

bradscholars

Cyberbullying detection in Urdu language using machine learning

Item Type	Conference paper
Authors	Khan, Sara;Qureshi, Amna
Citation	Khan S and Qureshi A (2022) Cyberbullying detection in Urdu language using machine learning. From: 2022 International Conference on Emerging Trends in Electrical, Control, and Telecommunication Engineering (EECTE). 2-4 Dec 2022, Lahore, Pakistan.
DOI	https://doi.org/10.1109/EECTE55893.2022.10007379
Publisher	IEEE
Rights	© 2022 IEEE. Reproduced in accordance with the publisher's self-archiving policy.
Download date	2026-06-08 17:57:29
Link to Item	http://hdl.handle.net/10454/19312

Cyberbullying Detection in Urdu Language Using Machine Learning

Sara Khan and Amna Qureshi
Department of Computer Science,
University of Bradford, Bradford, UK.
E-mail: s.khan335, a.qureshi19@bradford.ac.uk

Abstract—Cyberbullying has become a significant problem with the surge in the use of social media. The most basic way to prevent cyberbullying on these social media platforms is to identify and remove offensive comments. However, it is hard for humans to read and remove all the comments manually. Current research work focuses on using machine learning to detect and eliminate cyberbullying. Although most of the work has been conducted on English texts to detect cyberbullying, limited to no work can be found in Urdu. This paper aims to detect cyberbullying from the users' comments posted in Urdu on Twitter using machine learning and Natural Language Processing (NLP) techniques. To the best of our knowledge, cyberbullying detection on Urdu text comments has not been performed due to the lack of a publicly available standard Urdu dataset. In this paper, we created a dataset of offensive user-generated Urdu comments from Twitter. The comments in the dataset are classified into five categories. n -gram techniques are used to extract features at character and word levels. Various supervised machine-learning techniques are applied to the dataset to detect cyberbullying. Evaluation metrics such as precision, recall, accuracy and F1 scores are used to analyse the performance of machine learning techniques.

Index Terms—Cyberbullying; Machine Learning; Natural Language Processing; Twitter

I. INTRODUCTION

The number of social media platforms and freedom of expression on these platforms has increased significantly. Users use these platforms to express their feelings, whether positive or negative. Negative comments are often offensive and categorised as offensive language, which can be of multiple forms, such as hate speech, aggressive content, and harmful content, among others. Hate speech is legislated as a crime because it can provoke a large number of communities. Social media platforms, Twitter, Instagram and Facebook, have placed anti-hate speech policies in place, e.g., Facebook uses Artificial Intelligence (AI) models to detect hate speech content. However, considering the sheer volume of the data being sent on these social media platforms, detecting and removing all offensive content in different languages is difficult, e.g., a recent study in Kenya suggested that on submission of advertisements by Global Witness [1], a non-profit group, the AI models of Facebook failed to detect violent hate speech present in those ads.

In recent years, the automatic detection of offensive language on social media platforms has become a hot research topic. Several machine learning (ML)-based techniques have

been applied to the text of various languages. However, the main focus of the research is on resource-rich languages like English. In contrast, resource-poor languages like Urdu have not gained much attention from the research community due to the lack of standard and annotated datasets. The authors in [2] concluded that several attempts were made to generate an Urdu dataset; however, most of these datasets are either not publicly available or annotated, which foiled the development of hate speech detection techniques for Urdu.

Urdu is Pakistan's national language [3] and is the fifth most spoken language in the world. It is spoken in many other countries, such as the UK, the US, Canada and the Middle East. The Urdu language has more alphabet than the English language. Unlike English, generating the Urdu alphabet digitally is quite difficult. The Urdu alphabet cannot be generated using an English keyboard; it requires an Urdu keyboard [4]. Due to such challenges, users of the social media platforms, such as Twitter, Facebook, etc., prefer using Roman Urdu or English to express their opinions. Pakistan alone has over 44 million social network users who communicate in Roman Urdu. However, some users, e.g., news channels, politicians, etc., prefer to express their feelings in Urdu instead of Roman Urdu.

Several studies based on ML techniques have been conducted to detect hate speech content in English, and Roman Urdu [5] on social media texts. However, in the literature, little to no work has been done on detecting offensive language from Twitter comments in Urdu. Also, no standard Urdu dataset is publicly available for offensive text detection.

Contributions and Plan of the Paper: This paper proposes a cyberbullying detection model based on the combined use of NLP and ML methods to classify and detect Urdu hate speech. Since the availability of publicly published annotated datasets for Urdu is scarce, one of the significant contributions of this research is the development of a dataset containing user-generated comments posted in Urdu on Twitter. The generated dataset is annotated manually into five different categories. The comments are classified into four classes using eight supervised ML algorithms. Also, the performance of these classifiers is evaluated and compared in terms of accuracy, precision, recall and F1 score.

The rest of the paper is organised as follows: Section II presents a literature review, where all previous work is analysed. Section III discusses the proposed cyberbullying de-

tection models. Section IV presents the performance analysis of the experimental results. Finally, Section V presents this work's conclusions and possible future research directions.

II. RELATED WORK

Authors proposed an identification system in [6] to prevent cyberbullying on Twitter. Two ML algorithms, Support Vector Machines (SVM) and Logistic Regressions (LR), were used. Feature extraction techniques, Term Frequency Inverse Document Frequency (TF-IDF), and n -gram were used. Experimental results showed that SVM achieved the accuracy and F1 score of 75.17% and 75%, respectively.

In [7], the authors proposed a cyberbullying detection of comments posted in English. TF-IDF was used as a feature representation technique with ML algorithms, namely SVM and Naïve Bayes. Experimental results showed that SVM had a higher accuracy of 71.25% and the Naïve Bayes (NB) had a 52.70% accuracy.

The authors in [8] implemented ML algorithms to detect hate speech in English tweets and adopted five classifiers: SVM, LR, multinomial NB, Random Forest (RF) and Stochastic Gradient Descent (GSD). In this work, the word TFIDF has been used. Experimental results indicated that LR performed best by achieving 91% precision, 94% recall and 93% F1 score.

In [9], the authors performed cyberbullying detection of Arabic comments using ML techniques. The dataset was divided into two categories: offensive and non-offensive. Three non-ensemble ML models (Decision Tree (DT), LR and SVM) and three ensemble ML models (Bagging, AdaBoost and RF) were trained. Experimental results showed that among ensemble ML techniques, Bagging performed the best with 88% of the F1 Score.

Hate speech recognition was performed using supervised ML techniques in [10]. A dataset of 5000 tweets in Roman Urdu was developed and annotated using three categories: simple-complex, offensive-hate speech and neutral-hostile. Regarding accuracy, LR performed well in distinguishing between offensive-hate speech and neutral-hostile.

In [11], nineteen classification algorithms were used to detect bullying in Turkish on social media platforms. The different evaluation metrics were used to evaluate the performance of each algorithm. The comparative results showed that the Light Gradient Boosting (LGB) achieved an accuracy rate of 90.949% with 90.949% of the F1 score value.

León-Paredes et al. [12] collected Spanish tweets and applied ML algorithms and NLP to develop a cyberbullying prevention system. Experimental results showed that NB, SVM, and LR achieved the maximum accuracy of 93%.

All of the work mentioned above is in languages (e.g., English, Arabic, Spanish, Roman Urdu, etc.) other than Urdu. These research works used various classifiers and pre-processing techniques to detect cyberbullying. The Urdu language has diversity and variation, which makes it quite challenging to identify offensive comments. Moreover, it is difficult to make a machine learn Urdu. This gap in the

literature review motivated a need to develop a mechanism for detecting cyberbullying in Urdu text using ML and pre-processing techniques.

III. METHODOLOGY

In this section, the proposed methodology is discussed in detail.

A. Data Collection Phase

In this work, a hate-speech dataset is developed using a data scraping library, sncscrape [13], to scrap comments in Urdu from Twitter. In advanced search, the language was set to Urdu, and the user accounts were set to be from Pakistan. To make the dataset effective and make classification more discrete, tweets were collected over a period of 20 days. A total of 7625 tweets were scrapped in this period. Fig. 1 illustrates tweets containing offensive Urdu language.



Fig. 1. Examples of Urdu comments on Twitter

B. Data Labelling and Validation

The comments were manually annotated and labelled by two people whose native language was Urdu. The generated dataset is unique in itself as it identifies words, phrases, and sentences in Urdu and divides them into five distinct classes: non-offensive (labelled as 0), aggressive/sexual abuse/general (labelled as 1), disruptive - any comment related to blood, catastrophic events, death, animal abuse etc. (labelled as 2), appearance - body shaming/racial abuse (labelled as 3), and political abuse (labelled as 4). Fig. 2 shows the distribution of 7625 into five classes. Figure 8 shows that 68.9% of the tweets were labelled as non-offensive, while 31.1% were labelled as offensive.

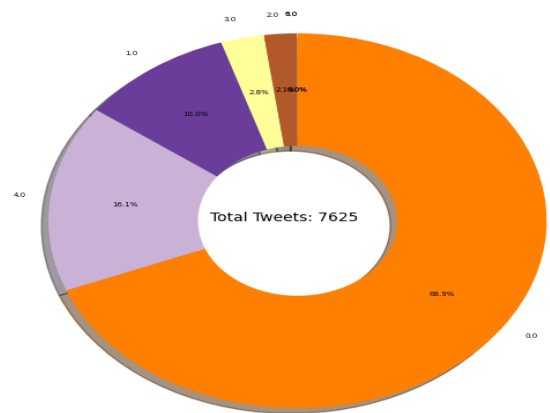


Fig. 2. Complete distribution of Tweets' labels

C. Data Pre-Processing

The tweets scrapped from Twitter contained a lot of irrelevant information, which made it difficult for the machine to predict accurately. To make the data usable for implementation, the pre-processing step was necessary to remove irrelevant information from tweets. In addition, symbols, emoticons, mentions, URLs, hashtags, special symbols, repetitive words/comments, etc., were removed from the tweets. Fig. 3 shows some comments containing irrelevant information (e.g., emoticons, URLs, etc.).

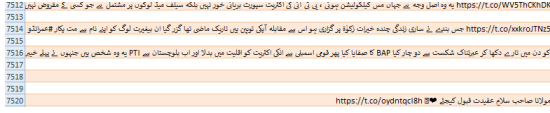


Fig. 3. An example of unwanted data in the dataset

1) **Tokenization:** In this step, the cleaned tweets were tokenized into separate tokens. Tokenization was performed using an Urduhack library [14] to prepare the dataset for ML algorithms. Fig. 4 illustrates an example of tokenization.

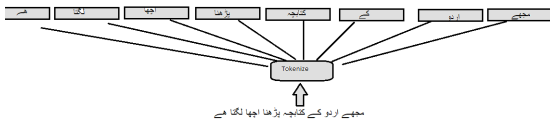


Fig. 4. An example of tokenization

2) **Lemmatization:** Another pre-processing step performed on the data is lemmatization, where the individual word is reduced to its base form. It is mainly performed on inflectional forms of each token and reduces them to a common base form. Fig. 5 shows an example of lemmatization.

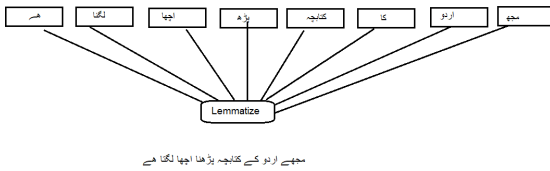


Fig. 5. An example of lemmatization

D. Feature Extraction

Text representation is a necessary step for any text classification process. In this step, the text is extracted, parsed, pre-processed and converted into numbers (called feature vectors) for its learning and prediction. Two feature extraction methods, namely, Bag of Words (BOW) and Term Frequency - Inverse Document Frequency (TF-IDF), were used for digitizing the text for classification. In BOW, every word is given equal importance, while in TF-IDF, the words that occur more frequently are given more importance as they are more useful for classification.

To extract both BOW and TF-IDF features, n -gram is used, which is a useful technique that assigns a probability value to a

word or a sequence of words from the text. The classifiers use the assigned probability value to classify the text. Similarly, character n -gram is used that represents a text as a sequence of characters. In character n -gram, n implies the number of contiguous characters instead of words. This study used uni-gram, bi-gram, and tri-gram to extract features from tweets.

E. Machine Learning Techniques

This section briefly introduces eight supervised ML techniques used in this work to detect cyberbullying on tweets posted in Urdu.

- **XGBoost:** It is a decision-tree-based ensemble ML algorithm that offers efficient memory usage and fast learning. It is the leading ML algorithm for classification problems.
- **Extra Tree Classifier:** It is a type of ensemble ML technique that randomizes certain decisions and subsets of data to minimize over-learning and overfitting.
- **K-Nearest Neighbours:** This ML algorithm is used to classify the data points by looking at what is the majority in its closest neighbours.
- **Logistic Regression:** It is a frequently utilized classification algorithm in ML that is used to describe the dependent data and explain the relationship between one or more existing independent variables.
- **Random Forest:** Random forest algorithm builds a forest of random trees. It functions by building decision trees on different samples and determines the outcome using either majority voting or averages.
- **Linear Support Vector Classification (SVC):** This type of algorithm works well for simple classification tasks. Classification is performed using straight lines, which shows that data can be separated linearly.
- **Decision Tree:** It is a widely used ML algorithm that makes decisions on the basis of a set of rules. It uses a tree-like structure and all possible combinations to solve a particular problem.
- **Multinomial NB:** Naïve Bayes algorithm (NB) is mostly used in NLP problems. Multinomial NB is a specialized version of NB and is used to classify the data into multi-class categories.

IV. RESULTS AND ANALYSIS

This section presents the experimental setup and analyses the performance of the eight selected ML classifiers using two text representation techniques.

A. Experimental Setup

The experiments were performed using the newly built dataset of Urdu tweets. The proposed detection method was implemented using the Python programming language. Several Python libraries (Pandas, SKLearn, Numpy, Matplotlib) were utilized for implementing the detection model. Urduhack, an NLP library, was used to perform data pre-processing like tokenization and lemmatization. The experimental dataset was divided into 70% training and 30% test sets. Different evaluation metrics, accuracy, precision, recall and F1 score,

were computed from a confusion matrix (CM) to evaluate the performance of the selected classifiers. These metrics were obtained from the four elements of the CM: true positive, false positive, true negative and false negative.

B. Word n -Gram Models

This section presents the performance analysis of eight classifiers on six types of n -gram models in terms of training time (the time the classifier consumes to train the model), accuracy (proportion of accurate predictions relative to the total number of predictions), precision (proportion of correctly detected predictions), recall (proportion of predicted attack occurrences relative to all attack instances reported) and F1 score (performance of the model as expressed by the harmonic mean of precision and recall).

TABLE I
TRAINING TIME OF ML ALGORITHMS FOR WORD BOW AND TF-IDF n -GRAM MODELS

Algorithm	Training Time					
	BOW Uni-Gram	BOW Bi-Gram	BOW Tri-Gram	TF-IDF Uni-Gram	TF-IDF Bi-Gram	TF-IDF Tri-Gram
Multinomial NB	0.00	0.00	0.01	0.00	0.00	0.00
Decision Tree	0.56	2.01	2.01	0.66	0.58	0.56
Linear SVC	0.17	0.02	0.02	0.41	0.16	0.16
Random Forest	2.43	3.19	3.19	2.72	2.54	2.34
Logistic Regression	1.00	1.81	1.81	1.21	1.00	0.96
K -Nearest Neighbour	0.00	0.00	0.00	0.00	0.00	0.00
Extra Tree Classifier	4.21	6.13	6.13	5.12	4.32	4.33
XGBoost	1.25	1.75	1.75	1.90	1.22	1.25

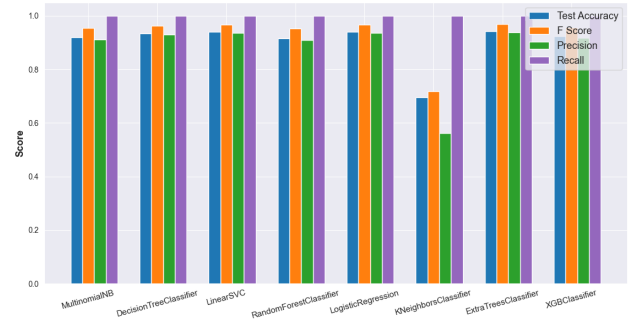
Table I presents the training time taken by each ML model for word BOW and TF-IDF n -gram models. The selection of an appropriate classifier is based on training a model in a reasonable amount of time. A classifier with somewhat lower accuracy but reduced training time would be preferred to algorithms achieving high accuracy with increased training time. Table I shows that the Multinomial NB and K -Nearest Neighbour have the best training time. In contrast, the Extra Tree classifier has the worst training time in all text feature representation techniques.

Table II and Fig. 6 present the performance of ML algorithms for BOW and TF-IDF uni-gram models in terms of the evaluation metrics. It can be seen that the classifiers achieved

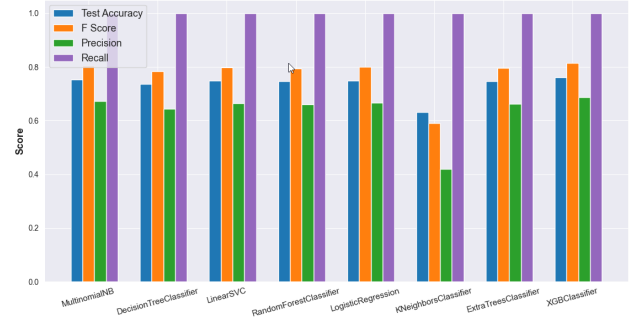
TABLE II
PERFORMANCE OF ML ALGORITHMS FOR WORD BOW AND TF-IDF UNI-GRAM MODELS

	Algorithm	Accuracy	Recall	Precision	F1 Score
	BOW Uni-gram	Multinomial NB	0.918768	1.0	0.911585
Decision Tree		0.939828	1.0	0.935976	0.966929
Linear SVC		0.939828	1.0	0.935976	0.966929
Random Forest		0.908587	1.0	0.887195	0.940226
Logistic Regression		0.939828	1.0	0.935976	0.966929
K -Nearest Neighbour		0.694915	1.0	0.560976	0.718750
Extra Tree Classifier		0.934473	1.0	0.929878	0.963665
XGBoost		0.923944	1.0	0.917683	0.957075
TF-IDF Uni-gram	Algorithm	Accuracy	Recall	Precision	F1 Score
	Multinomial NB	0.753247	1.0	0.672414	0.804124
	Decision Tree	0.735557	1.0	0.640485	0.780849
	Linear SVC	0.748566	1.0	0.664112	0.798158
	Random Forest	0.745714	1.0	0.659004	0.794457
	Logistic Regression	0.749641	1.0	0.666028	0.799540
	K -Nearest Neighbour	0.632472	1.0	0.418902	0.590459
	Extra Tree Classifier	0.747494	1.0	0.662197	0.796773
XGBoost	0.761673	1.0	0.687101	0.814534	

maximum performance using the BOW uni-gram model. For



(a) Evaluation results of ML models for BOW uni-gram



(b) Evaluation results of ML models for TF-IDF uni-gram

Fig. 6. Performance analysis of ML algorithms for word BOW and TF-IDF uni-gram models

BoW uni-gram, it is clear that LR achieved higher performance than the other classifiers. XGBoost classifier achieved better accuracy and F1 score for TF-IDF uni-gram in comparison to the other classifiers. For both BOW and TF-IDF uni-gram, the worst classification result was obtained with the K Nearest Neighbour algorithm.

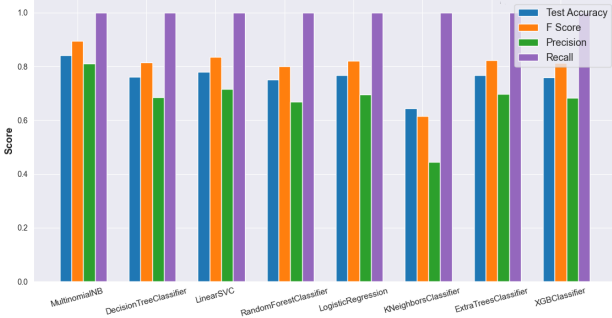
Table III and Fig. 7 present the performance of ML algorithms for BOW and TF-IDF bi-gram models. Similar to uni-gram, all classifiers showed high accuracy rates for the BOW bi-gram model. It is evident from Table III that the highest accuracy was achieved using Multinomial NB. The precision, recall, and F1 score were the highest for Multinomial NB as well. For TF-IDF bi-gram model, all classifiers except K Nearest Neighbour exhibited better accuracy, precision and F1 score values.

Table IV and Fig. 8 present the performance of ML algorithms for BOW and TF-IDF tri-gram models. It is evident from Table IV that all classifiers achieved high accuracy rates for the TF-IDF tri-gram model. For the BOW tri-gram model, the highest accuracy was achieved using Multinomial NB. The precision, recall, and F1 score were also the highest for Multinomial NB. All classifiers except K Nearest Neighbour exhibited better accuracy, precision and F1 score values for the TF-IDF tri-gram model.

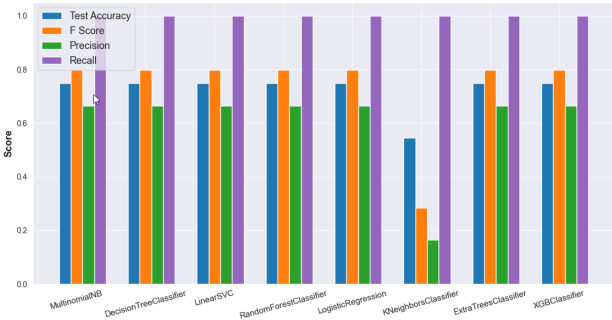
From Tables II, III and IV, it can be observed that the BOW uni-gram model is the best n -gram than other text representation features because all the classifiers achieved maximum performance in terms of accuracy, precision, and F1 score.

TABLE III
PERFORMANCE OF ML ALGORITHMS FOR WORD BOW AND TF-IDF BI-GRAM MODELS

	Algorithm	Accuracy	Recall	Precision	F1 Score
	BOW Bi-gram	Multinomial NB	0.841026	1.0	0.810976
Decision Tree		0.764569	1.0	0.692073	0.818018
Linear SVC		0.779097	1.0	0.716463	0.834813
Random Forest		0.742081	1.0	0.652439	0.789668
Logistic Regression		0.766355	1.0	0.695122	0.820144
K-Nearest Neighbour		0.643137	1.0	0.445122	0.616034
Extra Tree Classifier		0.766355	1.0	0.695122	0.820144
XGBoost		0.759259	1.0	0.682927	0.811594
TF-IDF Bi-gram	Algorithm	Accuracy	Recall	Precision	F1 Score
	Multinomial NB	0.748566	1.0	0.664112	0.798158
	Decision Tree	0.748566	1.0	0.664112	0.798158
	Linear SVC	0.748566	1.0	0.664112	0.798158
	Random Forest	0.748566	1.0	0.664112	0.798158
	Logistic Regression	0.748566	1.0	0.664112	0.798158
	K-Nearest Neighbour	0.545075	1.0	0.165390	0.283836
	Extra Tree Classifier	0.748566	1.0	0.664112	0.798158
XGBoost	0.748566	1.0	0.664112	0.798158	



(a) Evaluation results of ML models for BOW bi-gram



(b) Evaluation results of ML models for TF-IDF bi-gram

Fig. 7. Performance analysis of ML algorithms for word BOW and TF-IDF bi-gram models

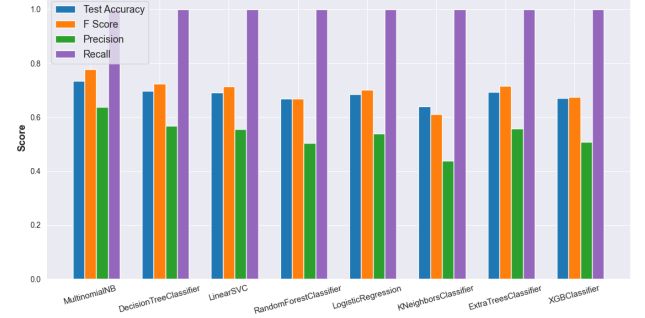
C. Character n -Gram Models

Table V presents the training time taken by each ML model for character TF-IDF n -gram. It can be observed from Table V that the Multinomial NB and K -Nearest Neighbour classifiers have the best training time (0.00s), while the Extra Tree classifier has the worst training time for both character bi- and tri-gram TF-IDF models.

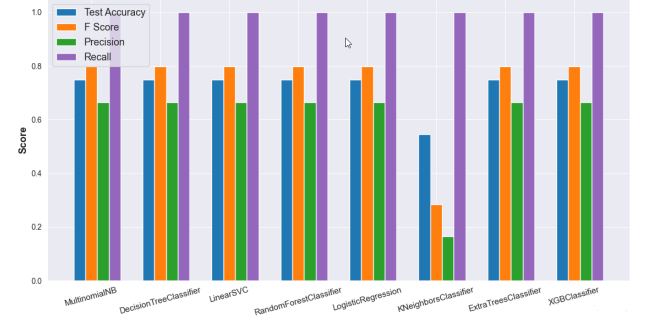
Tables VI and Fig. 9 present the performance of ML algorithms for character TF-IDF n -gram. For the character bi-gram TF-IDF model, LR showed better performance than other models and achieved 74.8% accuracy and 79.8% F1 score. For the character TF-IDF tri-gram model, the XGBoost classifier exhibited better performance by achieving 75.1% accuracy, 66.8% precision and 80.1% F1 score value.

TABLE IV
PERFORMANCE OF ML ALGORITHMS FOR WORD BOW AND TF-IDF TRI-GRAM

	Algorithm	Accuracy	Recall	Precision	F1 Score
	BOW Tri-gram	Multinomial NB	0.733781	1.0	0.637195
Decision Tree		0.697872	1.0	0.567073	0.723735
Linear SVC		0.691983	1.0	0.554878	0.713725
Random Forest		0.672131	1.0	0.512195	0.677419
Logistic Regression		0.684760	1.0	0.539634	0.700990
K-Nearest Neighbour		0.640625	1.0	0.439024	0.610169
Extra Tree Classifier		0.696391	1.0	0.564024	0.721248
XGBoost		0.670757	1.0	0.509146	0.674747
TF-IDF Tri-gram	Algorithm	Accuracy	Recall	Precision	F1 Score
	Multinomial NB	0.745359	1.0	0.658365	0.793993
	Decision Tree	0.742884	1.0	0.653895	0.790734
	Linear SVC	0.747494	1.0	0.662197	0.796773
	Random Forest	0.748208	1.0	0.663474	0.797697
	Logistic Regression	0.748566	1.0	0.664112	0.798158
	K-Nearest Neighbour	0.599311	1.0	0.331418	0.497842
	Extra Tree Classifier	0.746781	1.0	0.660920	0.795848
XGBoost	0.747851	1.0	0.662835	0.797235	



(a) Evaluation results of ML models for BOW tri-gram



(b) Evaluation results of ML models for TF-IDF tri-gram

Fig. 8. Performance analysis of ML algorithms for word BOW and TF-IDF tri-gram models

From tables and figures, it can be observed that for word n -gram, the performance of the BOW uni-gram is the best n -gram than other word and character n -gram models on the generated Urdu dataset. In terms of ML classifiers, among the selected eight classifiers, LR, Multinomial NB and Extra Tree classifiers showed good performance in detecting offensive tweets from the dataset. In terms of time to build models, LR and Multinomial models had minimum training time, but the Extra Tree classifier took longer time to build both word and character n -gram models.

V. CONCLUSION

Considering the importance of detecting hate speech on social media platforms, in this research work, we proposed a method for classifying and detecting comments posted in Urdu

TABLE V
TRAINING TIME OF ML ALGORITHMS FOR CHARACTER TF-IDF n -GRAM MODELS

Algorithm	Training Time	
	TF-IDF Bi-Gram	TF-IDF Tri-Gram
	Multinomial NB	0.00
Decision Tree	0.56	0.56
Linear SVC	0.16	0.18
Random Forest	2.44	2.45
Logistic Regression	0.96	1.00
K -Nearest Neighbour	0.00	0.00
Extra Tree Classifier	4.34	4.25
XGBoost	1.25	1.22

TABLE VI
PERFORMANCE OF ML ALGORITHMS FOR CHARACTER TF-IDF BI- AND TRI-GRAM

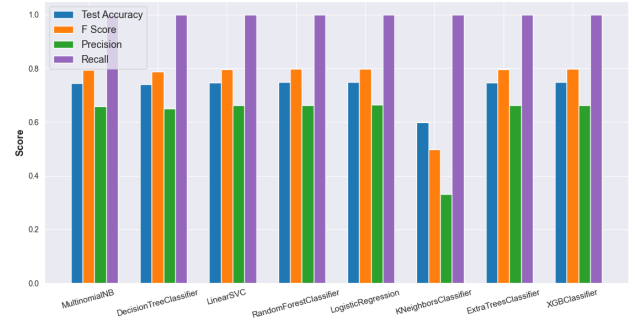
Char TF-IDF Bi-gram	Algorithm	Accuracy	Recall	Precision	F1 Score
	Multinomial NB	0.745359	1.0	0.658365	0.793993
	Decision Tree	0.742884	1.0	0.653895	0.790734
	Linear SVC	0.747494	1.0	0.662197	0.796773
	Random Forest	0.748208	1.0	0.663474	0.797697
	Logistic Regression	0.748566	1.0	0.664112	0.798158
	K -Nearest Neighbour	0.599311	1.0	0.331418	0.497842
	Extra Tree Classifier	0.746781	1.0	0.660920	0.795848
	XGBoost	0.747851	1.0	0.662835	0.797235
	Char TF-IDF Tri-gram	Algorithm	Accuracy	Recall	Precision
Multinomial NB		0.749282	1.0	0.665390	0.799080
Decision Tree		0.741126	1.0	0.650702	0.788395
Linear SVC		0.748566	1.0	0.664112	0.798158
Random Forest		0.749282	1.0	0.665390	0.799080
Logistic Regression		0.748566	1.0	0.664112	0.798158
K -Nearest Neighbour		0.602076	1.0	0.339080	0.506438
Extra Tree Classifier		0.747851	1.0	0.662835	0.797235
XGBoost		0.751079	1.0	0.668582	0.801378

on Twitter by considering two text features, BOW and TF-IDF. A significant contribution of this work was the generation of a unique Urdu dataset, which was annotated into one of the following five classes: non-offensive, aggressive/sexual abuse/general, disruptive, appearance and political abuse. Eight supervised ML algorithms were selected in this research work, and each classifier was performed with the combination of different text feature vectors and n -gram. The performance of different algorithms was compared in terms of training time, accuracy, precision, recall value and F1 score.

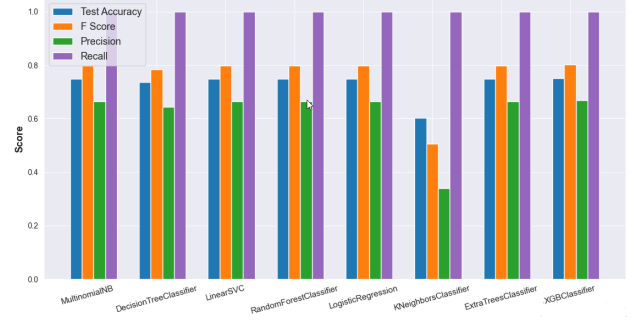
Future work includes the implementation of an auto-correction feature and analysis of sentiments in the Urdu text.

REFERENCES

- [1] Facebook unable to detect hate speech weeks away from tight Kenyan election, [Online]. Available: <https://www.globalwitness.org/en/campaigns/digitalthreats/hate-speech-kenyan-election/>.
- [2] Javed Ashraf, Naveed Iqbal, N. Sarfaraz Khattak, and A. Mohsin Zaidi, *Speaker Independent Urdu Speech Recognition using HMM*, in the Proceedings of Natural Language Processing and Information Systems (NLDB). 2010.
- [3] *Population Census*, [Online]. Available: <https://www.pbs.gov.pk/content/populationcensus/>.
- [4] S.M.Lodhi and M.A.Matin, *Urdu Character Recognition using Fourier Descriptors for Optical Networks*, In the Proceedings of SPIE Photonic Devices and Algorithms for Computing VII. 2005.
- [5] Fahad Rasheed, Mehmood Anwar, and Imran Khan, *Detecting Cyberbullying in Roman Urdu Language using Natural Language Processing Techniques*, PakJET, vol. 5, pp. 198203, 2022.
- [6] Andrea Perera and Pumudu Fernando, *Accurate Cyberbullying Detection and Prevention on Social Media*, Procedia Computer Science, vol. 181, pp. 605611, 2021.



(a) Evaluation results of ML models for character TF-IDF bi-gram



(b) Evaluation results of ML models for character TF-IDF tri-gram

Fig. 9. Performance analysis of ML algorithms for character TF-IDF bi- and tri-gram models

- [7] Rahul Ramesh Dalvi, Sudhanshu Baliram Chavan, and Aparna Halbe, *Detecting a Twitter Cyberbullying using Machine Learning*, in the Proceedings of 4th International Conference on Intelligent Computing and Control Systems. 2020.
- [8] Rashi Shah, Srushti Aparajit, Riddhi Chopdekar, and Rupali Patil, *Machine Learning-based Approach for Detection of Cyberbullying Tweets*, International Journal of Computer Applications, vol. 175, pp. 5156, 2020.
- [9] Fatemah Husain, *Arabic Offensive Language Detection using Machine Learning and Ensemble Machine Learning Approaches*, CoRR, vol. abs/2005.08946, 2020. [Online]. Available: <https://arxiv.org/abs/2005.08946>.
- [10] M. Moin Khan, Khurram Shahzad, and M. Kamran Malik, *Hate Speech Detection in Roman Urdu*, CoRR, vol. abs/2108.02830, 2021. [Online]. Available: <https://arxiv.org/abs/2108.02830>.
- [11] Emre Cihan Ates, Erkan Bostanci, and Mehmet Serdar Güzel, *Comparative Performance of Machine Learning Algorithms in Cyberbullying Detection: Using Turkish Language Pre-processing Techniques*, CoRR, vol. abs/2101.12718, 2021. [Online]. Available: <https://arxiv.org/abs/2101.12718>.
- [12] Gabriel A. León-Paredes et al., *Presumptive Detection of Cyberbullying on Twitter through Natural Language Processing and Machine Learning in the Spanish Language*, in the Proceedings of IEEE CHILEAN Conference on Electrical, Electronics Engineering, Information and Communication Technologies. 2019.
- [13] JustAnotherArchivist, *snsrape*, [Online]. Available: <https://github.com/JustAnotherArchivist/snsrape#readme>. 2019.
- [14] Ikram Ali and Imgbot, *Urduhack*, [Online]. Available: <https://github.com/urduhack/urduhack.com>, 2020.