UDK 004.738.5:007.52

DOI 10.7251/ZRSNG2324063J

Decoding Digital Discourse: Artificial Intelligence – powered Cyberbullying Detection

Aleksandar Jokic, Marko Sarac, University Singidunum Belgrade

Abstract— **Social media platforms have fundamentally transformed how people share information and communicate. While they offer significant benefits, they also pose challenges, such as the increasing prevalence of cyberbullying. While many studies have emphasized the accuracy of text classification techniques for detecting cyberbullying, this research explores the potential of automating not just the detection but also the reporting of harmful posts. We developed a Support Vector Machine model using WEKA, designed to identify cyberbullying statements in the English language. This model yielded an accuracy of 57% with a kappa score of 0.2094. After developing the model, we extracted public posts from Twitter and applied text preprocessing methods, including cleaning and tokenization. These preprocessed data were then transformed into a Bag-of-Words (BoW) representation. When a post is identified as cyberbullying by our model, a comprehensive report is generated detailing the author's name, post content, and the timestamp. This innovative method holds promise for the timely detection of malicious content, offering social media platform administrators an efficient tool for prompt intervention.**

Keywords – ***Cyberbullying Detection, Natural Language Processing, Text Classification, Support Vector Machine, Crowdsourcing.***

I. INTRODUCTION

The rise of digital social platforms has provided novel avenues for online engagement. In contemporary times, individuals can effortlessly communicate through various means such as emails, instant messages, forums, and social networking platforms. Nevertheless, the surge in social media utilization has given rise to significant societal challenges, with cyberbullying becoming particularly salient.

Cyberbullying is delineated as the purposeful, recurrent, and hostile use of digital mechanisms to torment or distress an individual (Stopbullying.gov, 2014). Such practices encompass the distribution of intimidating messages, spreading misinformation, exhibiting degrading images, or digital aggression. The virtual nature of cyberbullying amplifies its severity compared to conventional bullying, partly due to the anonymity it offers via pseudonymous profiles and the broad digital audience it reaches.

Recognizing the global challenge posed by cyberbullying, many governments have initiated measures to counteract it. For example, Austria introduced a law 2015 mandating all primary and secondary educational institutions to implement anti-cyberbullying measures. Later, a proposed bill in 2021 sought to penalize cyber bullies with incarceration periods ranging from six months to six years with monetary penalties, too. Beyond legislative action, digital platforms worldwide have adopted initiatives to protect their users. Common approaches include user moderation to identify and remove offensive content, enhanced privacy settings, and specialized reporting systems. Prominent platforms like YouTube offer a "Safety Mode", Facebook employs moderation and profanity filters, and Twitter has a "Mute" feature.

Despite the breadth of the digital landscape, current measures frequently need more comprehensive effectiveness. The onus to report typically lies with the victims, highlighting the urgent need for technological innovations to combat online bullying. This study explores the potential of automating cyberbullying post detection on social platforms using text classification and Support Vector Machine (SVM) techniques, irrespective of language or region.

II. REVIEW OF RELATED LITERATURE

Over time, various techniques have been introduced to identify instances of cyberbullying on social media platforms. Many of these techniques tackle the issue by framing it as a categorization challenge, segmenting messages into groups like 'cyberbullying' and 'non-cyberbullying'.

Dinakar, Reichart, and Lieberman [1] proposed a structured machine-learning technique to spot cyberbullying events. They collected 50,000 comments from YouTube and segmented them into four distinct groups: physical traits, sexuality, ethnic and cultural backgrounds, and intellectual capacity. Their analysis indicated that JRip delivered optimal accuracy, while SVM was deemed the most consistent using kappa metrics. Notably, binary classification systems outperformed those designed for multiple labels.

In 2015, Van Hee and colleagues [2] delved into the linguistic aspects of cyberbullying, distinguishing them into detailed categories, encompassing threats, sexual comments, insults, curses, defences, slander, and encouragements. They further mapped out the roles present in a cyberbullying scenario, namely bully, target, observer-defender, and observer-ally. Their research incorporated 90,000 German comments from Ask.fm, leveraging the Support Vector Machine (SVM) for the categorization process. Their results displayed a Kappa score of 0.69 for identifying cyberbullying events and scores ranging between 0.52 and 0.66 for the various categories.

Dadvar, Jong, Ordeiman, and Trieschnigg [3] pursued a Gender-Centric Method to discern cyberbullying on Myspace. Using a Support Vector Machine with WEKA, they trained their classifier on a dataset from Fundacion Barcelona Media, containing 381,000 posts, of which females and 64% males penned 34%. Impressively, their method elevated the baseline by 39% in precision, 6% in recall, and 15% in the F-measure.

Cheung and colleagues [4] research focused on discerning cyberbullying roles such as accuser, perpetrator, defender, informant, and victim. Their study comprised 6,000 comments/posts from platforms like Facebook and YouTube. They utilized the Support Vector Machine to distinguish cyberbullying events and their respective roles. Their most effective model achieved an accuracy rate of 59.7% with 171 distinct word attributes and a Kappa score of 42.3% in identifying the roles within cyberbullying.

III. THEORETICAL FOUNDATIONS

1) Segmentation of the Audience

The concept of audience segmentation, proposed by Ervin Goffman, emphasizes the varied roles individuals assume in diverse scenarios to present themselves in a positive light. This perspective allows us to understand how one's behaviour may change depending on the audience and context, shedding light on cyberbullying phenomena. Firstly, individuals can effortlessly disguise their true identity online using altered photos, pseudonyms, and fabricated contact details. The perceived anonymity on digital platforms can act as a catalyst, prompting individuals to exhibit behaviours or make statements that they would not typically do in indirect interactions. Furthermore, given the limitless expanse of the digital realm, the audience is not restricted to a particular locale like a school or workplace but could potentially span globally.

Within Goffman's theory, he identifies three pivotal roles: the performer, the audience, and the outsider. These can be equated to the roles of a victim, bully, and observer in a bullying context. When we conceptualize bullying as a theatrical act, it provides a lens through which we can see the observer group as an audience and how varying environments might influence young individuals' actions towards their peers. Goffman delineates three zones of social engagement: the front stage (the public performance space), the backstage (a private space for performers to prep or for group members to jointly devise the image they aim to project), and the external zone, which is not encompassed by either the front or backstage. Through Goffman's performance theory, cyber interactions can be seen as the bully operating backstage, influencing the victim on the more public front stage. The backstage, being a secluded space, offers the bully both the opportunity and privacy to strategize their actions. The inherent distance in online interactions allows the bully to control the image they project better, hide their true self, and leave their actions open to broader interpretations.

2) Text Classification

Text classification involves assigning predefined categories to textual documents. Manually sorting documents into respective categories can be a time-consuming endeavour, particularly with a vast amount of text. Thankfully, machine learning offers an automated solution to text classification. Utilizing machine learning, text classification aims to construct classifiers by recognizing category traits from a collection of previously categorized documents, as noted by Sebastiani (2002). There exists a variety of classifiers, each tailored to specific text classification challenges. Hence, selecting an appropriate classifier becomes essential for optimal system performance. The criterion a classifier decides is derived directly from the training data. Consequently, after the training phase, the classifier can categorize new, unseen data. This methodology is often referred to as statistical text classification.

3) Support Vector Machines

Support Vector Machines (SVM) is a supervised learning algorithm for classification tasks. When provided with a collection of labelled training data, the SVM algorithm builds a model to categorize new, unseen data points into one of the predefined categories, making it a non-probabilistic binary linear classifier. Conceptually, SVM represents each data point (or support vector) in a multidimensional space. The primary objective of SVM is to identify a dividing line, or more generally, a hyperplane, that best separates the data points based on their respective labels. Alongside the primary hyperplane, two parallel dashed lines are established, indicating the nearest data points from each class to this hyperplane. The gap between these dashed lines and the primary hyperplane is called the "margin". An optimal hyperplane maximizes this margin. When new data is introduced, its position relative to the hyperplane decides the category to which it belongs.

4) Overview of Additional Machine Learning Algorithms

a) Naïve Bayes

Naïve Bayes is a classification technique rooted in the Bayes Theorem. It assumes that every feature is independent and contributes separately to the probability of an item's class designation, ignoring potential correlations among them. Given a set of features, it predicts a class using probability based on the formula:

$$\mathrm{P(E)} = \frac{[\mathrm{P(H)} * \mathrm{P(H)}]}{\mathrm{P(E)}}$$

A strength of the Naïve Bayes algorithm is its efficiency, only necessitating a single scan of the training data. Plus, it is adept at training with limited datasets. However, its assumption of feature independence can limit its performance in datasets with interrelated features.

b) J48

J48 is an implementation of the C4.5 decision tree algorithm tailored for classification. It forms binary trees, employing information entropy to model data classification. The algorithm identifies the attribute providing the highest normalized information gain for data splitting and continues recursively on refined subsets. The division halts once all subset instances belong to an identical class, leading to a leaf node in the tree specifying that class. J48 is versatile, accommodating continuous and discrete attributes, missing values, and differing attribute costs. Furthermore, it supports post-hoc tree pruning.

c) ZeroR

ZeroR is a rudimentary rule-based classifier, concentrating solely on the target while dismissing predictors. By referencing a frequency table, it discerns the predominant class. The purpose is to pinpoint the mean (for numeric targets) or mode (for nominal targets). While ZeroR lacks predictive potency, it is a foundational baseline, offering a performance reference for other classifiers.

d) Decision Stump

A Decision Stump is a singular-level decision tree. Comprising one root linked to its terminal leaves, it's also termed a 1-rule due to its predictions based on the value of a single input feature (Holte, 1993). Variations exist based on the input type. Nominal features may lead to a stump with leaves for every feature value or two leaves – one for a specific category and another for the remaining categories. Binary features align with these two structures. Moreover, for continuous features, a threshold divides the stump into two leaves: one for values below the threshold and another for those above.

e) Random Forest

Random Forest is an ensemble learning method that uses multiple decision trees during training and outputs the average prediction of the individual trees for regression tasks or the class with the most votes for classification tasks. Each tree is constructed using a bootstrap sample of the data and random subsets of the predictors. This ensures that the trees are uncorrelated and, therefore, reduces the variance of the predictions. A characteristic feature of Random Forest is its ability to measure the importance of predictors and its robustness to overfitting.

f) REPTree

REPTree stands for Reduced Error Pruning Tree. It is a decision tree algorithm that constructs a tree using information gain and prunes it using a method called reduced-error pruning. This algorithm integrates the principles of a regression tree, creating multiple trees over various iterations and then selecting the optimal tree to represent the dataset. The pruning mechanism reduces the likelihood of overfitting, making the algorithm more generalizable to unseen data.

g) Decision Table

This algorithm operates on the principle of simplifying data by transforming it into a decision table, which has the same number of attributes as the original dataset. The classification of new data is determined by matching its attribute values with the rows of the decision table. The Decision Table algorithm employs the wrapper method to discern the best subset of attributes to be included, ensuring that irrelevant or redundant attributes are removed, leading to a more concise and effective decision-making tool.

h) Hoeffding Tree

Originating from the Hoeffding bound concept, the Hoeffding Tree is designed to decide when enough data has been seen to make confident decisions about splits. It is particularly adept for data stream mining due to its consistent learning time. The Hoeffding bound determines the number of data instances required to decide on an attribute split with a certain confidence level. One of its salient features is its consistency in producing similar results regardless of the underlying probability distributions, although the required number of observations may vary.

i) JRip (RIPPER)

JRip, or RIPPER (Repeated Incremental Pruning to Produce Error Reduction), is a rule-based classifier. The algorithm iteratively refines its rule set by focusing on misclassified instances. For each class, JRip creates a rule set and then moves on to the next class. This process continues until all classes are addressed, resulting in a comprehensive rule-based model for the entire dataset.

j) OneR

OneR, which stands for "One Rule", is a simplistic classification algorithm that functions by identifying the single attribute that performs the best at predicting the class value. It creates one rule for every attribute and picks the rule with the smallest error rate. If multiple rules possess the same error rate, a rule is chosen at random. Despite its simplicity, OneR can produce decent results in various scenarios. The rule generation revolves around discerning the most frequent class for each attribute value. OneR's straightforward nature and ability to pinpoint pertinent patterns in data make it an efficient tool for preliminary data analysis.

5) Performance Measures for Classification

When assessing the performance of classification algorithms, it's imperative to consider various metrics to ensure a comprehensive evaluation. Each metric provides a different perspective on the model's capabilities.

a) Accuracy

Accuracy reflects the overall effectiveness of a classifier by calculating the ratio of correctly classified instances to the total instances.

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

However, its main limitation is that it might not be suitable for imbalanced datasets, where the distribution of classes is skewed.

b) Kappa Statistics

The Kappa statistic (or Cohen's kappa) measures the agreement between two raters who each classify items into categorical classes. The idea is to account for the possibility of agreement occurring by chance, thus providing a more robust measure than a straightforward per cent agreement.

c) Precision

Precision gauges the model's reliability when it predicts a positive class.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

High precision indicates fewer false positives, meaning that when the model predicts an instance as positive, it is likely correct.

In conclusion, selecting the appropriate performance measure is crucial for model evaluation. While accuracy is the most common metric, more is needed in cases with imbalanced datasets or when false positives and false negatives have different implications. Combining multiple metrics provides a more holistic view of the classifier's performance.

d) Support Vector Machine (SVM)

Support Vector Machine (SVM) is a widely used algorithm for classification tasks. It works by finding a hyperplane that best divides a dataset into classes. For this project, SVM is leveraged to classify data into cyberbullying, non-cyberbullying, or ambiguous cyberbullying categories. The words from the Bag-of-Words representation act as features, and SVM learns from them to distinguish cyberbullying instances.

e) Cyberbullying Detection Model

This phase involves using the SVM algorithm on the entire dataset of 2,000 statements. Employing the WEKA tool, the classifier gets trained, and the actual flagging or identifying cyberbullying instances from the data occurs.

Gathering of Public Textual Posts

The Twitter4J library interfaces with the Twitter API to fetch public Twitter posts. Post authentication using OAuth tokens obtained from Twitter's Application Management portal, the library can use its inherent functions to extract data. As new posts get fetched, they are added to the existing corpus.

f) Preprocessing of Acquired Statements

Acquired statements undergo multiple preprocessing steps before classification:

1. **Cleaning:** Redundant words or characters are automatically removed using Java's String functions as soon as they are added to the corpus.
2. **Tokenization:** The cleaned statements are broken down into individual words.
3. **Bag-of-Words Representation**: Post tokenization, these words are transformed into the Bag-of-Words (BoW) unigram model, which is the format recognized by WEKA. All special characters and numbers are replaced with spaces through the `replaceAll` function.

g) Identification of Cyberbullying Statements

The core of this feature is the trained classifier. It automates distinguishing cyberbullying statements from non-cyberbullying ones in real-time. The classifier processes statements gathered via Twitter4J after they have been added to the corpus and preprocessed.

h) Flagging of Cyberbullying Statements

After classification, each statement gets tagged with one of the three labels: Cyberbullying ("C"), Not Cyberbullying ("NC"), or Ambiguous Cyberbullying ("AC").

i) Reporting of Cyberbullying Statements

All identified cyberbullying statements are displayed in a structured tabular format. Each entry provides detailed information about the tweet, including the poster's username and the timestamp. While "NC" labelled statements are omitted, those marked as "AC" are considered for further refinement of the application's detection capability.

In essence, this methodology paints a comprehensive picture of how cyberbullying detection can be automated using machine learning techniques, specifically SVM. The system seamlessly integrates data acquisition from Twitter, preprocessing, and classification to provide a real-time solution to cyberbullying.

IV. RESULTS AND DISCUSSION

1) Baseline Results

In the first experiment, the SVM model was tested against a dataset of 500 instances for each run. The primary focus was to understand the relationship between the volume of training data and the model's performance. The results in the table suggest a positive correlation between the two: as the volume of training data increases, both accuracy and kappa statistics saw marginal improvements. This demonstrates the importance of having a larger dataset for training purposes.

Training data %	Testing data %	Accuracy	Kappa Statistics
60	40	45.882	0.091
70	30	47.333	0.114
80	20	56	0.218
90	10	51	0.133

However, the Kappa statistics' highest value was 0.2312, achieved during the third run. This indicates a fair agreement between the human annotators and the model. Interestingly, increasing the dataset size only sometimes guarantees a better Kappa score. There can be biases on the annotator's side, influencing the agreement score.

2) Percentage Split

This experiment aimed to determine the optimal ratio of data splitting into training and testing subsets. Based on the results from Table 5.11, an 80/20 split (80% data for training and 20% for testing) is the most appropriate for this dataset.

# of Training data	# of Testing data	Accuracy	Kappa Statistics
200	500	49.6	0.16
500	500	52.6	0.22
700	500	55.58	0.25
1000	500	57.87	0.23

3) K-Fold Cross Validation

K-fold cross-validation is a robust method for understanding the performance of a model. By partitioning the data into "k" segments or folds, the model is trained "k" times, each time leaving out one of the folds for validation. The results from these multiple rounds give a more comprehensive understanding of the model's capability.

K-Fold	Accuracy	Kappa Statistics
2	57.6	0.190
3	57.6	0.200
4	58.2	0.208
5	58.0	0.209
6	58.1	0.208
7	58.8	0.227
8	56.9	0.207
9	58.7	0.208
10	57.9	0.209

The table suggests that dividing the dataset into ten folds yielded the highest accuracy and kappa score. This means that, for this dataset, 10-fold cross-validation is optimal for evaluating the SVM model's performance.

4) Discussion

The three experiments provide insights into various aspects of the SVM classifier:

1. **Volume of Training Data**: More data generally improves the performance, but it does not guarantee better agreement between human annotators and the model.

2. **Data Splitting**: An 80/20 split was the most suitable for this dataset. Such insights are essential because the right split can significantly influence the model's performance.
3. **Cross-Validation**: Using 10-fold cross-validation provides a more rigorous and reliable evaluation of the model's performance for this data.

These results underscore the importance of proper data preparation and evaluation techniques in machine learning projects. The right choices can lead to more accurate and reliable models, while poor choices can mislead and result in suboptimal models.

5) Comparison of Machine Learning Algorithms

In this study, SVM was compared against 11 other machine learning algorithms to determine which was best suited to classify cyberbullying instances. The overall goal was to identify not only the highest accuracy but also to assess the models based on other metrics for a holistic comparison.

Algorithm	Accuracy	Kappa Statistics	Precision	Recall	F-Measure	MCC
SVM	57.95	0.20	0.54	0.60	0.55	0.22
Naïve Bayes	46.3	0.13	0.52	0.49	0.48	0.14
J48	53.8	0.17	0.51	0.52	0.52	0.17
ZeroR	57.8	0	0.32	0.41	0.42	0.14
Decision Stump	56.9	0	0.33	0.42	0.42	0
Random Tree	49.55	0.11	0.48	0.48	0.49	0
Random Forest	61	0.17	0.56	0.53	0.52	0.10
RepTree	55.9	0.10	0.48	0.49	0.50	0.20
Hoeffding Tree	55.8	0	0.32	0.44	0.40	0.11
Decision Table	58.8	0.11	0.54	0.50	0.50	0
JRip	57.9	0.06	0.48	0.46	0.47	0.1
OneR	55	0.05	0.48	0.46	0.47	0.09

- Accuracy: RandomForest and Decision Table achieved the highest accuracy scores of 61% and 58.8%, respectively. In contrast, SVM managed an accuracy of 57.95%.
- Kappa Statistics: This metric evaluated the accuracy of the classification algorithms by comparing observed accuracy with expected accuracy. SVM had the top score of 0.2094.
- Precision and Recall: RandomForest displayed the highest precision (0.560) and recall (0.610). SVM followed with precision and recall values of 0.540 and 0.580, respectively.
- F-measure: An average of precision and recall, the F-measure of SVM was the highest at 0.553.
- Matthews Correlation Coefficient (MCC): SVM also achieved the top MCC value, a balanced measure of true and false positives and negatives, with a score of 0.223.

The time taken to construct each model was another performance indicator. Among all the algorithms, ZeroR was the quickest, taking only 0.02 seconds. However, its predictability power must be improved, primarily as a benchmark. Any other machine learning algorithm tested on the same dataset should ideally have a higher accuracy than ZeroR.

Algorithm	Time seconds
SVM	47.5
Naïve Bayes	4.9
J48	61.8
ZeroR	0.02
Decision Stump	2.7
Random Tree	2.9
Random Forest	40.2
RepTree	14
Hoeffding Tree	17.1
Decision Table	628
JRip	48.2
OneR	1.5

In machine learning, especially in sensitive areas like cyberbullying detection, it is not only about achieving the highest accuracy. Several metrics, such as kappa statistics, precision, recall, F-measure, and MCC, are crucial to understanding the model's performance. In this study, while RandomForest and Decision Table scored high in accuracy, SVM stood out in kappa statistics, F-measure, and MCC. This highlights the importance of considering multiple metrics for a comprehensive evaluation. The challenges faced in language evolution emphasize the need for regularly updated datasets and the importance of cultural and linguistic understanding in developing cyberbullying detection tools.

6) Imbalanced Dataset and Its Implications

Class imbalance is a common issue in many real-world classification problems. When dealing with imbalanced datasets, the challenges introduced can severely skew the performance and evaluations of machine learning algorithms.

The Problem:

Most traditional machine learning algorithms are designed to assume equal distribution among classes. When this assumption is not met, the model can be biased towards the majority class, leading to misleading results. For instance, in the described dataset, "Non-Cyberbullying" accounts for 49%, whereas "Cyberbullying" accounts for 34%, and "Ambiguous Cyberbullying" stands at 18%. The consequence is that models trained on such a dataset might have a bias toward predicting "Non-Cyberbullying" instances, as the model will try to optimize its performance based on the majority class.

7) Challenges with Traditional Metrics:

- **Accuracy:** In the case of imbalanced datasets, a high accuracy might not indicate a well-performing model. For instance, if a model were to predict only the "Non-Cyberbullying" class for all instances, it could still achieve a 49% accuracy, which is misleading.
- **Precision and Recall**: While they provide a more nuanced view of performance than accuracy, these metrics alone do not capture the whole story. Precision tells us about the accuracy of optimistic predictions but neglects the true negatives. Meanwhile, recall focuses solely on the positives and ignores the nuances of the other class predictions.
- **F-measure**: While it is a harmonic mean of precision and recall and provides a balance between the two, it still might need to be more comprehensive for imbalanced datasets.

8) A Solution: Matthews Correlation Coefficient (MCC):

The MCC is a more robust metric for binary classification problems with imbalanced datasets. It returns a value between -1 and 1:

- +1 represents a perfect prediction.
- 0 represents a prediction no better than random.
- -1 indicates complete disagreement between prediction and observation.

MCC takes into account true and false positives and negatives. Hence, it provides a balanced view of the classifier's performance across all classes. A high MCC score means that the classifier has balanced performance across both the majority and minority classes.

V. CONCLUSION AND FUTURE WORKS

As the digital era evolves, social media has become an integral part of global communication. This increased connectivity, while fostering a sense of global community, has also exacerbated the issue of cyberbullying on a worldwide scale. The reliance on users to report malicious activities or harmful posts makes it challenging for platforms to promptly address cyberbullying, primarily due to the vast volume of data and the hesitancy of some users to report such incidents.

In light of this, implementing intelligent systems to automate cyberbullying detection is essential to ensure that social media remains a safe environment for all. While previous studies primarily focused on achieving the highest accuracy, they often overlooked the importance of post-detection strategies to address the identified issues. This paper has attempted to bridge that gap by presenting an approach that not only detects harmful messages efficiently but also offers mechanisms for timely intervention by platform administrators.

Our methodology commenced with data collection from major platforms like Facebook, YouTube, and Twitter. After preprocessing the data, it was represented using the Bag-of-Words (BoW) model. Our comparative analysis of machine learning algorithms, considering a plethora of performance metrics, underscored that while Random Forest models showcased impressive accuracy, precision, and recall metrics, the Support Vector Machine (SVM) was superior in dealing with imbalanced datasets, as evidenced by its higher kappa, F-Measure, and MCC scores. Thus, SVM emerged as the most optimal algorithm for our classification task.

Looking ahead, we aim to refine our system, Quickgarde, in various ways:

1. **Data Expansion**: By integrating more data, we aim to be more inclusive of linguistic variations globally. This will enhance the classifier's capability to detect cyberbullying instances in diverse languages and dialects.
2. **Performance Evaluation**: Future iterations will explore alternative performance metrics like ROC Area to ensure a comprehensive evaluation of our SVM classifier.
3. **Integration with Other Data Mining Techniques**: Exploring the compatibility of Quickgarde with other data mining techniques, such as sentiment analysis, will be crucial. This will not only offer avenues for system improvement but also provide a multi-faceted approach to cyberbullying detection.

In conclusion, as the world becomes increasingly interconnected via digital platforms, it is paramount to ensure that these spaces are free from harm. Automated cyberbullying detection systems like Quickgarde are essential tools in this endeavour, and continuous research and refinement in this domain will significantly impact the global digital community's well-being.

ACKNOWLEDGEMENT

This paper delves into the pressing issue of cyberbullying detection on prominent social media platforms. Focusing on accurate detection and post-detection strategies underscores the importance of automated interventions. Through rigorous analysis, the study identifies SVM as the most effective algorithm, especially when handling imbalanced datasets. The research also highlights potential future enhancements, such as expanding data sources, considering alternative performance metrics, and integrating other data mining techniques.

REFERENCES

[1] D. Karthik, R. Roi i L. Henry, "Modeling the detection of textual cyberbullying," 2011 International Conference on Weblogs and Social Media, ICWSM Workshop, Barcelona, Catalonia, Spain, 2011.

[2] V. H. Cynthia, L. Els, V. Ben, M. Julie, D. Bart, D. P. Guy, D. Walter i H. Veronique, "Detection and Fine-Grained Classification of Cyberbullying Events," you *Proceedings of the International Conference Recent Advances in Natural Language Processing*, Hissar, Bulgaria, 2015.

[3] M. Dadvar, D. Trieschnigg, R. Ordelman i F. de Jong, "Improving Cyberbullying Detection with User Context," *European Conference on Information Retrieval*, 2013.

[4] T. K. H. Chan, C. M. K. Cheung I Z. W. Y. Lee, "Cyberbullying on social networking sites: A literature review and future research directions.," u *Information & Management, Vol 58(2), Mar 2021, Article 103411.*, Durham, United Kingdom, 2021.

[5] J. Aleksandar, "Project files and documentation of Research and Model Testing Sinergija 2023," 27 10 2023. Available: https://github.com/salecivija/sinergija-2023.