets following a 70/30 split, where 70% of the data is used to train the model, and 30% of the data is used to test the model. In addition, SMOTE is used to handle the class imbalance discovered during the exploratory data analysis (EDA) performed in the data understanding phase. These steps are explained in more details in the Appendix.

4.4 Text Preparation for Deep Learning models

The text preparation method used to implement the deep learning models, i.e., the CNN and Bi-LSTM, are word embeddings, which are a learned representation for text where similar words have similar representations (TensorFlow, 2021). (Rezaeinia et al., 2019) mentions that an extensive training corpus would be required to effectively implement the pre-trained word embeddings such as **word2vec**, **GloVe**, or BERT which are popular techniques of converting words into meaningful vectors (Rezaeinia et al., 2019). Since our training corpus is not very large, the implementation seemed counter-productive. Thus, the word embeddings used for this project are created from our word vectors instead of pre-trained ones.

4.4.1 Word embeddings

Before the word embeddings are generated from the original text, the comments were cleaned by removing stop words and all non-words. Next, the words from each comment id are then reaggregated and joined to the original dataset. Then, to create the word embeddings, the textual data is tokenised to convert the words into a sequence of integers (indices) where each sequence represents a particular word. The maximum number of features that have manually been set for the word embeddings is 10,000, as this presented a best performance. The document count, which represent the unique words in all comments 111,668, and the words' indices are extracted to overview the tokenization of the training data. Once the text is converted to a dictionary of integers, a list of integers for each comment is created, see sample word tokens for first six comments in Fig. 6.

Fig. 6 Sample word tokens for first six comments

```
[[1]]
[1]   24   24 1398   19  345 1811 7365 1794

[[2]]
[1]   12    1 5513 5617 1981 1269

[[3]]
 [1]  122  257  298  102  313 2488   96   28  453 1052   96  222  628   11

[[4]]
[1]  306  808  778 1377   65 7653  197

[[5]]
 [1]  669  626  854 3426 1828   19    1    5    2  362   81   49 2320   12 2767  112  356 9901 7772 1683 3354 9138 2546
[24]  421 1664 1260 3275  485 2265   67  331   76  331 1802 1009  108  531 1003 1369 2572 3275  122  893    2   54    2
[47]   61  990  427

[[6]]
 [1]  660   65  248 3624  224  159    9  236  343    1   12  326  182   74    2   49  531   31   64 3154 9499    5
```

Next, paddings, using 0's, are added at the end of the integer sequence for each comment to make them all of the same length. In order to do that, however, we calculate a statistical summary of the integer sequences including min value, 1st quartile, median, mean, 3rd quartile and the max value (Ross, 2021) to be able to find the maximum cut-off length for the padding, see Table 3.

Table 3 shows that the mean value is 252, and the 3rd quartile value falls around 275. As a result, a middle value, 265, between 250 and 275 is chosen as the maximum cut-off length for the padding. This number is chosen as it is the most optimal configuration for this study. Once the maximum length is established, zeros were added to the integer sequences to makes them all of the same length. After the padding is added, the target variable, i.e., **harassment_or_not** for both the training and testing data, is converted to a multi-class matrix because neural networks cannot take in the standard factor class from a data frame. Then, the deep learning algorithms are trained using the integer sequences created.

4.5 Modelling: Machine Learning Models

This section explains the different models that are used in this study. The machine learning algorithms used include ***Naïve Bayes (NB), Random Forest (RF) and Support Vector Machine (SVM)***, which were trained using the training data created from the data preparation phase. Each algorithm was implemented twice, once using the unigrams and the other using bigrams generated in the data preparation phase. For the random forest algorithm, the maximum number of trees was set to 500 as this presented the best performance. Additionally, the importance of assessing the predictors is set as true. For the support vector machine, moreover, the kernel used in this study is the polynomial kernel as (Noviantho Isa & Ashianti, 2017) have found that the polynomial kernel performed the best for text classification tasks. The machine learning algorithms are evaluated using confusion matrices, which contain information about the algorithms' actual and predicted classifications (Deng et al., 2016).

Table 3 Summary statistics of the integer sequences

Min	1st Quartile	Median	Mean	3rd Quartile	Max
2	60	129	252	275	5000

4.5.1 Naïve Bayes with bigrams and unigrams

The confusion matrix in Fig. 7(a) shows that the class label "6" is the only class that has been correctly predicted, whereas the other classes are not predicted correctly. All the rows of data have been predicted to fall under the class label '6'. The confusion matrix in Fig. 7(b) is similar to the predictions made by the 'Naïve Bayes with bigrams" model. Only the class label '6' is correctly predicted. Once again, mainly all the data has been predicted to fall under the class label '6'. There are a few incorrect predictions where the data is predicted to fall under the class label, '5'.

Fig. 7 Confusion matrix for the Naïve Bayes model using bigrams (**a**) and unigrams (**b**)

	Reference						
Prediction	0	1	2	3	4	5	6
0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0
6	9558	308	101	131	32	11	1

(a) bigrams

	Reference						
Prediction	0	1	2	3	4	5	6
0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0
5	17	1	0	0	0	0	0
6	39106	1696	923	1131	475	95	8

(b) unigrams

4.5.2 Random Forest with bigrams and unigrams

The confusion matrix in Fig. 8(a) shows that the class label 0 has been correctly predicted by the algorithm while the other classes have a higher proportion of wrong predictions. While there are some correct predictions made for the other classes, it is a minuscule amount. The confusion matrix in Fig. 8(b) shows that majority of the data in the class label '0' are predicted correctly. The other classes are correctly predicted to a certain extent; however, many of the predictions fall under the class label '0'. None of the data from the class label '6' is correctly predicted.

4.5.3 Support Vector Machine with bigrams and unigrams

The confusion matrix in Fig. 9(a) shows that most of the values in the class label '0', '2', and '3' are correctly predicted. For classes '1' and '4', most of the predictions fall under the class label '2' and '5' respectively, and classes '5' and '6' show poor prediction results. The confusion matrix in Fig. 9(b) shows that the class label '0', '1', and '4' mostly have correct predictions. The other classes do have a good proportion of correct predictions. However, there are more misclassifications.

4.6 Modelling: Deep Learning Models

The Deep Learning models are implemented using Keras and TensorFlow in R. The neural networks contain different layers, each serving a different purpose. The common layers used between the two neural network algorithms used in this study, Bi-LSTM and CNN, are described below:

- We train our word embeddings for the embedding layer for Natural Language Processing (NLP) problems. The embedding layer takes in:
  - Input dimensions, which represents the size of the vocabulary. In this case, a variable containing the maximum number of features is set to be 10,000 as it proved to be the best configuration.
  - Output dimensions, which represent the length of the dimensions for each vector. In this case, the value is 64.
  - Input length, which represents the maximum length of the sequence (Keras, 2021e). In this case, it is a variable containing the maximum length of the vectors. This value is chosen to be 265 based on the summary of the text length carried out during the EDA.
- The dropout layer will randomly set input units to 0 with a frequency of a set rate at each step during the training, which helps prevent overfitting. The rate is a float value between 0 and 1 (Keras, 2021d).
- The dense layer is a regular, deeply connected neural network layer (Keras, 2021c).
- The activation layer applies an activation function to the output (Keras, 2021h). The activation function can be ReLU, Sigmoid, or SoftMax, among others.

These are the common layers used by the two neural network algorithms implemented in this study. The following sections will individually describe each of the neural networks in more detail.

4.6.1 Bi-LSTM

The Bi-LSTM algorithm has simple structure with an embedding layer, one dropout layer, and one dense layer. The dropout rate was set to 0.1, and the activation function used is SoftMax. The SoftMax layer is usually the last layer

Fig. 8 Confusion matrix for the Random Forest model using bigrams (**a**) and unigrams (**b**)

Reference

Prediction	0	1	2	3	4	5	6
0	9551	302	97	106	24	4	1
1	6	2	0	2	0	0	0
2	1	0	2	2	1	0	0
3	0	2	1	13	3	1	0
4	0	2	1	8	4	6	0
5	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0

(a) bigrams

Reference

Prediction	0	1	2	3	4	5	6
0	33113	722	228	124	28	4	0
1	3598	472	207	143	43	8	0
2	1441	218	176	168	49	4	0
3	631	192	219	447	129	17	0
4	273	66	70	195	159	34	4
5	67	27	23	53	67	28	4
6	0	0	0	1	0	0	0

(b) unigrams

Fig. 9 Confusion matrix for the Support Vector Machine model using bigrams (**a**) and unigrams (**b**)

Reference

Prediction	0	1	2	3	4	5	6
0	4362	55	18	13	5	3	0
1	1277	63	18	12	7	0	0
2	2768	113	37	47	4	1	1
3	442	46	21	29	5	1	0
4	522	26	6	16	8	4	0
5	187	5	1	14	3	2	0
6	0	0	0	0	0	0	0

(a) bigrams

Reference

Prediction	0	1	2	3	4	5	6
0	29216	431	131	69	17	4	0
1	8814	876	374	263	82	14	0
2	663	206	195	217	59	8	0
3	134	104	100	268	120	16	2
4	227	54	95	270	139	36	1
5	65	24	28	42	58	17	5
6	4	2	0	2	0	0	0

(b) unigrams

of a classification network (Keras, 2021m). Since this study is a multi-class classification problem, the SoftMax layer seems ideal.

4.6.2 CNN

The implementation of the CNN algorithm includes more layers compared to the Bi-LSTM algorithm. The CNN algorithm consists of an embedding layer, multiple dropout layers, multiple dense layers, a flatten layer, an activation layer, and a pooling layer. The model's infrastructure is created by modifying the CNN model implemented by (Haddad et al., 2020). The settings and hyperparameters are changed to fit the needs of our study as detailed below:

- A convolutional layer is created by convolving the input over a spatial dimension (Keras, 2021b). The convolutional layer has filters, i.e., the dimensionality of the output space, set to 64, the kernel size, i.e., the size of the 1d convolutional window set to 7, and the activation function used is **ReLU**. As previously mentioned, this model is an adaptation of the model implemented by (Haddad et al., 2020).
- Down-sampling is a process ideally used for image classification where the size of the digital image is made smaller by removing pixels (Zhang et al., 2011). However, in the context of this project, it means reducing the dimensionality to reduce the possibility of overfitting. The parameter is set to take the maximum value over the time dimension to down-sample the input representation (Keras, 2021g).
- A flatten layer us used to flatten the input it receives (Keras, 2021f).

The two models are then compiled and trained using the relevant functions in Keras (Keras, 2021k). For the compilation process, three parameters, loss, optimizer and metrics, were configured. The loss function helps compute the quantity that a model should reduce during the training process (Keras, 2021i). The loss function used for this study is categorical cross-entropy, which computes the loss between the labels and the predictions. The optimizers are algorithms in neural networks that change the attributes of the neural network, such as weights and learning rates, to reduce losses (Keras, 2021l). The optimizer used for this study was the **adam** optimizer, as used by (Haddad et al., 2020). Finally, a metric is used to judge the model's performance (Keras, 2021j). The metric used for this study was accuracy. Accuracy calculates how often the predictions and labels are predicted correctly (Keras, 2021a).

For training the model, on the other hand, the number of training iterations (epochs) is set to 20, which indicates that the model will be trained 20 times. Early stopping callbacks, which are used to monitor the validation loss, are added to prevent the model from overfitting. The minimum delta value is set to 0.001, which means that the training process will stop if the absolute change of the validation loss is less than 0.001. Additionally, the batch size, which is the number of samples per gradient update, is set to 32, and the validation split is set to 0.2, which means that 20% of the training data is used to validate the model. Moreover, the class imbalance issue is handled using class weights. Class weights give all the classes equal weights during the training process. (Zhu et al., 2018) state that the use of class weights help to improve the performance of the minority classes while maintaining the performance for the majority classes. (Burez & Van den Poel, 2009)'s use of class weights to overcome class imbalance significantly improves the model performance. As a result, class weights are preferred over over-sampling because (Chen et al., 2018) found that the use of oversampling techniques to combat class imbalance in their study caused their deep learning models to perform poorly. Finally, the shuffle parameter, which is a Boolean value that states if the training data should be shuffled before each epoch, is set to 'TRUE' in this study.

The deep learning models are first evaluated by visualizing the training history using the loss and accuracy plots. The plots are created during the model training phase, and the resulting plots for the CNN and Bi-LSTM (Figs. 10 and 11) show that the loss graphs for the validation test keep decreasing for every epoch and become more and more horizontal, which indicates that the Bi-LSTM and CNN algorithms exhibit less overfitting (Smith, 2018).

Furthermore, confusion matrices are generated to evaluate both models. Figure 12(a) shows that most of the classes have been correctly predicted by the model. While the model makes misclassifications, almost all classes acquired most of the correct predictions. The confusion matrix in shown in Fig. 12(b) shows a mix of correct and incorrect predictions made by the model. The model was unable to predict for the class label "6". However, the other classes are predicted correctly to a certain extent.

Overall, eight models have been implemented to compare the performance of the different machine learning and deep learning algorithms in detecting harassment severity in text. The performance measures used for the deep learning and machine learning methods are accuracy, weighted precision, weighted recall, and the weighted F1 score. It must be noted that the weighted precision, weighted recall, and weighted F1 score were manually computed using the confusion matrix. The formulas to calculate the precision, recall, and F1 score are as follows (Tripathi et al., 2018):

$$\text{Precision} = \text{Sum of True Positives}/\text{Sum of (True Positives} + \text{False Positives)}$$

$$\text{Recall} = \text{Sum of True Positives}/\text{Sum of (True Positives} + \text{Negatives)}$$

$$\text{F1} = (2 * \text{precision} * \text{recall})/(\text{precision} + \text{recall})$$

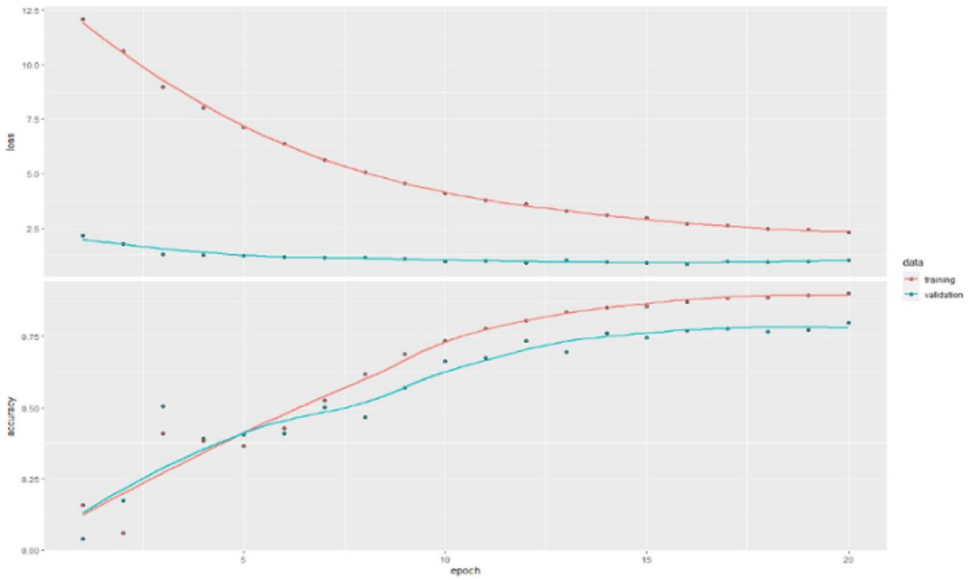

Fig. 10 Accuracy and loss plots for the CNN model

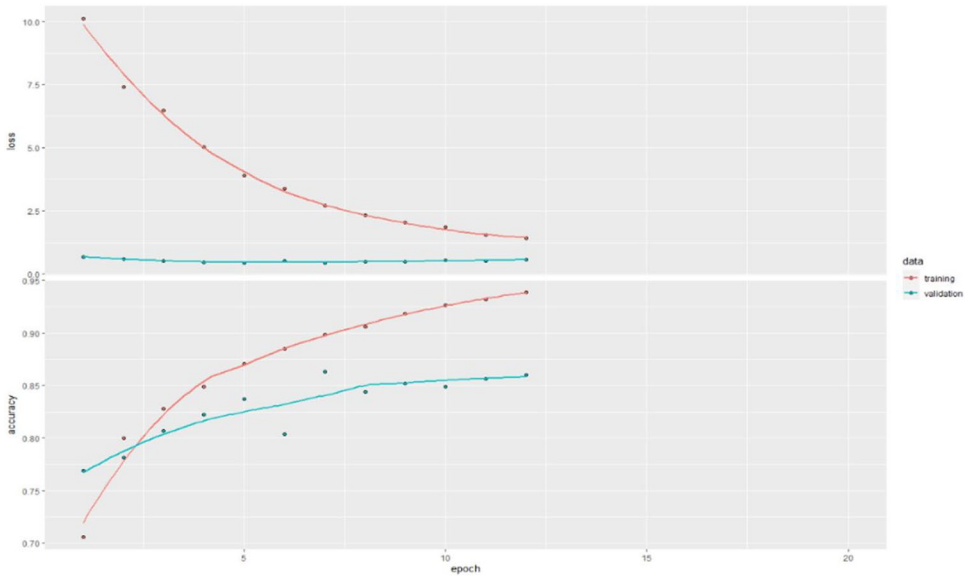

Fig. 11 Accuracy and loss plots for the Bi-LSTM model

Reference

Prediction	0	1	2	3	4	5	6
0	39557	612	122	91	32	3	0
1	2259	669	271	165	33	4	0
2	397	274	259	254	68	9	0
3	166	91	129	234	74	9	0
4	107	53	76	258	179	29	3
5	83	48	61	152	125	52	4
6	451	100	100	129	60	6	1

(a) Bi-LSTM

Reference

Prediction	0	1	2	3	4	5	6
0	36981	1077	392	262	56	5	1
1	4140	322	93	39	7	1	0
2	1603	358	313	372	96	11	0
3	225	67	171	411	205	25	2
4	21	16	38	170	167	38	0
5	12	1	5	21	31	31	5
6	38	6	6	8	9	1	0

(b) CNN

Fig. 12 Confusion matrix for the Bi-LSTM (**a**) and CNN (**b**) models

Table 4 Summary of the performance metrics used to evaluate the performance of the ML and DL models

Algorithm	Accuracy	Precision	Recall	F1 score
Bi-LSTM	0.86	0.90	0.86	0.88
CNN	0.80	0.88	0.80	0.83
Naïve Bayes with unigrams	$2e^{-04}$	$3.39e^{-08}$	$1.84e^{-04}$	$6.78e^{-08}$
Naïve Bayes with bigrams	$1e^{-04}$	$9.72e^{-0.9}$	$9.72e^{-09}$	$1.94e^{-08}$
Random Forest with bigrams	0.94	0.91	0.94	0.92
Random Forest with unigrams	0.79	0.89	0.79	0.83
SVM with bigrams	0.44	0.92	0.44	0.59
SVM with unigrams	0.71	0.90	0.71	0.78

Table 4 shows the results of these metrics used to evaluate the performance of the machine learning and deep learning algorithms developed in this research.

5 Discussion and Implications

5.1 Key Findings

This research study aimed to assess the performance of Machine Learning (ML) and Deep Learning (DL) Algorithms to identify the best technique in detecting the severity of abusive comments online; by investigating the effect of various text pre-processing techniques on the performance of the ML and DL models. The study sought to determine whether deep learning algorithms perform better than machine learning algorithms in detecting the severity of abusive comments instead of only focusing on detecting and classifying abusive comments and hate speech. When considering the use of unigrams for the machine learning algorithms, i.e., SVM, Random Forest and Naïve Bayes, the Random Forest model performs best with an accuracy of 0.79 and an F1 score of 0.83. Similarly, the Random Forest model performs best when considering bigrams for data pre-processing scoring 0.94 and 0.92 for accuracy and F1, respectively. Among the deep learning methods, i.e., CNN and Bi-LSTM, the later performs the best with an accuracy of 0.86 and an F1 score of 0.88. The better performance of the Random Forest algorithm using SMOTE is in line with previous studies and findings concluded by (Al-Garadi et al., 2016) and (Talpur & O'Sullivan, 2020), where they argue that Random Forest is the best performing algorithm in the context of detecting cyberbullying. Table 4 shows that the Naïve Bayes algorithms perform poorly with their accuracy and F1 scores nearing zeros.

5.2 Theoretical Implications

Past research has reported that Deep Learning (DL) approaches such as Recurrent Neural Networks (RNN) and deep convolutional forest perform better compared to machine Learning (ML) algorithms when used for text-based emotion recognition or spam detection in text (Kratzwald et al., 2018; Zinovyeva et al., 2020; Shaaban et al., 2022). This research, nonetheless, aims to investigate the effect of text pre-processing and representation on the performance of machine and deep learning algorithms used for abusive text classification. Observing the performance metrics shown in Table 4, the main contribution of this research constitutes the adoption of bigrams extracted from the text, the process followed for data pre-processing and text representation; as well as the use of the Random Forest (RF) to efficiently detect the severity of hate speech in online platforms. The combination of bigrams and Random Forest (RF) exhibits the best performance compared to all other tested techniques including Deep Learning (DL) algorithm. Our findings are consistent with previous studies that compared random forest and deep learning models using different layer architectures for malware detection (Sewak et al., 2018) and automated legal text classification (Chen et al., 2022). In this study, we show that using bigrams for text pre-processing and representation has significant impact on the performance of the machine learning models. This study adds to the body of research concerned with various approaches for text representation such as GloVe, doc2vec, word2vec and BERT (Chen et al., 2022; Phan et al., 2023; Zhang et al., 2023). Moreover, the positive impact of using bigrams as a text representation technique is studied and proven by previous studies focusing on hate speech detection (Abro et al., 2020), handwritten text recognition (España-Boquera et al., 2011) and student dropout prediction based on textual data (Phan et al., 2023). Additionally, the use of Random Forest (RF) algorithm proves to support prediction and classification explainability as studied by (Ferrettini et al., 2022), which helps understand decisions made by artificial intelligence-based systems (Dennehy et al., 2022).

5.3 Practical Implications

The present study has important practical implications. First, the findings of the study can help practitioners design automated methods for evaluating and detecting the severity of abusive content in online platforms. Furthermore, our findings offer important insights for social media companies (e.g., Twitter, YouTube, etc.), as well all relevant online businesses that offer communication between users (e.g., online forums), in their continuous endeavours to flag abusive content in their platforms and eradicate abusive online

material. As demonstrated in this work, the explained approach highlights the importance of the techniques used to pre-process the textual data and their effect on the accuracy of the algorithm. Various word embeddings are proposed in the literature, see for example (Chen et al., 2022; Phan et al., 2023). Choosing the right method for word embedding during the data preparation phase has an impact of the algorithm accuracy, and thus, integrating the best combination of text representation and algorithm in the mobile apps used by online media platforms and communication companies can help combat the phenomenon of hate speech (Salminen et al., 2020). Our findings further enrich current practice, offering the added aspect of rating and detecting the severity of the hate speech in textual communications allowing the text filtering process to be more configurable and flexible to where users can decide to what extent they would like the information to be filtered from the abusive content. Social media firms can build on the findings presented in this work to develop tools that are capable to censor or flag hate speech based on pre-configured criteria for hate severity.

6 Conclusion, Limitations and Future Research

The present study discusses the widespread presence of abusive and harassing text present in online platforms. While there is a growing body of studies focused on detecting abusive text online, there is surprising paucity of research aiming to evaluate the severity of abusive comments being posted in online platforms. Aiming to address this challenge, the present study proposes a novel approach to pre-process the data and detect the severity of harassment and abusive text in online platforms using a multi-classification scheme; levels are ranging from 0 to 6, where 0 indicates the absence of abuse in a text and 6 indicates the extremely abusive language in the text. This study examines the ability of deep learning algorithms to outperform machine learning algorithms in detecting the severity of abusive and harassing text online. Eight different models are deployed: six machine learning techniques, (i.e., SVM, Random Forest, and Naïve Bayes), each implementing unigrams and bigrams and two different deep learning techniques, (i.e., Bi-LSTM and CNN) using word embeddings. The results show that the Random Forest algorithm with bigrams achieves the best performance in detecting hate speech online.

This research uses a publicly available dataset to demonstrate the different data pre-processing techniques, and to train and evaluate the developed models. This could limit the generalisability of the findings discussed in this study. Additionally, the small size of the data has limited the purpose of using known word embedding techniques such as GloVe, word2vec or BERT, thus manual word embedding was performed. Considering these limitations, future studies should consider employing larger datasets extracted from social media platforms to achieve better evaluation and comparison of the performance of the pre-processing techniques and the developed models. Furthermore, implementing the deep learning algorithms using the pre-trained word embeddings such as GloVe, word2vec and BERT with added attention layers to further improve their performance can be useful to see how they compare against the performance of the Random Forest algorithm. The Naïve Bayes' performance can also be improved by implementing a multinomial Naïve Bayes algorithm. Future work should also include analysing the abusers' previous posts to determine if the context in which the comment was made falls under the "abusive" category or not.

Appendix

Text Preparation for Machine Learning Models

The text preparation for the machine learning models involved creating unigrams and bigrams. The first step to creating unigrams is to convert the text into individual tokens. A word vector is created to store these tokens. Once the tokens are stored, they need to be filtered to make the text features more valuable and informative by removing stop words and all other non-words including numbers, punctuation, words with more than two repeated letters, any single letters, and non-ASCII characters. Once the data is cleaned, stemming is carried out on the tokens. Figure 13 shows six sample tokens in the word vector.

The process of creating bigrams is very similar to the process undergone to produce the unigrams. Then the two tokens (*henceforth referred to as* word1 and word2) are separated. There are a few extra steps in the filtering stage when cleaning the bigrams, as work needs to be done on two different tokens. The two bigram tokens: word1 and word2, need to be filtered to remove any stop words and non-words. Stemming is also carried out for word1 and word2. Once the bigram tokens are cleaned, the bigrams are reunited. Figure 14 illustrate six sample bigrams tokens.

Then, the count for each token is summarised at a comment ID level — this results in an extensive and sparse feature set because many of the tokens will have a majority of 0's across the many word features created. The newly created data frame is such that every token contains a 0 or 1 value. The former indicates the absence of the token in the comment, and 1 represents the presence of the token in the comment. For instance, Fig. 15 shows a small snippet of the word features and associated comment IDs in the newly created data frame. In the first comment, the

Fig. 13 Sample unigram tokens

```
"absolut" "abus"    "academ"  "accept"  "access"  "account"
```

```
[1] "abusing power"       "accept copyrighted"  "acceptable additions" "access deneid"       "account creation"
[6] "accurate information"
```

Fig. 14 Sample bigram tokens

comment_id <chr>	ab <dbl>	abandon <dbl>	abc <dbl>	abid <dbl>	abort <dbl>	abraham <dbl>	absent <dbl>	absolut <dbl>	abstract <dbl>
0000997932d777bf	0	0	0	0	0	0	0	0	0
000103f0d9cfb60f	0	0	0	0	0	0	0	0	0
000113f07ec002fd	0	0	0	0	0	0	0	0	0
00013fa6fb6ef643	0	0	0	0	0	0	0	0	0
0001b41b1c6bb37e	0	0	0	0	0	0	0	0	0
0001d958c54c6e35	0	0	0	0	0	0	0	0	0
00024b59235015f3	0	0	0	0	0	0	0	0	0
00025465d4725e87	0	0	0	0	0	0	0	0	0
0002bcb3da6cb337	0	0	0	0	0	0	0	0	0
0002bfc2abe2a51f	0	0	0	0	0	0	0	0	0

1–10 of 153,806 rows | 1–10 of 3031 columns

Previous 1 2 3 4 5 6 … 100 Next

Fig. 15 The snippet of the unigram word features

comment_id <chr>	abusing power <dbl>	accept copyrighted <dbl>	acceptable additions <dbl>	access deneid <dbl>	account creation <dbl>
0000997932d777bf	0	0	0	0	0
000113f07ec002fd	0	0	0	0	0
0002eeaf4c0cdf35	0	0	0	0	0
0005300084f90edc	0	0	0	0	0
0005be6eea9c30e8	0	0	0	0	0
0006f16e4e9f292e	0	0	0	0	0
00078f8ce7eb276d	0	0	0	0	0
000b08c464718505	0	0	0	0	0
000baabcf7fa7436	0	0	0	0	0
000c6a3f0cd3ba8e	0	0	0	0	0

1–10 of 55,395 rows | 1–6 of 1214 columns

Previous 1 2 3 4 5 6 … 100 Next

Fig. 16 The snippet of the bigram word features

	comment_id <chr>	harassment_or_not <fctr>	ab <dbl>	abandon <dbl>	abc <dbl>	abid <dbl>	abort <dbl>	abraham <dbl>	absent <dbl>
1	0000997932d777bf	0	0	0	0	0	0	0	0
2	000103f0d9cfb60f	0	0	0	0	0	0	0	0
3	000113f07ec002fd	0	0	0	0	0	0	0	0
4	0001b41b1c6bb37e	0	0	0	0	0	0	0	0
5	0001d958c54c6e35	0	0	0	0	0	0	0	0
6	00025465d4725e87	0	0	0	0	0	0	0	0

6 rows | 1–10 of 3032 columns

(a)

	comment_id <chr>	harassment_or_not <fctr>	abusing power <dbl>	accept copyrighted <dbl>	acceptable additions <dbl>	access deneid <dbl>
1	0000997932d777bf	0	0	0	0	0
2	000113f07ec002fd	0	0	0	0	0
3	0005300084f90edc	0	0	0	0	0
4	0006f16e4e9f292e	0	0	0	0	0
5	00078f8ce7eb276d	0	0	0	0	0
6	000b08c464718505	0	0	0	0	0

6 rows | 1–7 of 1215 columns

(b)

Fig. 17 The final resulting unigrams (**a**) and bigrams (**b**) combined with the original dataset

break. <dbl>
0
0
0
0
0
0

	harassment_or_not <fctr>	abusing.power <dbl>	accept.copyrighted <dbl>	acceptable.additions <dbl>	access.deneid <dbl>	account.creation <dbl>
1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	0	0	0	0	0	0
4	0	0	0	0	0	0
5	0	0	0	0	0	0
6	0	0	0	0	0	0

Fig. 18 The final resulting unigrams and bigrams combined with the original dataset with modified variables names

word, **abandon**, is 0 because that word is not present in the comment corresponding to that comment ID. Additionally, Fig. 16 displays the snippet for the bigram word features.

Finally, the newly created unigrams and bigrams features are combined with the original dataset by joining them by using the comment ID as the key. The final resulting unigram data frame with all the added features can be seen in Fig. 17(a) and the final resulting bigram data frame with all the added features is shown in Fig. 17(b).

Some of the algorithms used in this research, however, can't process unigrams used as variables names, see Fig. 10a, because some of them have special meaning for the programming language used in this research, R, such as the unigram "break". Similarly, bigrams used as variables names shown in Fig. 10b, contain spaces and can't be used as columns names. Thus, unigrams variable names were appended with a period (.) and the spaces between the bigrams words were replaced by a period (.) (see Fig. 18).

Lastly, the modified dataset is split into training and testing datasets following a 70/30 split, where 70% of the data is used to train the model, and 30% of the data is used to test the model. In addition, SMOTE is used to handle the class imbalance discovered during the exploratory data analysis (EDA) performed in the data understanding phase.

Authors' Contributions All authors listed in this work have equally participated in this work. In more details:

Dr Alaa Marshan: Worked on idea conception, the literature review, the practical work, the discussion and revised the manuscript.

Ms Farah Nasreen Mohamed Nizar: Worked on idea conception, the introduction, the literature review, and the practical work.

Dr Athina Ioannou: Worked on the introduction, the literature review, the discussion and revised the manuscript.

Dr Konstantina Spanaki: Worked on the introduction, literature review, the discussion and revised the manuscript.

Funding This research didn't use any fund (public or private).

Data Availability The data used in this work is a public dataset.

Declarations

Ethics Approval and Consent to Participate The data used in this research is a public dataset and, thus, ethical approval and consent to participate were not necessary.

Consent for Publication We consent to publish this work in Information Frontiers Systems Journal.

Competing Interests Not applicable.

References

Abro, S., et al. (2020). Automatic hate speech detection using machine learning: A comparative study. *International Journal of Advanced Computer Science and Applications, 11*(8), 484–491. https://doi.org/10.14569/IJACSA.2020.0110861

Al-Ajlan, M. A., & Ykhlef, M. (2018). Optimized twitter cyberbullying detection based on deep learning. In *21st Saudi Computer Society National Computer Conference, NCC 2018*, pp. 1–5. https://doi.org/10.1109/NCG.2018.8593146

Alam, S., & Yao, N. (2019). The impact of preprocessing steps on the accuracy of machine learning algorithms in sentiment analysis. *Computational and Mathematical Organization Theory, 25*(3), 319–335. https://doi.org/10.1007/s10588-018-9266-8

Al-Garadi, M. A., Varathan, K. D., & Ravana, S. D. (2016). Cybercrime detection in online communications: The experimental case of cyberbullying detection in the Twitter network. *Computers in Human Behavior, 63*, 433–443. https://doi.org/10.1016/j.chb.2016.05.051

Au, C. H., Ho, K. K. W., & Chiu, D. K. W. (2021). The role of online misinformation and fake news in ideological polarization: barriers, catalysts, and implications. *Information Systems Frontiers*, 1331–1354. https://doi.org/10.1007/s10796-021-10133-9

Awal, M. A., Rahman, M. S., & Rabbi, J. (2018). Detecting abusive comments in discussion threads using naïve bayes. *2018 International Conference on Innovations in Science, Engineering and Technology, ICISET 2018*, (October), 163–167. https://doi.org/10.1109/ICISET.2018.8745565

Badjatiya, P. et al. (2017). Deep learning for hate speech detection in tweets. *CoRR*, abs/1706.0.

Bashar, M. A., et al. (2019). Misogynistic tweet detection: Modelling CNN with small datasets. *Communications in Computer and Information Science, 996*, 3–16. https://doi.org/10.1007/978-981-13-6661-1_1

Burez, J., & Van den Poel, D. (2009). Handling class imbalance in customer churn prediction. *Expert Systems with Applications, 36*(3, Part 1), 4626–4636. https://doi.org/10.1016/j.eswa.2008.05.027

Castaño-Pulgarín, S. A. et al. (2021). Internet, social media and online hate speech. Systematic review. *Aggression and Violent Behavior*, 58(January). https://doi.org/10.1016/j.avb.2021.101608

Chen, H., et al. (2022). A comparative study of automated legal text classification using random forests and deep learning. *Information Processing and Management, 59*(2), 102798. https://doi.org/10.1016/j.ipm.2021.102798

Chen, H., McKeever, S., & Delany, S. J. (2018). A comparison of classical versus deep learning techniques of abusive content detection on social media sites. *' in Social Informatics. Springer,* . pp. 117–133. https://doi.org/10.1007/978-3-030-01129-1_8

Deng, X., et al. (2016). An improved method to construct basic probability assignment based on the confusion matrix for classification problem. *Information Sciences, 340–341*, 250–261. https://doi.org/10.1016/j.ins.2016.01.033

Dennehy, D., et al. (2022). Artificial intelligence (AI) and information systems : Perspectives to responsible AI. *Information Systems Frontiers, 24*, 49–75.

DePaolo, C. A., & Wilkinson, K. (2014). Get your head into the clouds: Using word clouds for analyzing qualitative assessment data. *TechTrends, 58*(3), 38–44. https://doi.org/10.1007/s11528-014-0750-9

Desrul, D. R. K., & Romadhony, A. (2019). Abusive language detection on indonesian online news comments. In *2019 2nd International Seminar on Research of Information Technology and Intelligent Systems*, ISRITI 2019, pp. 320–325. https://doi.org/10.1109/ISRITI48646.2019.9034620

Eshan, S. C., & Hasan, M. S. (2017). An application of machine learning to detect abusive Bengali text. In *2017 20th International Conference of Computer and Information Technology (ICCIT)*, pp. 1–6. https://doi.org/10.1109/ICCITECHN.2017.8281787

España-Boquera, S., et al. (2011). Improving offline handwritten text recognition with hybrid HMM/ANN models. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 33*(4), 767–779. https://doi.org/10.1109/TPAMI.2010.141

Ferrettini, G., et al. (2022). Coalitional Strategies for Efficient Individual Prediction Explanation. *Information Systems Frontiers, 24*(1), 49–75. https://doi.org/10.1007/s10796-021-10141-9

Founta, A. M. et al. (2019). A unified deep learning architecture for abuse detection. *WebSci 2019 - Proceedings of the 11th ACM Conference on Web Science*, abs/1802.0, pp. 105–114. https://doi.org/10.1145/3292522.3326028

Gambäck, B., & Sikdar, U. K. (2017). Using convolutional neural networks to classify hate-speech. In *Proceedings of the First Workshop on Abusive Language Online*. Vancouver, BC, Canada: Association for Computational Linguistics, pp. 85–90. https://doi.org/10.18653/v1/w17-3013

Georgakopoulos Spiros V., Tasoulis Sotiris K., Vrahatis Aristidis G., & Plagianakos Vassilis P. (2018). Convolutional neural networks for toxic comment classification. *In Proceedings of the 10th Hellenic Conference on Artificial Intelligence, 35*, 6.

Haddad, B. et al. (2020). {A}rabic offensive language detection with attention-based deep neural networks. *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*, (May), pp. 76–81.

Ibrohim, M. O., & Budi, I. (2018). A Dataset and Preliminaries Study for Abusive Language Detection in Indonesian Social Media. *Procedia Computer Science, 135*, 222–229. https://doi.org/10.1016/j.procs.2018.08.169

Jabeen, F., et al. (2023). The dark side of social media platforms: A situation-organism-behaviour-consequence approach. *Technological Forecasting and Social Change, 186*(PA), 122104. https://doi.org/10.1016/j.techfore.2022.122104

Kaggle. (2021). *Toxic comment classification challenge*.

Karatsalos, C., & Panagiotakis, Y. (2020). Attention-based method for categorizing different types of online harassment language. *Communications in Computer and Information Science*, 1168 CCIS, pp. 321–330. https://doi.org/10.1007/978-3-030-43887-6_26

Keerthi Kumar, H. M., & Harish, B. S. (2018). Classification of short text using various preprocessing techniques: An empirical evaluation. *Advances in Intelligent Systems and Computing, 709*, 19–30. https://doi.org/10.1007/978-981-10-8633-5_3

Keras. (2022a). Accuracy metrics. Access at: https://keras.io/api/metrics/accuracy_metrics/

Keras. (2022b). Conv1D layer. Access at: https://keras.io/api/layers/convolution_layers/convolution1d/

Keras. (2022c).Dense layer. Access at: https://keras.io/api/layers/core_layers/dense/

Keras. (2022d). Dropout layer. Access at: https://keras.io/api/layers/regularization_layers/dropout/

Keras. (2022e). Embedding layer. Access at: https://keras.io/api/layers/core_layers/embedding/

Keras. (2022f). Flatten layer. Access at: https://keras.io/api/layers/reshaping_layers/flatten/

Keras. (2022h). Keras layers API. Access at: https://keras.io/api/layers/

Keras. (2022i). Losses. Access at: https://keras.io/api/losses/

Keras. (2022g). GlobalMaxPooling1D layer. Access at: https://keras.io/api/layers/pooling_layers/global_max_pooling1d/

Keras. (2022j). Metrics. Access at: https://keras.io/api/metrics/

Keras. (2022k). Model training APIs. Access at: https://keras.io/api/models/model_training_apis/

Keras. (2022l). Optimizers. Access at: https://keras.io/api/optimizers/

Keras. (2022m). *Softmax layer*. Access at: https://keras.io/api/layers/activation_layers/softmax/

Kiilu, K. K. et al. (2018). Using naïve bayes algorithm in detection of hate tweets. *International Journal of Scientific and Research Publications (IJSRP), 8*(3). https://doi.org/10.29322/ijsrp.8.3.2018.p7517

Kim, J. Y., Sim, J., & Cho, D. (2022). Identity and status: When counterspeech increases hate speech reporting and why. *Information Systems Frontiers* [Preprint], (0123456789).https://doi.org/10.1007/s10796-021-10229-2

Kratzwald, B., et al. (2018). Deep learning for affective computing: Text-based emotion recognition in decision support. *Decision*

Support Systems, 115(March), 24–35. https://doi.org/10.1016/j.dss.2018.09.002

Kwak, S. K., & Kim, J. H. (2017). Statistical data preparation: Management of missing values and outliers. *Korean Journal of Anesthesiology, 70*(4), 407–411. https://doi.org/10.4097/kjae.2017.70.4.407

Lecluze, C., et al. (2013). Which granularity to bootstrap a multilingual method of document alignment: Character N-grams or word N-grams? *Procedia - Social and Behavioral Sciences, 95*, 473–481. https://doi.org/10.1016/j.sbspro.2013.10.671

Lee, Y., Yoon, S., & Jung, K. (2019). Comparative studies of detecting abusive language on twitter. *CoRR*, abs/1808.1, pp. 101–106. https://doi.org/10.18653/v1/w18-5113

Lynn, T. et al. (2019). A comparison of machine learning approaches for detecting misogynistic speech in urban dictionary. In *2019 International Conference on Cyber Situational Awareness, Data Analytics And Assessment (Cyber SA)*, pp. 1–8. https://doi.org/10.1109/CyberSA.2019.8899669

MacAvaney, S., et al. (2019). Hate speech detection: Challenges and solutions. *PLoS ONE, 14*(8), 1–16. https://doi.org/10.1371/journal.pone.0221152

Malmasi, S., & Zampieri, M. (2017). Detecting hate speech in social media. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, {RANLP} 2017*. Varna, Bulgaria: INCOMA Ltd., pp. 467–472. https://doi.org/10.26615/978-954-452-049-6_062

Marbán, Ó., Mariscal, G. and Segovia, J. (2009). A Data mining & knowledge discovery process model in real life applications. *IntechOpen*, (February), p. 436.

Marshan, A., Kansouzidou, G., & Ioannou, A. (2021). Sentiment Analysis to Support Marketing Decision Making Process: A Hybrid Model. *Advances in Intelligent Systems and Computing, 1289*, 614–626. https://doi.org/10.1007/978-3-030-63089-8_40

Martinez-Plumed, F., et al. (2019). CRISP-DM twenty years later: From data mining processes to data science trajectories. *IEEE Transactions on Knowledge and Data Engineering, 33*(8), 1. https://doi.org/10.1109/tkde.2019.2962680

Marwa, T., Salima, O., & Souham, M. (2018). Deep learning for online harassment detection in tweets. In *2018 3rd International Conference on Pattern Analysis and Intelligent Systems (PAIS)*, pp. 1–5. https://doi.org/10.1109/PAIS.2018.8598530

Matamoros-fernández, A. (2021). Racism, Hate Speech, and Social Media : A Systematic Review and Critique. *Television & New Media, 22*(2), 205–224. https://doi.org/10.1177/1527476420982230

Matamoros-Fernández, A., & Farkas, J. (2021). Racism, hate speech, and social media: A systematic review and critique. *Television and New Media, 22*(2), 205–224. https://doi.org/10.1177/1527476420982230

Meske, C., & Bunde, E. (2022). *Design principles for user interfaces in ai-based decision support systems: the case of explainable hate speech detection, information systems frontiers*. Springer US. https://doi.org/10.1007/s10796-021-10234-5

Mikolov, T. et al. (2013). Distributed representation of words and phrases and their compositionality. *Advances in neural information processing systems*, 3111–3119. https://doi.org/10.18653/v1/d16-1146

Modha, S., et al. (2020). Detecting and visualizing hate speech in social media: A cyber Watchdog for surveillance. *Expert Systems with Applications, 161*, 113725. https://doi.org/10.1016/j.eswa.2020.113725

Muneer, A., & Fati, S.M. (2020). A comparative analysis of machine learning techniques for cyberbullying detection on twitter. *Future Internet*, *12*(11). https://doi.org/10.3390/fi12110187

Nascimento, F. R. S., Cavalcanti, G. D. C., & Da Costa-Abreu, M. (2022). Unintended bias evaluation: An analysis of hate speech detection and gender bias mitigation on social media using ensemble learning. *Expert Systems with Applications, 201*(April), 117032. https://doi.org/10.1016/j.eswa.2022.117032

Nobata, C. et al. (2016). Abusive language detection in online user content. In *Proceedings of the 25th International Conference on World Wide Web*. Republic and Canton of Geneva, CHE: International World Wide Web Conferences Steering Committee (WWW '16), pp. 145–153. https://doi.org/10.1145/2872427.2883062

Novalita, N., et al. (2019). Cyberbullying identification on twitter using random forest classifier. *Journal of Physics: Conference Series, 1192*, 12029. https://doi.org/10.1088/1742-6596/1192/1/012029

Noviantho Isa, S. M., & Ashianti, L. (2017). Cyberbullying classification using text mining. In *2017 1st International Conference on Informatics and Computational Sciences (ICICoS)*, pp. 241–246. https://doi.org/10.1109/ICICOS.2017.8276369

Oriola, O., & Kotze, E. (2020). Evaluating machine learning techniques for detecting offensive and hate speech in south african tweets. *IEEE Access, 8*, 21496–21509. https://doi.org/10.1109/ACCESS.2020.2968173

Özel, S.A. et al. (2017). Detection of cyberbullying on social media messages in Turkish. In *2017 International Conference on Computer Science and Engineering (UBMK)*, pp. 366–370. https://doi.org/10.1109/UBMK.2017.8093411

Park, J. H., & Fung, P. (2017). One-step and Two-step Classification for Abusive Language Detection on Twitter. *CoRR*, abs/1706.0, pp. 41–45. https://doi.org/10.18653/v1/w17-3006

Pennington, J., Socher, R., & Manning, C. (2014). GloVe: global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing ({EMNLP})*. Doha, Qatar: Association for Computational Linguistics, pp. 1532–1543. https://doi.org/10.3115/v1/D14-1162

Phan, M., De Caigny, A., & Coussement, K. (2023). A decision support framework to incorporate textual data for early student dropout prediction in higher education. *Decision Support Systems, 168*(January), 113940. https://doi.org/10.1016/j.dss.2023.113940

Pitsilis, G. K., Ramampiaro, H., & Langseth, H. (2018). Effective hate-speech detection in Twitter data using recurrent neural networks. *Applied Intelligence, 48*(12), 4730–4742. https://doi.org/10.1007/s10489-018-1242-y

Putri, T., et al. (2020). A comparison of classification algorithms for hate speech detection. *IOP Conference Series: Materials Science and Engineering, 830*, 32006. https://doi.org/10.1088/1757-899X/830/3/032006

Rahul et al. (2020). Classification of online toxic comments using machine learning algorithms. In *2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS)*, pp. 1119–1123. https://doi.org/10.1109/ICICCS48265.2020.9120939

Rezaeinia, S. M., et al. (2019). Sentiment analysis based on improved pre-trained word embeddings. *Expert Systems with Applications, 117*, 139–147. https://doi.org/10.1016/j.eswa.2018.08.044

Ross, S. M. (2021). Chapter 2 - descriptive statistics. In S. M. Ross (Ed.), *Introduction to probability and statistics for engineers and scientists* (6th ed., pp. 11–61). Academic Press. https://doi.org/10.1016/B978-0-12-824346-6.00011-9

Saif, H. et al. (2014). On stopwords, filtering and data sparsity for sentiment analysis of twitter. In *Proceedings of the 9th International Conference on Language Resources and Evaluation, LREC 2014*. Reykjavik, Iceland: European Language Resources Association (ELRA), pp. 810–817.

Salminen, J., et al. (2020). Developing an online hate classifier for multiple social media platforms. *Human-centric Computing and Information Sciences, 10*(1), 1–34. https://doi.org/10.1186/s13673-019-0205-6

Sewak, M., Sahay, S. K., & Rathore, H. (2018). Comparison of deep learning and the classical machine learning algorithm for the malware detection. *Proceedings - 2018 IEEE/ACIS 19th International Conference on Software Engineering, Artificial Intelligence,*

Networking and Parallel/Distributed Computing, SNPD 2018, pp. 293–296. https://doi.org/10.1109/SNPD.2018.8441123

Shaaban, M. A., Hassan, Y. F., & Guirguis, S. K. (2022). Deep convolutional forest: a dynamic deep ensemble approach for spam detection in text. *Complex and Intelligent Systems, 8*(6), 4897–4909. https://doi.org/10.1007/s40747-022-00741-6

Singh, J. P., et al. (2022). Attention-based LSTM network for rumor veracity estimation of tweets. *Information Systems Frontiers, 24*(2), 459–474. https://doi.org/10.1007/s10796-020-10040-5

Smith, L. N. (2018). A disciplined approach to neural network hyper-parameters: Part 1 - learning rate, batch size, momentum, and weight decay. *arXivpreprintarXiv:1803.09820.*

Talpur, B. A., & O'Sullivan, D. (2020). Cyberbullying severity detection: A machine learning approach. *PLOS ONE, 15*(10), 1–19. https://doi.org/10.1371/journal.pone.0240924

TensorFlow. (2021). Word embeddings. Access at: https://www.tensorflow.org/text/guide/word_embeddings

Tontodimamma, A., et al. (2021). Thirty years of research into hate speech: topics of interest and their evolution. *Scientometrics, 126*(1), 157–179. https://doi.org/10.1007/s11192-020-03737-6

Tripathi, D., et al. (2018). Credit scoring model based on weighted voting and cluster based feature selection. *Procedia Computer Science, 132*, 22–31. https://doi.org/10.1016/j.procs.2018.05.055

Vidgen, B., Margetts, H., & Harris., A. (2019). *How much online abuse is there ? A systematic review of evidence for the UK*. In Alan Turing Institute. Access at: https://www.turing.ac.uk/sites/default/files/2019-11/online_abuse_prevalence_full_24.11.2019_-_formatted_0.pdf

Wu, G., & Zhu, J. (2020). Multi-label classification: Do hamming loss and subset accuracy really conflict with each other? *In Proceedings of the 34th International Conference on Neural Information Processing Systems, NIPS'20. Curran Associates Inc.*

Zhang, Y., et al. (2011). Interpolation-dependent image downsampling. *IEEE Transactions on Image Processing, 20*(11), 3291–3296. https://doi.org/10.1109/TIP.2011.2158226

Zhang, D., et al. (2022). A deep learning approach for detecting fake reviewers: Exploiting reviewing behavior and textual information. *Decision Support Systems, 166*(January 2022), 113911. https://doi.org/10.1016/j.dss.2022.113911

Zhu, M., et al. (2018). Class weights random forest algorithm for processing class imbalanced medical data. *IEEE Access, 6*, 4641–4652. https://doi.org/10.1109/ACCESS.2018.2789428

Zinovyeva, E., Härdle, W. K., & Lessmann, S. (2020). Antisocial online behavior detection using deep learning. *Decision Support Systems, 138*(July), 113362. https://doi.org/10.1016/j.dss.2020.113362

Alaa Marshan is a Senior Lecturer at the School of Computer Science and Electronic Engineering at the university of Surrey, where he teaches a variety of topics related to data science and machine learning. His research primarily focuses on intelligent data analysis, information management and improving operational business information systems. He has a specific interest in applying Social Network Analysis (SNA), and Machine Learning (ML) techniques in various research domains such as Finance and Healthcare to support information inferencing and decision making; developing new methods and models to analyse large transactional-based datasets and enhancing human sense-making within organisational settings for better decision-making.

Farah Nasreen Mohamed Nizar holds the position of Associate Consultant at Capco, a renowned business management consultancy firm located in London, England. She is an alumna of Brunel University London, where she graduated with an MSc degree in Data Science and Analytics, achieving the distinction. Her academic and professional endeavours primarily focus on business management consultancy and the creation of solutions aimed at improving the decision-making processes within this domain.

Athina Ioannou is a Lecturer in Business Analytics at Surrey Business School, University of Surrey. Her research is primarily around data and information management focusing on technology adoption and diffusion as well as the implications of technology use in individual, organisational and social contexts. She has also particular interest in the application of mindfulness within organisational settings in order to improve both individual and business outcomes.

Konstantina Spanaki is an Associate Professor of IS and Supply Chain Management at Audencia Business School. Prior to this role Konstantina has been working at Loughborough University and Imperial College London in areas around Technology Management. Konstantina's main research areas lie within the intersection of Information Systems (IS) and Operations Management (OM). Recently, she is actively involved in projects related to Digital Supply Chain, Data and Technology Management, Data Sharing and Disruptive Technologies.