IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received August 23, 2021, accepted September 6, 2021, date of publication September 14, 2021,
date of current version September 22, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3112500

Threatening Language Detection and Target Identification in Urdu Tweets

MAAZ AMJAD[1], NOMAN ASHRAF[1], ALISA ZHILA[2], GRIGORI SIDOROV[1], ARKAITZ ZUBIAGA[3], AND ALEXANDER GELBUKH[1]

[1]Centro de Investigación en Computación (CIC), Instituto Politécnico Nacional, Mexico City 07738, Mexico
[2]San Francisco, CA, USA
[3]School of Electronic Engineering and Computer Science, Queen Mary University of London, London E1 4NS, U.K.

Corresponding author: Alexander Gelbukh (gelbukh@gelbukh.com)

This work was supported in part by the Mexican Government through Consejo Nacional de Ciencia y Tecnología (CONACYT), Mexico, under Grant A1-S-47854; and in part by the Secretaría de Investigación y Posgrado of the Instituto Politécnico Nacional, Mexico, under Grant 20211784, Grant 20211884, and Grant 20211178.

ABSTRACT Automatic detection of threatening language is an important task, however, most of the existing studies focused on English as the target language, with limited work on low-resource languages. In this paper, we introduce and release a new dataset for threatening language detection in Urdu tweets to further research in this language. The proposed dataset contains 3,564 tweets manually annotated by human experts as either threatening or non-threatening. The threatening tweets are further classified by the target into one of two types: threatening to an individual person or threatening to a group. This research follows a two-step approach: (i) classify a given tweet as threatening or non-threatening and (ii) classify whether a threatening tweet is used to threaten an individual or a group. We compare three forms of text representation: two count-based, where the text is represented using either character *n*-gram counts or word *n*-gram counts as feature vectors and the third text representation is based on fastText pre-trained word embeddings for Urdu. We perform several experiments using machine learning and deep learning classifiers and our study shows that an MLP classifier with the combination of word *n*-gram features outperformed other classifiers in detecting threatening tweets. Further, an SVM classifier using fastText pre-trained word embedding obtained the best results for the target identification task.

INDEX TERMS Threatening language detection, threat target identification, annotated dataset, Urdu language.

I. INTRODUCTION

The emergence of the Internet and communication technology has enabled online social networks to become a significant part of our daily lives, as the number of social media users is growing exponentially. For example, StatInvestor[1] reported that the number of social media users has tripled from 0.97 billion to 3.02 billion from 2010 to 2021. Recent statistics published by Statista[2] showed that Twitter has more than 353 million monthly active users who post more than 200 billion tweets per year. Twitter is one of the most popular social media platforms, which is used to read and share short texts with a maximum length of 280 characters per tweet. Platforms like Twitter welcome a diverse set of people from different ethnic, cultural, linguistic, and religious groups [1]. While censorship of free expression in online content on social media platforms such as Twitter restricts freedom of speech [2]. Cyber offenders have been using Twitter as a new medium to commit various forms of online crimes such as phishing, spamming, malware spreading, and cyber-bullying [3]–[6]. Moreover, the controversial content brings more challenging issues such as incitement to self-harm or sexual predating. Likewise, this can induce threats against groups of victims, gender-based violence, and physical violence [7]. For example, in the GamerGate[3] scandal the Twitter

The associate editor coordinating the review of this manuscript and approving it for publication was Sergio Consoli.

[1]https://statinvestor.com/data/22389/number-of-social-media-users-worldwide/

[2]https://www.statista.com/statistics/282087/number-of-monthly-active-twitter-users/

[3]https://www.businessinsider.com/gamergate-death-threats-2014-10?r=MX&IR=T

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

platform was used to send rape and death threats to women by video-game lovers.

Among other malpractices, some users manipulate the Twitter platform to threaten other people and to promote violence by posting threatening content (i.e. content expressing an intent to cause harm to others). This has led to a growing body of research investigating the spread of threatening content in social media, among others by examining threatening language and by attempting to detect this type of content [8]–[10]. Given the distress this can cause in online users, furthering research in automatic threatening language identification is of utmost importance to tackle this problem at the scale of a large social media platform like Twitter.

While technology for automatic threatening language detection is still in its infancy, automatic detection mechanisms for threatening language on Twitter have gone beyond only English language [11]–[15]. Multiple studies investigated automatic threatening language detection in various languages such as Bengali, German, Dutch, Italian, Indonesian, and Arabic [7], [16]–[20]. These studies examined the linguistic aspects and linguistic resources for automatic threatening language detection. However, to the best of our knowledge, there has been no work on threatening language detection in Urdu, and no relevant dataset of Twitter postings has been previously collected. This paper presents the first such effort specifically targeting tweets in Urdu, a low-resource language.

The key contributions of this study are as follows:

- The first threatening language detection dataset in Urdu, with data extracted from Twitter and manually annotated by experts following a set of guidelines;
- A hierarchical annotation scheme, making the dataset suitable for two distinct tasks: threatening language detection and threat target identification;
- Baseline results of threatening language detection and threat target identification using several machine learning models (LR, MLP, Ada-Boost, RF, and SVM; see Section IV-B) as well as deep learning models (1D-CNN and LSTM; see Section IV-C) and three text representations (word *n*-grams, character *n*-grams, and fastText pre-trained word embeddings).

Our dataset provides a benchmark enabling further research in threatening language detection in Urdu.

The rest of the paper is structured as follows: Section II provides the background details of the study. Section III discusses the data collection and data annotation procedure. Section IV shows the benchmark results. Section V discusses results and analysis. Finally, the conclusion of the study is presented in Section VI.

II. RELATED WORK

In this section, we first discuss how threatening language is defined and then discuss existing research in threatening language detection.

A. DEFINING AND CHARACTERIZING THREATENING LANGUAGE

When it comes to the definition of what is considered a threat, the Twitter platform defines it as ''a statement of an intent to kill or inflict serious physical harm on a specific person or group of people''.[4] Threat is characterized as an expression of a clear intent to cause bodily or other form of harm. For example, in ''shut your mouth or, you will see red'' the word ''red'' can be perceived as threat to cause injury to someone or in the worst-case scenario bloodshed or killing. Therefore, such expressions can be considered a vile aspersion. To take this into account, Twitter has introduced a number of initiatives to mitigate the spread of threatening content on its platform. For example, they may use the timeout feature i.e., suspending an account for several hours if a their behaviour was flagged as potentially abusive.[5] However, more effort is clearly needed, in particular, in detecting abusive content in various languages since the presence of abusive and threatening language in social media is still pervasive.[6]

B. DATASETS AND APPROACHES TO THREATENING LANGUAGE DETECTION

Detecting threatening language is a challenging task, particularly when it comes to distinguishing it from other types of offensive content or even benevolent content where there might be an overlapping use of vocabulary. It is common to see the use of negative or profane terms for amusement and sarcasm. Some examples of such widely known terms used to threaten are ''Blood, kill, murder, death, and stab''. To detect threatening language, the natural language processing community has been focusing on online platforms like YouTube, Twitter, Facebook, Instagram, and blogs [2], [20], [22]–[26], [29], [34]–[37]. Multiple studies relied on chi-square feature selection as well as used lexicon based techniques to study automatic threatening language detection. In addition, many researchers used character *n*-grams [16], [21], [23], [24], [27], [33], word *n*-grams and their combinations for threatening language detection [1], [16], [20], [24]–[26], [28], [31], [34].

A few studies investigated automatic detection of threatening language in multilingual datasets and used machine learning techniques. For example, several studies have utilized Support Vector Machines (SVM) and Logistic Regression (LR) classifiers to detect threatening speech in tweets, articles, blogs, Facebook and Reddit [16], [20], [23], [25]–[28], [34], [37]. Similarly, Naive Bayes (NB) was used to classify threatening comments in user generated YouTube comments and NewsGroups while Decision Tree (DT) was used to detect threatening language in Turkish tweets and Instagram content [1], [24], [27], [30], [37]. Moreover, only one study used k-nearest neighbors (KNN) on the dataset of

[4]https://help.twitter.com/en/rules-and-policies/violent-threats-glorification

[5]https://social.techcrunch.com/2017/02/16/twitter-starts-putting-abusers-in-time-out

[6]https://blog.hubspot.com/marketing/twitter-harassment-cyberbullying

TABLE 1. Summary of social media datasets to detect threatening language.

Platform	Language	Features	Methods
Twitter [22]	English	char n-grams (1−4)	LR
NewsGroup [23]	English	complement NB	Multinomial Decision Table
Twitter [24]	English	BoW, char n-grams	SVM, LR, CNN
YouTube [25]	English	BoW, word n-grams (2, 3, 5)	SVM, NB
YouTube [15]	English	BoW, GloVe, fastText	1D-CNN, LSTM, and BiLSTM
Twitter [26]	English	unigram	SVM, CNN, BiLSTM
Twitter [27]	English	word n-grams (1−8)	SVM
Twitter [6]	English	Latent Dirichlet Allocation (LDA)	LR
Online Comments [1]	English	word n-grams, char n-grams	NB, SVM
Twitter [28]	English	word n-gram (3−8), char n-grams (1−3)	CNN, RNN, RF, NB, SVM
Twitter [29]	English	word n-grams	SVM (linear, polynomial, radial)
Twitter, Articles [30]	English	abusive and non-abusive word list	k-means
Twitter, Blogs [31]	English, Portuguese	hateword2vec, hatedoc2vec, unigrams	NB, SVM
YouTube [21]	Arabic	word n-gram	SVM
Twitter [32]	Spanish	word n-grams, char n-grams	LR
Twitter [33]	Indonesian	Latent Dirichlet Allocation (LDA)	–
Twitter, Facebook, Reddit [34]	Danish, English	char n-grams	LR, BiLSTM
Twitter [17]	German	Wikipedia embedding	CNN
Twitter [19]	Italian	BERT tokens	AlBERTo
Blogs [35]	Japanese	word n-grams (1-5)	SVM
Facebook [36]	Bengla	word n-grams (1-3)	MNB, SVM, CNN, LSTM
Twitter, Instagram [36]	Turkish	–	MNB, SVM, DT (C4.5), KNN

tweets and Instagram posts or comments written in Turkish language [37].

Recent studies also explored deep learning based models for the detection of threatening language. For example, Convolutional Neural Networks (CNNs) were used to classify potentially threatening content posted online on Twitter and Facebook in German and English tweets [1], [6], [16], [17], [21]–[30], [35]. These studies showed that CNNs outperform other neural networks based models. Moreover, a set of studies on detecting threatening language utilized Recurrent Neural Networks (RNN) and LSTM models to classify threatening content on Facebook in Bengali language [16], [25], [27], [34]. BiLSTM and Graph Convolutional Network were applied on English Twitter content [21], [23], [25], [33].

A few researchers explored threatening language detection in languages: English, Italian, German, Turkish, Bengali, Japanese, Danish, Indonesian, Arabic, Portuguese, Spanish [1], [6], [16], [17], [19], [21]–[34], [37].

We summarize existing literature in threatening language detection in Table 1. As can be seen by looking at the ''Language'' column, no prior work has been conducted for the Urdu language, to the best of our knowledge. From our exploration of the literature and its lack of application to the Urdu language, we emphasize the following gaps in the research which we tackle in our work:

1) lack of threatening language datasets in the Urdu language,
2) lack of appropriate feature engineering: Most of the studies used lexical features such as character n-grams and word n-grams,
3) lack of comparison between classifiers: Most of the studies used either only machine learning ML) or only deep learning (DL) techniques, while no comparison was done between ML and DP models to define the best classifier for this task.

Our research contributes in these three directions by creating and releasing a dataset in Urdu, comparing the performance of lexical and embedding features, and comparing machine learning and deep learning models to assess their performance.

III. BUILDING THE DATASET

In this section, we describe the data acquisition and annotation process that we followed to create the first Urdu dataset for threatening language detection, as well as present the statistics of the resulting dataset.

A. DATA COLLECTION

We used the Tweepy[7] library to collect tweets through the Twitter Developer Application Programming Interface (API).[8]

To build a dictionary of seed words, we started with a manually crafted list of words that are used to threaten in Urdu, which was subsequently used to search for other words used in Twitter postings through a snowballing process. After searching for tweets with these keywords, we manually inspected the tweets and identified other words and phrases used for threatening objectives. Eventually, this process was followed until we ended up with enough words and phrases that are used to threat individuals. Some keywords contained only one word while some keywords contained two or more than two words. It is important to mention that the selection of most frequent words does not depend on how many times

[7]http://www.tweepy.org

[8]https://developer.twitter.com/en/docs/tweets/search/api-reference/get-search-tweets

a threatening word is used in a tweet. If a word is appeared only a single time in a tweet to treat someone, this would be included in the dictionary. For instance, Urdu words such as ''بولتی بند'' and ''تباہ'' are examples of threatening words in our dictionary. This dictionary is publicly available for research purposes.[9]

Using this dictionary, we collected tweets containing any of these keywords for a 20 months period ranging from January 1st, 2018 to August 31st, 2019. The reason for choosing this time interval was the general elections being held in Pakistan in July 2018. Typically, during the election season, people tend to express emotions by showing support to political parties and express antagonistic behavior to supporters of opposing parties.

The collection process led to 55,600 tweets containing the seed words. Next, the data was cleaned by removing non-Urdu and duplicate tweets. Since Urdu has its roots[10] in the Arabic, Persian, and Turkish languages, querying tweets on single seed words retrieved posts in other languages that we had to discard. To discard the non-Urdu tweets, there is no public library that can differentiate Arabic, Turkish, Persian, and Urdu words. For instance, the ''ہلال'' word is same in all other languages. Therefore, tweets containing words from other languages were removed manually. Tables 2 and 3 show examples from our dataset.

While annotating threatening tweets, since we only used threatening words to collect the data, the majority of the tweets have abusive, hateful, sexist and threatening words. For example ''Shut your mouth mother fucker, otherwise I will kill you'', which shows both threatening and abusive words. Due to the different nature of the tweets, we created two different datasets: (i) abusive dataset;[11] containing 3,500 tweets, 1,750 of them are abusive and 1,750 of them are non-abusive (ii) threatening dataset;[12] containing 9,918 tweets, 1,782 threatening tweets and remaining tweets are non-threatening. This original dataset is imbalanced and it is selected in Fire 2021 competition.[13] For the experiments described in this paper, we used a balanced dataset with 3,564 tweets: 1,782 threatening and 1,782 non-threatening that are randomly selected from the imbalance dataset.

B. DATA ANNOTATION GUIDELINES

We recruited paid annotators for dataset annotation, through the use of a web-based annotation system to maximize the efficiency of the annotation process and reduce annotation errors. We recruited annotators from Pakistan all of whom satisfy the following criteria: (i) are familiar with Twitter; (ii) native speakers of Urdu; (iii) aged 20–35 years; (iv) detached from any political party or organization, and (v) have prior experience of annotating data. Among all annotators who expressed an interest, we selected three annotators, whose educational level was a masters degree or above. Instructions with the task definition (which we reproduce below) and examples were provided to the annotators.

In our dataset, a hierarchical annotation schema is used and we divided our dataset into two tasks to distinguish between whether the language is threatening or non-threatening (Task 1), and its target (Task 2) [38].

1) TASK 1: THREATENING LANGUAGE DETECTION

- Threatening: A tweet posted by a user to kill or inflict serious physical harm on a specific person or group of people.
- Non-threatening: A tweet posted by a user for other purposes. For example to advertise, phishing attempts, sarcasm, joke, or other suspicious nature.

2) TASK 2: THREAT TARGET IDENTIFICATION

Task 2 categorizes the targets of threats:

- Individual (IND): Tweets in which threatening language is used for an individual; it can be a person, a name or an unnamed participant.
- Group (GRP): Tweets in which the target is a group of people based on their religion, gender, sexual orientation, political affiliation or other common characteristic.

C. INTER-ANNOTATOR AGREEMENT

We computed Inter-Annotator Agreement (IAA) using Cohen's Kappa coefficient [39] as it is a statistic measure to check the reliability between two annotators. The measurement led to an overall Kappa coefficient of 90%.

D. DATASET STATISTICS

After normalization (correcting the encoding of Urdu characters in the unicode range 0600-06FF and fixing the issue of words that are joined together in Urdu), 3,564 tweets were used in threatening language dataset to perform the experiments. Tables 4 and 5 show dataset statistics.

IV. BENCHMARKS

We experiment with a set of classification models both to assess the challenging nature of our dataset and to determine which models can better tackle the task. We address two binary classification tasks: (i) Task 1 is to detect that a tweet is threatening or non-threatening (ii) Task 2 is to classify threatening tweets into individual vs. group threats.

A. EXPERIMENT SETTINGS

In all the experiments, we perform stratified 10-fold cross-validation using machine and deep learning algorithms including Logistic Regression (LR), Multilayer Perceptron (MLP), Ada-Boost, Random Forest (RF), Support Vector Machine (SVM), 1-Dimensional Convolutional Neural Network (1D-CNN), and Long Short-Term Memory (LSTM). These algorithms were selected because they showed

[9] https://github.com/MaazAmjad/Threatening_Dataset.git
[10] https://www.ucl.ac.uk/atlas/urdu/language.html
[11] https://github.com/MaazAmjad/Abusive_dataset.git
[12] https://github.com/MaazAmjad/Threatening_Dataset.git
[13] https://www.urduthreat2021.cicling.org/home

TABLE 2. Examples from the dataset containing tweets from the threatening vs. non-threatening classes.

Tweet (Translation)	Classes
بولتی بند کردوں گا تیری (I will kill you)	1
مجھے کوئی لے جا کر دکھائے تو سہی ٹانگیں توڑ دوں گا (If someone takes me, I will break his legs)	1
انشاءاللہ ان کے دانت توڑ دیں گے ہر محاذ پہ (With Allah will, we will break their teeth on every front)	1
اسٹاک مارکیٹ کا تو بیڑا غرق ہو گیا ہے پوائنٹس کی کمی اربوں کا نقصان (The stock market has sunk, billions of points are lost)	0
اللہ تعالی بے غیرتوں کو تباہ کرے (May Allah destroy the shameless)	0

TABLE 3. Examples from dataset containing tweets from individual vs. group threatening classes.

Tweet (Translation)	Classes
بولتی بند کردوں گا تیری (I will kill you)	1
مجھے کوئی لے جا کر دکھائے تو سہی ٹانگیں توڑ دوں گا (If someone takes me, I will break his legs)	1
انشاءاللہ ان کے دانت توڑ دیں گے ہر محاذ پہ (With Allah will, we will break their teeth on every front)	0

competitive performance for various NLP tasks [40]–[42]. After tokenizing the tweets, numerals in the Eastern Arabic-Indic system were converted to the Western Arabic to normalize the content. We also removed punctuation and stop words, removed characters that were not part of the UTF-8 encoding standard. Two types of count-based features, character *n*-grams and word *n*-grams were extracted using TF-IDF weighting scheme.[14] In addition, we used fastText [43] pre-trained word embeddings for the deep learning experiments.

B. MACHINE LEARNING CLASSIFIERS

We used five machine-language algorithms for both task 1 and task 2: (Logistic Regression (LR), Multilayer Perceptron (MLP), Ada-Boost, Random Forest (RF), Support Vector Machine (SVM).

1) LOGISTIC REGRESSION

The regression models are used as a statistical process to measure the relationship between a dependent variable and one or multiple independent variables [10]. The logistic regression (LR) model is a statistical learning classification technique, which uses features to construct linear model using multinomial logistic regression with ridge estimator [21]. It transforms nominal attributes into numeric attributes as well as replaces the missing attributes.

2) RANDOM FOREST

This is an ensemble and boosting based classifier, which is frequently used to mitigate challenges related to variance and over-fitting [37]. This algorithm is used for classification and regression problems that is based on multiple decision trees [44]. Each decision tree is constructed by using features of a dataset that are some randomly selected. Random forest (RF) consists of a multiple decision trees, which are used for model output generation. The value test determines the factor space and decision tree partition factor space that intent to non-linear classification. The nodes of the tree are determined to maximize the information gain and the most common criteria are used namely "GINI" and "Entropy" [10]. Each tree is used for classification task, and the desired output is performed by aggregating the majority votes of all the trees constructed with the input samples.

3) SUPPORT VECTOR MACHINE

Support Vector Machine (SVM) [44] is a machine learning classifier used for linear, nonlinear classification and regression problems. This algorithm is used to find out decision boundary by the support vectors (optimal hyper-plane) to separate output into different classes (into different data points). This task is done using the kernel trick to separate data points into classes by drawing n-dimensional hyper planes. SVM has been used in threatening language detection [16], [23], [26]. However, it is not recommended to use SVM, especially with spars datasets due to high memory and poor intractability. Previous studies [20], [27], [28] reported that SVM outperformed other classifiers to detect threatening speech in tweets.

4) ADA-BOOST

Adaptive Boosting algorithm [45], known as AdaBoost classifier, is an iterative ensemble method, which is widely used to solve classification and regression tasks. AdaBoost provides better results because it concatenates multiple machine learning classifiers and re-assigned all the weights to the input samples. Particularly, this algorithm assigns higher weights to mis-classified examples to obtain high accuracy strong classifier. Adaboot quickly overfits while dealing with highly noisy dataset. In contrast, this provides better results in case a dataset contains less noise patterns. Finally, if the irrelevant features are used to train the model, this algorithm provides insignificant results [44].

[14] use_idf=True, smooth_idf=True, number of features (Max) and the other than these with default values. https://scikit-learn.org/stable

TABLE 4. Threatening vs. non-threatening classes statistics.

Class	Words	Avg Word	Char	Avg Char	Tweets
Threatening	30518	17.12	134408	75.42	1782
Non-threatening	31135	17.47	140225	78.68	1782
All	61653	34.59	274633	154.1	3564

TABLE 5. Individual vs. group threatening classes statistics.

Class	Words	Avg Word	Char	Avg Char	Tweets
Threat group	24508	18.27	108459	80.87	1341
Threat individual	6010	13.62	25949	58.84	441
All	30518	31.89	134408	139.71	1782

TABLE 6. Deep learning parameters for threatening language detection.

Parameter	1D-CNN	LSTM
Epochs	100	150
Optimizer	Adam	Adam
Loss	mean squared error	mean squared error
Learning Rate	0.0001	0.0001
Regularization	0.001	-
Bias Regularization	0.0001	-
Validation Split	0.1	0.1
Hidden Layer 1 Dimension	16	16
Hidden Layer 1 Activation	linear	tanh
Hidden Layer 1 Dropout	0.2	0.2
Hidden Layer 2 Dimension	32	16
Hidden Layer 2 Activation	linear	tanh
Hidden Layer 2 Dropout	0.2	0.2

TABLE 7. Threatening language detection using word-level features (TFIDF-based).

Feature set	Features	--	Classifiers				
			LR	MLP	AdaBoost	RF	SVM
unigram	7,699	Acc	73.14	67.45	68.54	70.73	73.12
		P	74.85	67.16	71.27	72.47	75.91
		R	69.81	68.35	62.23	67.06	67.84
		F_1	72.20	67.71	66.39	69.62	71.61
bigram	36,012	Acc	70.39	68.29	60.52	68.18	68.79
		P	74.80	68.55	78.39	76.73	79.85
		R	61.61	67.78	29.07	53.42	50.33
		F_1	67.53	68.08	42.33	62.40	61.70
trigram	48,819	Acc	63.80	64.48	55.80	58.53	57.21
		P	75.52	74.70	86.27	81.88	83.94
		R	40.09	44.16	13.91	22.05	17.78
		F_1	52.98	55.26	23.93	34.62	29.30
combination (1-3)-gram	92,530	Acc	73.06	**72.50**	66.75	70.06	72.70
		P	76.19	**72.33**	69.52	78.71	78.19
		R	67.28	**73.28**	60.15	55.21	63.07
		F_1	71.40	**72.74**	64.40	64.83	69.76

5) MULTILAYER PERCEPTRON

A multilayer perceptron (MLP) is a feedforward artificial neural network [46], which is used for solving regression and classification tasks. This algorithm is mostly used for supervised learning tasks, and generates a set of outputs from a set of inputs. The algorithm uses back-propagation for training the model, which contains three layers (i) an input layer, (ii) a hidden layer, and (iii) a fully connected output layer. All

TABLE 8. Threatening language detection using char-level features (TFIDF-based).

Feature set	Features	--	Classifiers				
			LR	MLP	AdaBoost	RF	SVM
3-gram	9,406	Acc	72.47	67.11	66.35	70.31	72.64
		P	73.83	66.88	68.01	69.69	74.37
		R	69.64	67.84	61.95	71.94	69.13
		F_1	71.64	67.34	64.77	70.78	71.61
4-gram	31,280	Acc	73.31	69.47	66.80	71.07	73.00
		P	75.00	69.28	69.32	71.98	76.09
		R	69.97	69.92	60.54	69.18	67.17
		F_1	72.36	69.57	64.54	70.51	71.29
5-gram	67,905	Acc	73.20	70.73	66.80	70.51	72.89
		P	75.96	70.76	72.03	75.41	77.61
		R	67.95	70.87	55.16	60.94	64.53
		F_1	71.69	70.75	62.38	67.36	70.39
6-gram	107,233	Acc	72.39	70.93	65.29	69.36	71.57
		P	76.15	70.77	74.17	78.42	78.31
		R	65.32	71.49	47.08	53.47	59.59
		F_1	70.28	71.07	57.56	63.56	67.68
combination (3-6)-gram	215,824	Acc	**73.62**	71.29	66.46	70.42	73.45
		P	**75.85**	71.38	68.02	71.60	77.36
		R	**69.36**	71.38	62.23	67.78	66.38
		F_1	**72.43**	71.31	64.96	69.61	71.40

TABLE 9. Threatening language detection using pre-trained fastText embedding.

Feature set	Features	--	Classifiers						
			LR	MLP	AdaBoost	RF	SVM	1D-CNN	LSTM
fastText	300	Acc	68.68	68.63	64.59	66.44	70.62	68.15	**68.60**
		P	70.92	67.76	65.02	67.54	72.60	66.11	**66.50**
		R	63.35	71.04	63.13	63.19	66.22	74.69	**75.25**
		F_1	66.89	69.35	64.03	65.25	69.24	70.07	**70.56**

TABLE 10. Threat target identification (individual vs. group) using word-level features (TFIDF-based).

Feature set	Features	--	Classifiers				
			LR	MLP	AdaBoost	RF	SVM
unigram	4,635	Acc	76.99	76.54	77.27	77.94	**79.57**
		P	78.41	53.59	55.60	75.02	**65.12**
		R	08.62	42.19	41.28	17.70	**37.40**
		F_1	15.33	46.14	46.47	28.03	**47.05**
bigram	18,842	Acc	75.58	76.20	75.75	71.89	75.98
		P	40.00	53.59	54.50	45.26	65.47
		R	01.36	28.82	11.11	34.69	04.99
		F_1	02.62	36.91	18.21	36.98	09.16
trigram	24,183	Acc	75.30	74.63	75.59	42.87	75.98
		P	10.00	47.81	52.09	28.04	60.50
		R	00.02	21.79	08.17	8.982	04.09
		F_1	00.44	29.30	13.92	41.87	07.60
combination (1-3)-gram	47,660	Acc	75.75	77.55	77.27	76.99	76.54
		P	50.00	62.81	56.31	80.94	70.83
		R	02.04	22.69	36.26	09.53	07.25
		F_1	03.86	32.56	43.80	16.66	12.97

the data instances are fed as inputs, then dot product of the input instances with the weights are fed to the hidden layer and passed through as activation function. Finally, the output of the activation function is further multiplied (dot product) with weights, which are pushed forward to the output layer that provides the label for the classification task.

TABLE 11. Threat target identification (individual vs. group) using char-level features (TFIDF-based).

Feature set	Features	--	Classifiers				
			LR	MLP	AdaBoost	RF	SVM
3-gram	6,990	Acc	77.83	78.34	**79.52**	78.67	80.30
		P	69.14	58.20	**60.75**	76.28	64.40
		R	14.05	41.74	**47.64**	19.50	44.01
		F_1	23.05	47.93	**53.22**	30.82	51.96
4-gram	20,633	Acc	76.43	78.5	76.82	77.61	78.67
		P	70.00	62.32	53.77	69.28	68.01
		R	04.98	34.03	46.74	15.87	25.40
		F_1	09.16	43.27	48.62	25.62	36.74
5-gram	40,315	Acc	75.87	78.73	76.37	77.33	77.21
		P	50.00	64.48	56.83	70.73	72.68
		R	02.50	30.40	26.09	13.16	11.34
		F_1	04.70	40.89	35.03	21.94	19.20
6-gram	59,728	Acc	75.53	78.84	76.37	77.05	76.43
		P	40.00	67.31	55.41	74.02	72.50
		R	01.13	27.69	14.96	10.67	06.35
		F_1	02.20	38.33	23.25	18.34	11.47
combination (3-6)-gram	127,666	Acc	75.92	79.29	78.11	78.11	78.50
		P	60.00	66.07	56.85	76.74	74.28
		R	02.95	31.98	47.62	15.65	18.60
		F_1	05.52	42.69	51.37	25.63	29.41

TABLE 12. Threat target identification (individual vs. group) using pre-trained fastText embedding.

Feature set	Features	--	Classifiers						
			LR	MLP	AdaBoost	RF	SVM	1D-CNN	LSTM
fastText	300	Acc	77.10	79.12	77.38	77.66	**75.31**	79.24	79.40
		P	83.00	61.34	55.05	63.97	**49.78**	62.25	63.45
		R	08.84	43.55	42.41	21.78	**69.88**	40.16	39.02
		F_1	15.68	50.30	47.61	32.09	**57.84**	48.50	47.96

C. DEEP LEARNING CLASSIFIERS

Neural networks based techniques have been widely used in the threatening language detection task [17], [23], [25], [27]. In this study, we used two neural networks based models, (i) 1-Dimensional Convolutional Neural Network (1D-CNN) and (ii) Long Short-Term Memory Networks (LSTM). Table 6 depicts all the layers information, parameters and their values for threatening language detection and threat target identification.

1) 1-DIMENSIONAL CONVOLUTIONAL NEURAL NETWORK

A Convolutional Neural Network (CNN) is a class of deep neural networks used for image detection and classification [47]. CNNs are regularized versions of multi-layer perceptrons, which consist of multiple layers of neural network. Each hidden layer contains neurons and biases, where each input instance is multiplied (dot product) with weight and fed to the neuron. Each neuron takes the weighted sum of all the fed input instances and add bias to it, which is further pass to an activation function to receive an output of the particular neuron. The model is trained through back-propagation technique, which is a technique to minimize the error by adjusting the weights of all the layers starting from the output layer to the input layer. CNN is also computationally efficient because it offers the possibility to share the parameter of features and reduces the dimensionality.

The architecture of neural network can be used to process data with input shape, such as 1D matrix (text), 2D matrix (image), and 3D matrix (video). CNN has been widely used because it can automatically extract relevant and distinctive features efficiently and provides high accuracy, and computationally efficient as compared with feed-forward networks [47]. Pre-trained fastText embeddings, extracted from Urdu tweets, were used as an input for our 1D-CNN classifier and it was trained on 100 epochs for 10 times. The results were calculated by taking mean accuracy of 10 iterations. For the convolution layer, we set the filter size to 8 and kernal size to 1. Two fully connected layers were used with different neurons and activation function. In addition, dropout is applied to all the layers to avoid overfitting.

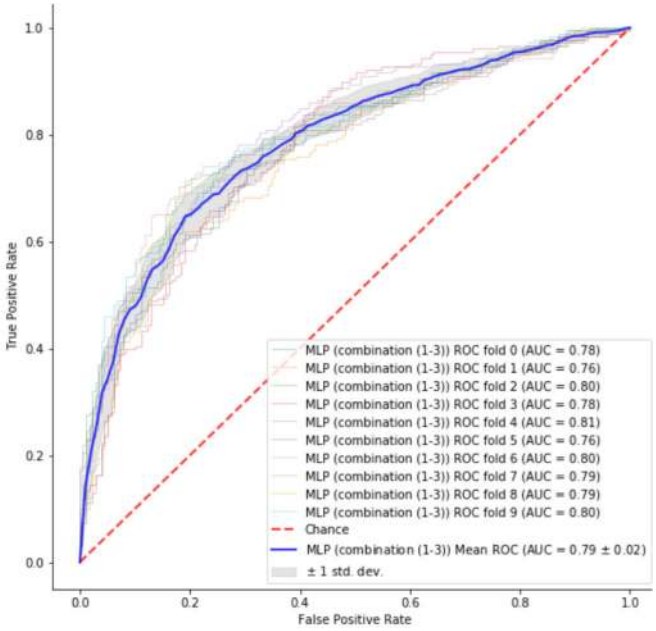

FIGURE 1. ROC curve for best performing model on threatening-language detection.

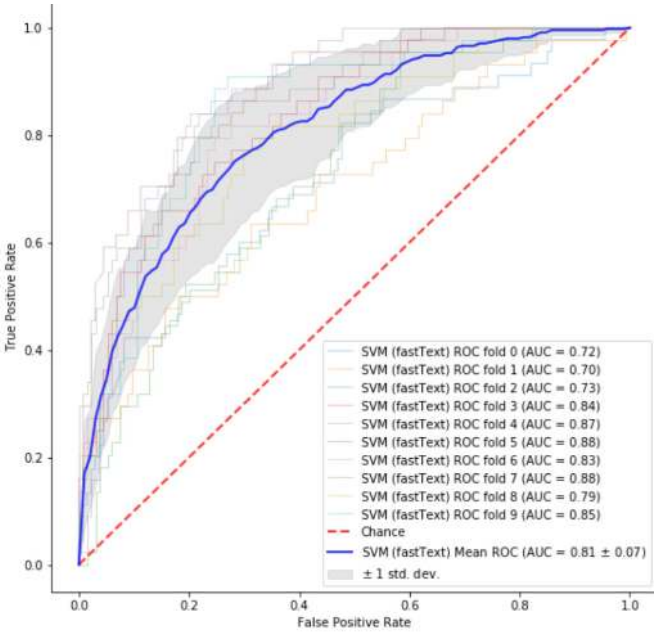

FIGURE 2. ROC curve for best performing model on threat target identification.

2) LONG SHORT-TERM MEMORY NETWORKS

Long Short-Term Memory (LSTM) networks are also a type of deep neural networks, which addressed the challenges related to order dependence in sequence prediction, tasks such as in machine translation, and speech recognition [48]. Moreover, this is a special type of recurrent neural network, which consists of four linear layers (MLP layer) per cell to run at and for each sequence time-step. Multiple studies have utilized Recurrent Neural Networks (RNNs) to detect threatening language [25], [27]. We used 150 epochs for each iteration in tenfold cross-validation. We used two fully dense layers with different neurons and activation functions. Apart from this, dropout is applied to all the layers to avoid over-fitting.

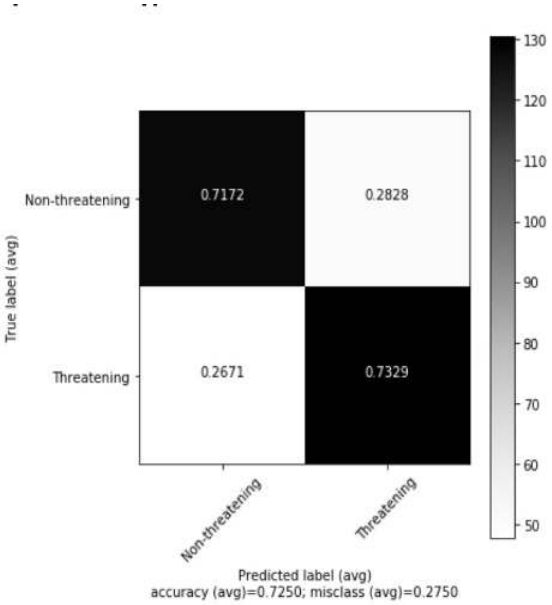

FIGURE 3. Confusion matrix for best performing model on threatening-language detection.

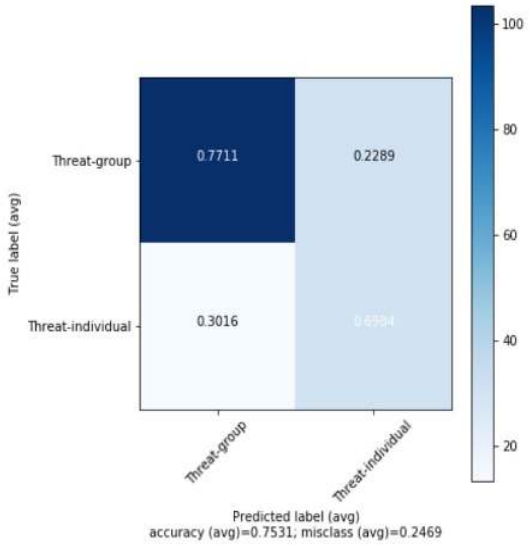

FIGURE 4. Confusion matrix for best performing model on threat target identification.

V. RESULTS AND ANALYSIS

Task 1 results are presented in Tables 7, 8 and 9 while Task 2 results are shown in Tables 10, 11 and 12 for all of the baseline classifiers using three feature representations. The feature column in the tables represents the maximum number of features for both character *n*-grams and word *n*-grams extracted using TF-IDF weighting scheme. In scikit-learn library, we set ''max_features'' parameter to ''None'' to extract maximum number of features. For instance, in Table 7, the number 7,699 in features column represents the maximum number of features obtained for unigram features.

Default parameters were applied in our experiments for all machine learning classifiers while deep leaning parameters are shown in Table 6. Accuracy, Precision, Recall, and F_1 scores are presented for all models: Logistic Regression (LR), Multilayer Perceptron (MLP), AdaBoost, Random Forest (RF), Support Vector Machine (SVM), 1-Dimensional Convolutional neural network (1D-CNN), and Long short-term memory (LSTM).

Task 1 and Task 2 experiments were performed on three text representations: word *n*-gram features, char *n*-gram features, and fastText pre-trained word embedding. In Task 1, word *n*-gram features performed better than char *n*-gram features and fastText embedding for threatening-language detection while in Task 2 fastText embedding yielded the best results for individual versus group threatening classification as compared to word *n*-gram and char *n*-gram features. The highest accuracy of 72.50% and F_1 score of 72.74% were achieved with the MLP using combination of word *n*-gram features for threatening language detection while on Task 2, threatening target identification, we achieved an accuracy of 75.31% and F_1 score of 57.84% with the SVM using fastText pre-trained word embedding.

Figures 1 and 3 show the ROC curve and the confusion matrix for the threatening language detection task using the MLP classifier. Likewise, Figures 2 and 4 show the ROC curve and the confusion matrix for the threat target identification task using SVM.

Overall, char *n*-gram obtain consistent results on Task 1, threatening language detection, for all machine-learning classifiers than word *n*-grams. FastText pre-trained embeddings perform worst on all of these features for threatening language detection. For Task 2, threat target identification, fastText achieves highest result than count based features. Word *n*-gram yielded the worst results in all the experiments while char *n*-gram achieves slightly poor results as compared to fastText embedding.

Notably, fastText did not perform well for Task 1, threatening-language detection. Perhaps this happens due to the limited amount of training data or threatening words could be missed as out-of-vocabulary. Furthermore, fastText performs better with words that are not commonly used as long as they are constructed from previously seen sub-words. During word embedding training, if a word was not seen, its embeddings can be obtained by fragmenting the word into character *n*-grams. Moreover, we foresee that the performance of the deep learning classifiers can be improved with the increase of the dataset size [49].

All in all, our results on both tasks are in line with state-of-the-art work in machine-learning and deep learning for threatening language detection and threat target identification, but also demonstrate that there is still significant room for improvement.

VI. CONCLUSION AND FUTURE WORK

Several studies have investigated automatic threatening language detection in English and other European languages. However, to the best of our knowledge, no study yet has

investigated threatening language detection in Urdu. In this paper, we have presented a new annotated dataset for threatening language detection and threat target identification in the Urdu language which is publicly available for research purposes. In particular, our dataset is balanced, and we selected 3,564 tweets in total, 1,782 threatening and 1,782 non-threatening, on which all annotators agreed. The threatening tweets were further annotated as threats to an individual or group. The experimental results reveal that MLP with the combination of word *n*-gram features outperformed other classifiers in detecting threatening tweets, whereas fastText pre-trained word embedding using SVM obtained the best results for the target identification task. In the future, our plan is to increase the size of the dataset and perform experiments using transformers to improve the results for threatening language detection.

ACKNOWLEDGMENT

The authors utilize the computing resources brought to them by the CONACYT through the Plataforma de Aprendizaje Profundo para Tecnologías del Lenguaje of the Laboratorio de Supercómputo of the INAOE, Mexico.

REFERENCES

[1] Y. Mehdad and J. Tetreault, "Do characters abuse more than words?" in *Proc. 17th Annu. Meeting Special Interest Group Discourse Dialogue*, 2016, pp. 299–303.

[2] V. Balakrishnan, S. Khan, T. Fernandez, and H. R. Arabnia, "Cyberbullying detection on Twitter using big five and dark triad features," *Personality Individual Differences*, vol. 141, pp. 252–257, Apr. 2019.

[3] A. Schmidt and M. Wiegand, "A survey on hate speech detection using natural language processing," in *Proc. 5th Int. Workshop Natural Lang. Process. Social Media*, 2017, pp. 1–10.

[4] P. Badjatiya, S. Gupta, M. Gupta, and V. Varma, "Deep learning for hate speech detection in tweets," in *Proc. 26th Int. Conf. World Wide Web Companion*, 2017, pp. 759–760.

[5] X. Wang, Y. Liu, S. U. N. Chengjie, B. Wang, and X. Wang, "Predicting polarities of tweets by composing word embeddings with long short-term memory," in *Proc. 53rd Annu. Meeting Assoc. Comput. Linguistics*, vol. 1, 2015, pp. 1343–1353.

[6] G. Xiang, B. Fan, L. Wang, J. Hong, and C. Rose, "Detecting offensive tweets via topical feature discovery over a large scale Twitter corpus," in *Proc. 21st ACM Int. Conf. Inf. Knowl. Manage.*, 2012, pp. 1980–1984.

[7] F. Del Vigna, A. Cimino, F. Dell'Orletta, M. Petrocchi, and M. Tesconi, "Hate me, hate me not: Hate speech detection on Facebook," in *Proc. 1st Italian Conf. Cybersecur. (ITASEC)*, 2017, pp. 86–95.

[8] V. Behzadan, C. Aguirre, A. Bose, and W. Hsu, "Corpus and deep learning classifier for collection of cyber threat indicators in Twitter stream," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2018, pp. 5002–5007.

[9] S. R. Vadapalli, G. Hsieh, and K. S. Nauer, "Twitterosint: Automated cybersecurity threat intelligence collection and analysis using Twitter data," in *Proc. Int. Conf. Secur. Manage. (SAM)*, 2018, pp. 220–226.

[10] S. Kok, A. Abdullah, N. Jhanjhi, and M. Supramaniam, "Ransomware, threat and detection techniques: A review," *Int. J. Comput. Sci. Netw. Secur.*, vol. 19, no. 2, p. 136, 2019.

[11] T. Davidson, D. Warmsley, M. Macy, and I. Weber, "Automated hate speech detection and the problem of offensive language," in *Proc. Int. AAAI Conf. Web Social Media*, 2017, vol. 11, no. 1, pp. 512–515.

[12] N. Dionisio, F. Alves, P. M. Ferreira, and A. Bessani, "Cyberthreat detection from Twitter using deep neural networks," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2019, pp. 1–8.

[13] F. Husain, "OSACT4 shared task on offensive language detection: Intensive preprocessing-based approach," 2020, *arXiv:2005.07297*. [Online]. Available: http://arxiv.org/abs/2005.07297

[14] S. Malmasi and M. Zampieri, "Detecting hate speech in social media," 2017, *arXiv:1712.06427*. [Online]. Available: http://arxiv.org/abs/1712.06427

[15] N. Ashraf, R. Mustafa, G. Sidorov, and A. Gelbukh, "Individual vs. Group violent threats classification in online discussions," in *Proc. Companion Proc. Web Conf.*, New York, NY, USA, Apr. 2020, pp. 629–633.

[16] P. Chakraborty and M. H. Seddiqui, "Threat and abusive language detection on social media in Bengali language," in *Proc. 1st Int. Conf. Adv. Sci., Eng. Robot. Technol. (ICASERT)*, May 2019, pp. 1–6.

[17] E. Eder, U. Krieg-Holz, and U. Hahn, "At the lower end of language—Exploring the vulgar and obscene side of German," in *Proc. 3rd Workshop Abusive Lang. Online*, 2019, pp. 119–128.

[18] N. Oostdijk and H. van Halteren, "N-gram-based recognition of threatening tweets," in *Proc. Int. Conf. Intell. Text Process. Comput. Linguistics*. Springer, 2013, pp. 183–196.

[19] M. Polignano, P. Basile, M. De Gemmis, and G. Semeraro, "Hate speech detection through AlBERTo Italian language understanding model," in *Proc. NL4AIAIIA*, 2019, pp. 1–13.

[20] A. Alakrot, L. Murray, and N. S. Nikolov, "Towards accurate detection of offensive language in online communication in Arabic," *Procedia Comput. Sci.*, vol. 142, pp. 315–320, Jan. 2018.

[21] P. Mishra, M. Del Tredici, H. Yannakoudakis, and E. Shutova, "Abusive language detection with graph convolutional networks," 2019, *arXiv:1904.04073*. [Online]. Available: http://arxiv.org/abs/1904.04073

[22] A. H. Razavi, D. Inkpen, S. Uritsky, and S. Matwin, "Offensive language detection using multi-level classification," in *Proc. Canadian Conf. Artif. Intell.* Springer, 2010, pp. 16–27.

[23] J. Ho Park and P. Fung, "One-step and two-step classification for abusive language detection on Twitter," 2017, *arXiv:1706.01206*. [Online]. Available: http://arxiv.org/abs/1706.01206

[24] Y. Chen, Y. Zhou, S. Zhu, and H. Xu, "Detecting offensive language in social media to protect adolescent online safety," in *Proc. Int. Conf. Privacy, Secur., Risk Trust Int. Confernece Social Comput.*, Sep. 2012, pp. 71–80.

[25] M. Zampieri, S. Malmasi, P. Nakov, S. Rosenthal, N. Farra, and R. Kumar, "Predicting the type and target of offensive posts in social media," 2019, *arXiv:1902.09666*. [Online]. Available: http://arxiv.org/abs/1902.09666

[26] P. Rani and A. K. Ojha, "KMI-coling at SemEval-2019 task 6: Exploring N-grams for offensive language detection," in *Proc. 13th Int. Workshop Semantic Eval.*, 2019, pp. 668–671.

[27] Y. Lee, S. Yoon, and K. Jung, "Comparative studies of detecting abusive language on Twitter," 2018, *arXiv:1808.10245*. [Online]. Available: http://arxiv.org/abs/1808.10245

[28] P. Burnap and M. L. Williams, "Us and them: Identifying cyber hate on Twitter across multiple protected characteristics," *EPJ Data Sci.*, vol. 5, pp. 1–15, Oct. 2016.

[29] H.-S. Lee, H.-R. Lee, J.-U. Park, and Y.-S. Han, "An abusive text detection system based on enhanced abusive and non-abusive word lists," *Decis. Support Syst.*, vol. 113, pp. 22–31, Sep. 2018.

[30] R. Pelle, C. Alcántara, and V. P. Moreira, "A classifier ensemble for offensive text detection," in *Proc. 24th Brazilian Symp. Multimedia Web*, Oct. 2018, pp. 237–243.

[31] H. Gómez-Adorno, G. B. Enguix, G. Sierra, O. Sánchez, and D. Quezada, "A machine learning approach for detecting aggressive tweets in Spanish," in *Proc. SEPLN*, 2018, pp. 102–107.

[32] T. Febriana and A. Budiarto, "Twitter dataset for hate speech and cyberbullying detection in Indonesian language," in *Proc. Int. Conf. Inf. Manage. Technol. (ICIMTech)*, Aug. 2019, pp. 379–382.

[33] G. Ingi Sigurbergsson and L. Derczynski, "Offensive language and hate speech detection for Danish," 2019, *arXiv:1908.04531*. [Online]. Available: http://arxiv.org/abs/1908.04531

[34] T. Ishisaka and K. Yamamoto, "Detecting nasty comments from BBS posts," in *Proc. 24th Pacific Asia Conf. Lang., Inf. Comput.*, 2010, pp. 645–652.

[35] M. E. Ptaszynski and F. Masui, *Automatic Cyberbullying Detection: Emerging Research and Opportunities: Emerging Research and Opportunities*. Hershey, PA, USA: IGI Global, 2018.

[36] Y. Zhao and Y. Zhang, "Comparison of decision tree methods for finding active objects," *Adv. Space Res.*, vol. 41, no. 12, pp. 1955–1959, 2008.

[37] S. A. Ozel, E. Sarac, S. Akdemir, and H. Aksu, "Detection of cyberbullying on social media messages in Turkish," in *Proc. Int. Conf. Comput. Sci. Eng. (UBMK)*, Oct. 2017, pp. 366–370.

[38] M. Zampieri, S. Malmasi, P. Nakov, S. Rosenthal, N. Farra, and R. Kumar, "Predicting the type and target of offensive posts in social media," in *Proc. Conf. North*, 2019, pp. 1415–1420.
[39] J. Cohen, "A coefficient of agreement for nominal scales," *Educ. Psychol. Meas.*, vol. 20, no. 1, pp. 37–46, 1960.
[40] R. U. Mustafa, N. Ashraf, F. S. Ahmed, J. Ferzund, B. Shahzad, and A. Gelbukh, "A multiclass depression detection in social media based on sentiment analysis," in *Proc. 17th Int. Conf. Inf. Technol.-New Generat. (ITNG)*. Cham, Switzerland: Springer, 2020, pp. 659–662.
[41] I. Ameer, N. Ashraf, G. Sidorov, and H. Gómez Adorno, "Multi-label emotion classification using content-based features in Twitter," *Computación y Sistemas*, vol. 24, no. 3, pp. 1–15, Sep. 2020.
[42] L. Khan, A. Amjad, N. Ashraf, H.-T. Chang, and A. Gelbukh, "Urdu sentiment analysis with deep learning methods," *IEEE Access*, vol. 9, pp. 97803–97812, 2021.
[43] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *Trans. Assoc. Comput. Linguistics*, vol. 5, pp. 135–146, Dec. 2017.
[44] V. Menger, F. Scheepers, and M. Spruit, "Comparing deep learning and classical machine learning approaches for predicting inpatient violence incidents from clinical text," *Appl. Sci.*, vol. 8, no. 6, p. 981, Jun. 2018.
[45] J. Zhu, H. Zou, S. Rosset, and T. Hastie, "Multi-class AdaBoost," *Statist. Interface*, vol. 2, no. 3, pp. 349–360, 2009.
[46] M. W. Gardner and S. R. Dorling, "Artificial neural networks (the multilayer perceptron)-a review of applications in the atmospheric sciences," *Atmos. Environ.*, vol. 32, nos. 14–15, pp. 2627–2636, Aug. 1998.
[47] K. O'Shea and R. Nash, "An introduction to convolutional neural networks," 2015, *arXiv:1511.08458*. [Online]. Available: http://arxiv.org/abs/1511.08458
[48] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
[49] N. Rusnachenko, R. Bauman, N. Loukachevitch, E. Tutubalina, R. Bauman, and R. Samsung, "Distant supervision for sentiment attitude extraction," in *Proc. Natural Lang. Process. Deep Learn. World*, Oct. 2019, pp. 1022–1030.

MAAZ AMJAD received the master's degree in applied mathematics and informatics from Moscow Institute of Physics and Technology, Russia. He is currently pursuing the Ph.D. degree with the Centro de Investigación en Computación, Instituto Politécnico Nacional (IPN). His research interests include computational linguistics, fake news detection, and deception detection.

NOMAN ASHRAF received the master's degree in computer science from the National University of Computer and Emerging Sciences, Islamabad, Pakistan. He is currently pursuing the Ph.D. degree with the Centro de Investigación en Computación, Instituto Politécnico Nacional (IPN). He worked as a Lecturer with The University of Lahore, Pakistan, from 2017 to 2019. His specialization lies in natural language processing (NLP) and his research interests include contextual abusive language detection, depression detection, and emotion detection.

ALISA ZHILA received the Ph.D. degree in computer science from the Instituto Politécnico Nacional (IPN). She studied at the Natural Language and Text Processing Laboratory, Centro de Investigación en Computación, IPN. She specialized in computational linguistics and natural language processing. More specifically, her research interests include open information extraction from text and its applications to text quality evaluation, particularly, text informativeness.

GRIGORI SIDOROV is currently a Full Professor and a Researcher at the Centro de Investigación en Computación (Center for Computing Research, CIC), which is a part of the Instituto Politécnico Nacional (National Polytechnic Institute, IPN), Mexico City, Mexico. He has coauthored more than 190 scientific publications with an H-index of 27. His scientific interests include computational linguistics, automatic word processing, and application of machine learning methods to natural language processing tasks. Apart from that, he is a Regular Member of the Mexican Academy of Sciences and the National Researcher of Mexico (SNI) level 3 (highest). He is the Editor-in-Chief of the research journal *Computación y Sistemas* (ISI-Thomson Web of Science [SciElo and CORE Collection (emerging sources)], Scopus, DBLP, and Index of Excellence of CONACYT).

ARKAITZ ZUBIAGA is currently a Lecturer with Queen Mary University of London, U.K., where he leads the Social Data Science Lab. His research interests include social data science, interdisciplinary research bridging computational social science, and natural language processing. He is particularly interested in linking online data with events in the real world, among others for tackling problematic issues on the Web and social media that can have a damaging effect on individuals or society at large, such as hate speech, misinformation, inequality, biases, and other forms of online harm. He has published over 100 papers (more than 40 journal articles) in interdisciplinary areas, including social data science, computational social science, and natural language processing. He serves as an academic editor for six journals.

ALEXANDER GELBUKH is currently a Research Professor and the Head of the Natural Language Processing Laboratory, Center for Computing Research, Instituto Politécnico Nacional, Mexico, and an Honorary Professor with Amity University, India. He is the author or the coauthor of more than 500 publications in computational linguistics, natural language processing, and artificial intelligence, recently with a focus on sentiment analysis and opinion mining. He is a member of the Mexican Academy of Sciences. He is a Founding Member of the Mexican Academy of Computing and the National Researcher of Mexico (SNI) at excellence level 3 (highest). He has been the chair or the program committee chair of over 50 international conferences. He is an editor-in-chief, an associate editor, or a member of editorial board for more than 20 international journals.