IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 9 May 2024, accepted 16 July 2024, date of publication 22 July 2024, date of current version 6 August 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3431939

RESEARCH ARTICLE

From Text to Insight: An Integrated CNN-BiLSTM-GRU Model for Arabic Cyberbullying Detection

EMAN-YASER DARAGHMI[1], SAJIDA QADAN[2], YOUSEF-AWWAD DARAGHMI[3], RAMI YOUSUF[3], OMAR CHEIKHROUHOU[4], AND MOHAMMED BAZ[5]

[1]Department of Computer Science, Palestine Technical University—Kadoorie, Tulkarm 00970, Palestine
[2]Faculty of Graduate Studies, Palestine Technical University—Kadoorie, Tulkarm 00970, Palestine
[3]Department of Computer Systems Engineering, Palestine Technical University—Kadoorie, Tulkarm 00970, Palestine
[4]Higher Institute of Computer Science of Mahdia, University of Monastir, Mahdia 5111, Tunisia
[5]Department of Computer Engineering, College of Computers and Information Technology, Taif University, Taif 21944, Saudi Arabia

Corresponding author: Eman-Yaser Daraghmi (e.daraghmi@ptuk.edu.ps)

This work was supported by Taif University, Saudi Arabia, under Project TU-DSSP-2024-143.

ABSTRACT Several research on cyberbullying detection have employed different deep learning and machine learning methodologies to achieve promising outcomes. Nevertheless, most of them have primarily concentrated on using English data for both purposes: training and testing, with only a limited number considering native languages such as Arabic. Thus, there is a critical need to address cyberbullying in its native linguistic context. The dataset utilized in this research has been compiled and sourced from various Kaggle and Github repositories. Six collected benchmark datasets from Facebook, Twitter and Instagram in addition to a developed Arabic cyberbullying lexicon were utilized to evaluate the efficiency of the proposed hybrid model. Prior to classification, data cleaning was carried out to preprocess the text. Moreover, word embedding as a natural language processing method is utilized. Numerous machine learning and deep learning algorithms were assessed, encompassing naïve bayes, support vector machines, k-nearest neighbors, decision trees, random forest, multi-layer perceptron neural networks, convolutional neural networks, recurrent neural networks, bidirectional long short-term memory, long short-term memory, and gated recurrent units, with a meticulous comparative analysis conducted. Given their demonstrated potential, hybrid techniques have emerged as promising model for effectively detecting instances of cyberbullying. Thus, the best performing algorithms is utilized to construct the hybrid model. This research introduces a hybrid deep learning model with stacked word embedding. This model consistently outperforms single models in terms of cyberbullying detection. We extensively investigated the performance of the proposed hybrid model across diverse data contexts. Through thorough study and validation, the proposed hybrid model demonstrates enhanced capabilities in feature extraction and accurate text classification.

INDEX TERMS Cyberbullying detection, deep learning, hybrid model, machine learning, word embedding.

I. INTRODUCTION

In the contemporary digital era, the internet has fundamentally transformed communication, offering a vast array of online platforms and social media applications. With technology advancing rapidly and mobile device usage on the rise, online engagement has escalated. However, this breathtaking rise in online interaction comes with serious risks, most notably cyberbullying. Cyberbullying, which affects both victims and perpetrators, has arisen as a result of social media platforms proliferation. Cyberbullying is evident primarily through text-based interactions and is

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

particularly prevalent on social media platforms, leaving victims emotionally vulnerable without adequate means to address the situation. Intelligent systems and automated detection have the ability to detect cyberbullying on social media platforms [1], [2], [3].

Continued research in this domain has leveraged different Machine Learning (ML) and Deep Learning (DL) methodologies to make promising achievements in detecting and preventing text-based cyberbullying [2], [3], [4], [5], [6], [7], [8], [9], [10], [11]. However, the majority of previous research have predominantly focused on English data for both training and testing, with only a limited number incorporating native languages such as Arabic [1], [2], [3], [4], [5], [6], [7], [8], [12]. Hence, a critical need arises to address cyberbullying within its native linguistic context. By incorporating Arabic data into our proposed learning algorithm, we aspire to create a potent tool capable of effectively detecting cyberbullying instances in real-time communication channels, including forums, social media platforms, and online communities. The integration of Arabic language processing capabilities will enable a more comprehensive approach to cyberbullying detection tailored to the unique linguistic and cultural aspects of Arabic.

The dataset utilized in this research has been compiled and sourced from various Kaggle and Github repositories, including Arabic Levantine Hate Speech dataset, Positive and Negative Tweets, and Arabic Cyberbullying Comments. Six collected benchmark datasets from Facebook, Twitter and Instagram in addition to a developed Arabic cyberbullying lexicon were utilized to evaluate the efficiency of the proposed hybrid model. Since these datasets originate from various sources, their original forms are not directly compatible due to differences in classification labels. To address this, a unified classification technique employing a 0-1 classifier was adopted to expedite the determination of whether a text contains cyberbullying content and simplify the training process for our proposed model.

Before classification, data cleaning was performed to preprocess the text, involving the removal of symbols, numbers, emails, URLs, stopwords, whitespace, punctuation, along with lemmatization, stemming, and single tokens. The resulting cleaned dataset provides a standardized and streamlined base for training the proposed model on Arabic cyberbullying detection. Furthermore, a Natural Language Processing (NLP) technique, namely, word embedding was employed in this research to map phrases and words from a vocabulary to real-number vectors. Thus, assisting tasks, such that word predictions and capturing semantic relationships as these embeddings hold valuable value when dealing with datasets rich in contextual information. The primary advantage of word embeddings is the ability to be generated from vast unannotated corpora, eliminating the need for costly annotation. Pre-trained embeddings could be employed in tasks requiring minimal labeled data. Additionally, stacked embeddings is introduced to combine conventional and contextual string embeddings to achieve optimal results.

Our research involved the assessments of a range of classification algorithms to determine the most effective model for improving the accuracy of cyberbullying classification. We investigated various ML models, including K-Nearest Neighbors (KNN), Naïve Bayes (NB), Decision Tree (DT), Random Forest, Multi-layer Perceptron Neural Network (MLPNN), and Support Vector Machine (SVM). A comprehensive comparison of these algorithms was conducted to evaluate their performance metrics on the dataset.

Expanding the research scope, DL which is a subset of ML known for its superior performance compared to traditional ML or statistical methods was adopted. DL algorithms, with their neuron layers, surpass in text classification tasks, outperforming other methods on academic benchmarks. Five DL techniques were evaluated for cyberbullying classification: Convolutional Neural Network (CNN), recurrent neural networks (RNNs), Long Short-Term Memory (LSTM), Bidirectional LSTM (Bi-LSTM), and Gated Recurrent Unit (GRU). Each of these DL algorithms has its strengths and areas of application. To evaluate their performance metrics on the dataset, a comprehensive evaluation was conducted.

Previous research suggested that hybrid ML and DL techniques have emerged as promising models for detecting cyberbullying. Thus, we employed the best-performing ML and DL algorithms to build the hybrid model. Moreover, this research integrated the proposed hybrid model with stacked word embeddings in order to consistently outperforming single models in terms of cyberbullying detection. The performance of the proposed model was investigated across multiple and diverse datasets.

In summary, this research introduces an "enhanced CNN-BiLSTM-GRU" hybrid DL model with stacked word embeddings for cyberbullying detection. The training data undergoes thorough cleansing and preprocessing before entering the stacked word embeddings. Notably, the hybrid model consistently outperforms single models in terms of its ability to detect cyberbullying instances. Extensive analysis explores how our proposed hybrid model adapts to diverse data variations. The hybrid model itself has undergone rigorous examination and validation, showcasing its heightened capability in feature extraction and precise text classification. Subsequently, this hybrid model is adopted to develop a prototype mobile forensics-based tool for a cyberbullying detection system, geared towards monitoring and mitigating cyberbullying instances across various social media platforms.

The paper is structured into several key sections to comprehensively explore Arabic cyberbullying detection. Section II, Related Work, provides an extensive review of existing literature and methodologies pertinent to the field. Section III, Cyberbullying Detection: Task Definition, precisely defines the criteria and characteristics used to identify cyberbullying in Arabic text. Section IV, Proposed Model, details the development process of a hybrid CNN-BiLSTM-GRU model across several phases. Section V, Results and Discussion, presents the evaluation of the proposed model's effectiveness and compares it with existing approaches, analyzing performance metrics. Finally, Section VI, Conclusion, summarizes the study's findings, underscores contributions, and proposes future research avenues in Arabic cyberbullying detection using advanced machine learning techniques.

II. RELATED WORK

Recently, cyberbullying have emerged as a significant hazard occurring on social media platforms impacting individuals and corporations. Various methods and mechanisms, including Machine Learning (ML) and Deep Learning (DL) models have been developed to detect cyberbullying and thus combating this behavior. Previous research [13], [14] conducted on cyberbullying detection techniques categorizes cyberbullying into different types and highlight their implications for data security and privacy. The authors discussed the various methods for identifying, detecting and preventing cyberbullying, as well as the vulnerabilities associated with each one. This section summarizes existing literature on cyberbullying detection adopting a diverse range of ML and DL methodologies in addition to various feature extraction and word vector techniques.

A study conducted in 2018 by Haidar et al. [15] aims at detecting cyberbullying in Arabic languages. The authors proposed a solution for detecting cyberbullying in Arabic language, utilizing natural language processing (NLP) to identify Arabic language words and ML techniques to detect cyberbullying content. Results indicated that their proposed system was able to detect 30.4% of cyberbullying content in Arabic language. However, in comparison with previous research on identifying English cyberbullying, research results were not perfect. The authors in [16] conducted new research with the aim of detecting hoax content on social media using Bi-LSTM and RNN. Four deep learning techniques were utilized in their study to construct four systems, including: Bi-directional Long Short-Term Memory (Bi-LSTM), Recurrent Neural Network (RNN), hybrid RNN-Bi-LSTM, and hybrid Bi-LSTM-RNN. For feature extraction, Term Frequency-Inverse Document Frequency (TF-IDF) and Global Vectors for feature expansion (GloVe) were adopted.

Another research was conducted [17] on detecting Arabic cyberbullying Tweets Using machine learning. The authors utilized an Arabic dataset containing 30,000 Twitter comments to train the SVM model. Due to the popularity of Twitter as a social media platform for acquiring text data to categorize cyberbullying remarks, the SVM model was assessed on a separate Twitter dataset. It was illustrated that Farasa NLTK's performance of SVM with the TF-IDF vectorizer achieved the best categorization of cyberbullying. The results then were compared with the Naïve Bayes (NB)classifier using various n-gram range parameters and additional feature extraction, including Bag of Words (BoW). Results found that SVM, with a percentage accuracy of 95.742, outperformed NB in identifying cyberbullying content. Additionally, [18] and [19] conducted a study on detecting Arabic cyberbullying using Naïve Bayes. According to the authors, the majority of previous research suggested methods to identify English cyberbullying and a few other languages, with only a few publications addressing Arabic cyberbullying identification. In this research, cyberbullying in Arabic social media streams was automatically recognized via machine learning as real data gathered from YouTube and Twitter were utilized for investigation. Their methodology employed the Naïve Bayes (NB) algorithm for cyberbullying detection with an accuracy of 95.9 % was achieved.

An innovative approach, namely Optimized Twitter Cyberbullying Detection based on Deep Learning (OCDD) that addresses the challenges in feature extraction was presented in [20] where Tweets were represented as sequences of word vectors to employ DL for classification. However, additional discussion on its limitations and comparative performance is needed. A deep neural networks and word embeddings for cyberbullying detection were explored in [21]. The authors integrated both stacked Bert and Glove embeddings; thus, their proposed classifier outperformed the conventional ML techniques. However, a comprehensive performance analysis with traditional ML methods will enhance the credibility of the study. Researchers in [22] proposed a methodology adopting ensemble-based voting models for cyberbullying detection. While their research presented effective classifiers, a critical evaluation of their real-world applicability is essential. Researchers in [23] proposed a BiGRU-CNN sentiment classifier to capture essential features. Their classifier exhibited remarkable classification accuracy compared to conventional approaches. Adopting a pre-trained BERT model, as proposed by [19] and [24], provides a robust solution for cyberbullying detection. While their model achieved good results, alternative DL models could be explored for enhanced performance. Researchers in [25] utilized both ML and DL algorithms for detecting Bangla text cyberbullying. Their study demonstrated the effectiveness of different algorithms in different dataset contexts. The authors in [26] presented an approach for cyberbullying detection across multiple social media platforms, utilizing LSTM layers to outperform traditional methods.

Additional efforts, such as [12], [24], [27], [28], [28], [29], [30], [31], [32], [33], [34], and [35] have contributed to advancing cyberbullying detection through various ML and DL techniques. These studies highlighted the evolving landscape of cyberbullying detection research and the ongoing research for effective detection and prevention strategies.

III. CYBERBULLYING DETECTION: TASK DEFINITION

Identifying cyberbullying poses inherent challenges due to its subjective nature. Different individuals may interpret the same content differently; what some perceive as cyberbullying may be considered harmless banter by others. This variability among annotators leads to low agreement, complicating the automation of cyberbullying detection. To improve the accuracy of automated systems, establishing a precise definition of cyberbullying is crucial. However, consensus on its definition remains elusive and lacks a universally accepted standard. Major social media platforms such as Twitter, Facebook, and YouTube have each formulated their own definitions, as outlined in below.

Twitter defines cyberbullying under its policy on hateful conduct, stating that users may not promote violence against or directly attack or threaten other people based on race, ethnicity, national origin, sexual orientation, gender, gender identity, religious affiliation, age, disability, or disease. Twitter aims to prevent hostile behaviors that can lead to real-world harm and distress among its users.

Facebook takes a similar stance, prohibiting content that attacks people based on their actual or perceived race, ethnicity, national origin, religion, sex, gender or gender identity, sexual orientation, disability, or disease. However, Facebook does allow clear attempts at humor or satire that might otherwise be considered offensive, recognizing the fine line between hate speech and free expression. This includes content like jokes, stand-up comedy, and popular song lyrics, acknowledging that what might be in bad taste for some could be humorous for others.

YouTube addresses cyberbullying by banning content that promotes violence or hatred against individuals or groups based on attributes such as race, ethnic origin, religion, disability, gender, age, veteran status, and sexual orientation/gender identity. YouTube highlights the complexity of defining hate speech, noting that criticism of nation-states is generally acceptable, whereas malicious hateful comments targeting people based on their ethnicity are not tolerated.

For the purposes of this research, we define cyberbullying as the act of using electronic communication to bully, harass, or intimidate an individual, often repeatedly, by sending, posting, or sharing negative, harmful, false, or mean content about someone else. It can include sharing personal or private information about someone else causing embarrassment or humiliation. The complexity of cyberbullying often makes it challenging to distinguish it from related concepts like hate speech, abusive language, and offensive language. Establishing clear definitions and guidelines is essential for accurate identification and effective mitigation within various contexts.

IV. PROPOSED MODEL

This research aims at developing a hybrid cyberbullying detection model to classify Arabic text as cyberbullying or not-cyberbullying. To build the hybrid model, multiple classification ML and DL algorithms were explored in order to compare their accuracy and effectiveness. The focus was on identifying the most suitable classifiers that improves the accuracy of cyberbullying classification to be adopted in building the hybrid model. Twelve ML and DL algorithms were investigated: 1) SVM, 2) KNN, 3) NB, 4) DT, 5) RF, 6) RBFNN, 7) MLPNN, 8) CNN, 9) LSTM, 10) RNNs), 11) Bi-LSTM and 12) GRU.

Building the proposed model has four main parts: dataset settings, the applied pre-process for cleaning the text, the models' settings and architectures and the performance evaluation method.

A. DATASETS

Six collected benchmark datasets from Facebook, Twitter and Instagram in addition to a developed Arabic cyberbullying lexicon were utilized to evaluate the efficiency of the proposed hybrid model. To determine whether the models consistently produce accurate results regardless of the social network dataset's type and size, multiple and diverse datasets are utilized. However, several criteria were applied to select these datasets, such as size, dialects that cover, accessibility and availability. The details of dataset setting are listed below in the following subsections.

1) PHASE 1: BUILDING AN ARABIC CYBERBULLYING LEXICON: DATA COLLECTION AND ANNOTATION PROCESS

This phase is dedicated to constructing a lexicon of Arabic cyberbullying terms. To compile this lexicon, we conducted an online survey involving 100 Arabic-speaking participants, primarily university students aged between 17 and 30. Each student was requested to list the ten Arabic cyberbullying words they encounter most frequently. Subsequently, we selected the top twenty words for further analysis. The survey identified variations and spelling differences of the same terms, which were then merged to reflect real-world usage accurately.

a: DATA ACQUISITION

Cyberbullying terms that were identified served as keywords in a data acquisition program which in turn employed these keywords as seeds to retrieve data through the Twitter Application Programming Interface (API). Specific criteria were set for the collected tweets, specifically gathering tweets containing more than three Arabic words using the Twitter API to construct a manually annotated dataset. This data collection process extended over three months, starting from March to June 2022, yielding an initial dataset of 14,596 tweets

b: ENRICHING THE DATASET

To build a more comprehensive dataset, the data collection efforts were extended to include tweets from popular Twitter hashtags related to trending topics and well-known public figures such as movie actors, sports personalities, and politicians. These additional supplementary datasets were then merged with the initial dataset, forming the core corpus for this research with 22,898.

c: DATA PREPROCESSING AND ANNOTATION

The compiled corpus subjected to a preprocessing phase involving several steps to convert tweets into plaintext.

Subsequently, manual annotation of the dataset was conducted, assigning each line to one of three categories based on guidelines established in previous research:

- Normal tweets (A1): These tweets do not contain any offensive, aggressive, derogatory, or violent language.
- Abusive tweets (A2): This category encompasses tweets that include abusive, aggressive, disrespectful, or profane language.
- Hate tweets: These tweets meet specific criteria, including (A3) targeting of individuals or groups with such language, (A4) and humiliation or dehumanization based on attributes such as race, gender, religion, disability, skin color, or belief.

d: ANNOTATION CONSENSUS AND EXPERT GUIDANCE

In cases where annotators disagreed on categorizing tweets, we sought the expertise of NLP specialist to give a definitive resolution. Ultimately, 183 tweets were annotated in consultation with the NLP expert, while consensus among annotators was used to label 2,089 tweets. The inter-annotator agreement, as measured by Cohen's Kappa, yielded an average value of 0.86, indicating a substantial level of consensus. The internal agreement among our four annotators is displayed in Fig.1.

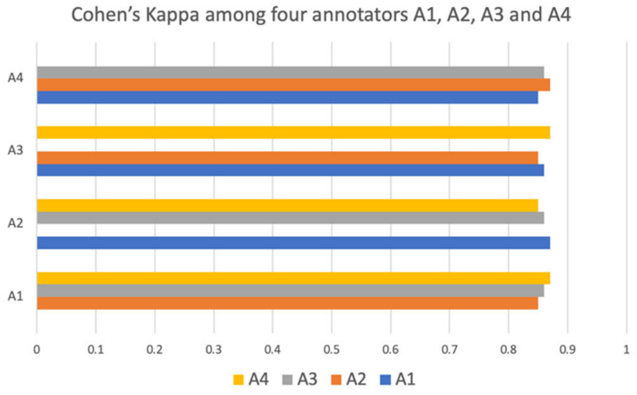


FIGURE 1. Cohen's Kappa among four annotators.

2) PHASE 2: BENCHMARK DATASETS COLLECTION

Six open-source benchmark Arabic datasets, consisting of 243,849 records, were utilized in this research.

The first dataset, is a GitHub dataset [36], entails 72,730 records with two columns: the 1st column entails text messages, and the 2nd column entails the classifications of the message denoted as 0 or 1 (0 indicates not cyberbullying, while 1 indicates a cyberbullying text). Data1 is a combination of several datasets of Arabic comments and hate speech. It contains 37,093 records with a label of 0, and 35,637 records are labeled as 1.

The second dataset is a Kaggle dataset [37] referred to as "Arabic Sentiment Twitter Corpus" which was collected in April 2019. The dataset encompasses 58K Arabic tweets containing positive and negative classifications, consisting of 47K for training and 11K for testing, which includes both negative and positive classes. The dataset was compiled to ensure an even distribution of positive and negative texts.

The third dataset is also sourced from Kaggle entitled "Arabic Levantine Hate Speech Detection" [38]. This dataset was introduced during the 3rd Workshop ALW-2019, held in conjunction with ACL-2019 in Florence, Italy. It marked the inception of the first Arabic Levantine Hate Speech and Abusive Language Dataset. It comprises 1171 records and is formatted as a Tab-delimited values (TSV) file. The dataset encompasses two columns: the tweet content and the class label. The classes are categorized as hateful, abusive, and normal. Within this dataset, the primary classification objectives include Binary Classification (Normal, Abusive) and multi-class classification (Normal, Abusive, Hate). The dataset underwent preprocessing, with the hate and abusive classifications being converted to 1, indicating cyberbullying, while the normal status was converted to 0, representing the absence of cyberbullying. The dataset underwent further division into training and testing subsets. The features within the dataset are the tweet content and the corresponding annotations (Normal, abusive, and hateful).

The forth dataset is a GitHub dataset [36] referred to as "Arabic-Abusive-Datasets" which was collected in 2021. The dataset encompasses 50K Arabic Instagram comments containing positive and negative classifications, consisting of 35K for training and 15K for testing, which includes both negative and positive classes. The dataset was compiled to ensure an even distribution of positive and negative texts.

The fifth dataset is the one that was developed in 2022 by Albayari and Abdallah [39]; it consists of 46,898 comments collected from Instagram. According to [39], it is the largest publicly available dataset for detecting cyberbullying covering various dialects.

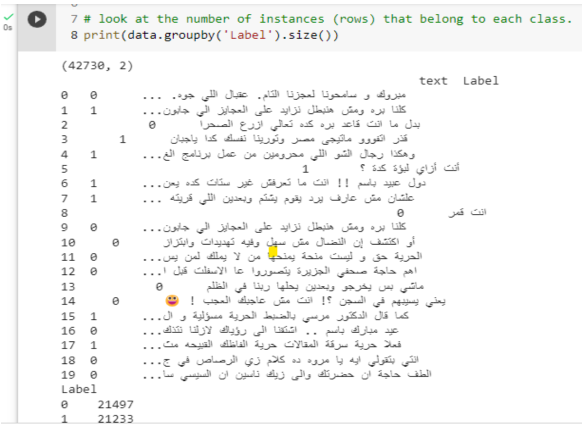
The sixth dataset is presented by Alakrot et al. in 2028 [40] consisting of comments that are offensive: 5,813 and Not offensive: 9,237. According to [39], it is the second largest dataset regarding the number of offensive comments with manual labelling. It also includes a variety of dialects. We name it as authors' names-Binary Class-Balanced (AMN-BCU). Fig. 2 shows a dataset sample.

3) DATASET INTEGRATION

Subsequently, six datasets consisting of 243,849 records in addition to the built cyberbullying lexicon with 22,898 records were merged to form a comprehensive "cyberbullying dataset" consisting of 266,747 record (see Fig. 3), where 49% of the dataset is labeled as not cyberbullying and 51% of the dataset is labeled as cyberbullying. All datasets undergo the preprocessing phase that is depicted in the following section.

B. DATA PREPROCESSING PHASE

To apply the ML and DL algorithms to the text dataset, several preprocessing steps were considered involving:



```
# look at the number of instances (rows) that belong to each class.
print(data.groupby('Label').size())
```

```
(42730, 2)
                                                    text  Label
0    0     مبروك و سامحونا لعجزنا التام. عقبال اللي جوه. ...
1    1     كلنا بره ومش هنبطل نزايد على العجايز الي جابون...
2          0     بدل ما انت قاعد بره كده تعالي ازرع الصحرا
3        1     قذر اتفووو ماتيجي مصر وتورينا نفسك كدا ياجبان
4    1     وهكذا رجال الشو اللي محرومين من عمل برنامج الغ...
5                        1     أنت أزاي لبؤة كدة ؟
6    1     دول عبيد باسم !! انت ما تعرفش غير ستات كده يعن...
7    1     علشان مش عارف يرد يقوم يشتم وبعدين اللي قريته ...
8                              0     انت قمر
9    0     كلنا بره ومش هنبطل نزايد على العجايز الي جابون...
10       0     أو اكتشف إن النضال مش سهل وفيه تهديدات وابتزاز
11   0     الحرية حق و ليست منحة يمنحها من لا يملك لمن يس...
12   0     اهم حاجة صحفي الجزيرة يتصوروا عا الاسفلت قبل ا...
13         0     ماشي بس يخرجو وبعدين يحلها ربنا في الظلم
14       0     يعني يسيبهم في السجن ؟! انت مش عاجبك العجب ! 😛
15   1     كما قال الدكتور مرسي بالضبط الحرية مسؤلية و ال...
16   0     عيد مبارك باسم .. اشتقنا الي رؤيإك لازلنا نتذك...
17   1     فعلا حرية سرقة المقالات حرية الفاظك القبيحه مث...
18   0     انتي بتقولي ايه يا مروه ده كلام زي الرصاص في ج...
19   0     الطف حاجة ان حضرتك والي زيك ناسين ان السيسي سا...
Label
0    21497
1    21233
```

FIGURE 2. Sample of the dataset.

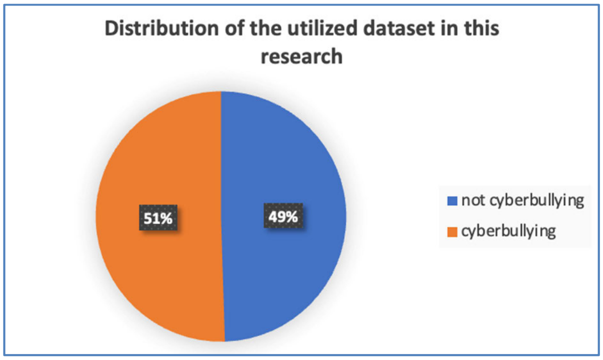


FIGURE 3. Distribution of the utilized dataset in this research.

- Text Cleaning: This step involves the removal of NaN values, empty cells, and irrelevant attributes. Additionally, the data is formatted to eliminate unnecessary characters like special symbols, punctuation marks, or HTML tags, which might not add to the textual meaning. As a result, uniformity in the data type is achieved throughout the dataset.
- Data Transformation: Given that the three datasets originate from distinct sources, directly amalgamating them in their original formats would be incompatible due to divergent classification labels. To address this, a unifying approach is adopted, wherein different label sets are consolidated into a singular classification technique, the 0-1 classifier. This approach simplifies the task by categorizing whether the text contains cyberbullying content or not, creating a clear distinction for training the model and removing any ambiguous cases.
- Data Integration: The datasets are consolidated into a single CSV file, serving as a comprehensive source for subsequent stages of preprocessing.
- Tokenization (Data Discretization): At this phase, the data is tokenized, breaking down sentences into individual words for efficient analysis. This process aids in segmenting the text into meaningful units, facilitating further processing.
- Data Reduction: This text preprocessing stage involves the removal of elements such as URLs, special characters, '@' symbols, and stopwords from the text. This pruning operation reduces noise within the data.
- Text Normalization: The text is standardized by applying actions like converting all text to lowercase, eliminating extraneous whitespaces, and handling contractions. This process ensures uniformity in the text data.
- Stemming or Lemmatization: This phase is employed to streamline words to their root forms or base, promoting normalization by addressing variations. While stemming removes prefixes and suffixes to obtain root forms, lemmatization maps words to dictionary forms. This step is crucial for simplifying the data to its core components.
- Vectorization: To process the dataset, we converted the text into numerical representations that ML and DL algorithms can process. Word Embeddings techniques including both FastText and GloVe modeling were utilized. Word Embeddings encode words as dense vectors within a high-dimensional space, capturing semantic relationships among words. FastText is a lightweight, open-source library that enables users to acquire text representations and build text classifiers. It takes a unique approach by treating words as compositions of character n-grams, and a word's vector representation is derived from the summation of its character n-grams. FastText provides a wide array of input parameters for classification training, and to optimize accuracy and precision, these parameters were fine-tuned using a grid-search method. On the other hand, GloVe is employed to learn word embeddings by analyzing extensive text corpora, thereby capturing the semantic relationships among words. In contrast to some other word embedding techniques like Word2Vec, which focus on predicting a word from its context or predicting context from a word, GloVe is specifically designed to directly model the co-occurrence statistics of words within a corpus.
- Sequence Padding: This step ensures that all input sequences have the same length. For DL models, it is necessary to pad or truncate the sequences to a fixed length.

C. BUILDING THE HYBRID MODEL PHASE

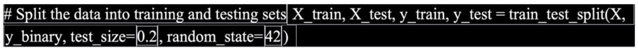
1) TRAINING THE ALGORITHMS ON THE DATASET

In this phase, seven ML algorithms in addition to five DL algorithms including, 1) SVM, 2) KNN, 3) NB, 4) DT, 5) RF, 6) RBFNN, 7) MLPNN, 8) CNN, 9) LSTM, 10) RNNs), 11) Bi-LSTM and 12) GRU were used for training. A target training error is set to 1e-5. The training process will be repeated until the training error is less than target training error. Subsequently, the model underwent rigorous testing and evaluation using the remaining 20% of the local dataset. Based on the results, the best performing algorithm(s) will be employed to build the hybrid model.

In our experiment, the train_test_split function from the scikit-learn library was employed to partition the data into training and testing sets. We designated 20% of the data for testing using the test_size parameter. Furthermore, we set the random_state parameter to guarantee the reproducibility of the splitting process, as depicted in the Fig. 4. The results of the testing, encompassing accuracy, sensitivity, precision, specificity, and F-Score, are listed in Table 1 and Fig. 5.

TABLE 1. Performance metrices for cyberbullying detection.

Algorithm	Accuracy	Recall	Precision	Specificity	F1-Score
SVM	91.17 %	90.29%	92.06%	92.90%	91.17%
NB	88.24%	87.76%	88.72%	89.70%	88.24%
DT	90.58%	89.88%	91.29%	92.22%	90.58%
RF	92.34%	91.74%	93.00%	93.93%	92.34%
KNN	88.42%	87.92%	88.92%	90.00%	88.42%
MLPNN	91.29%	90.67%	91.91%	92.59%	91.29%
RBFNN	89.54%	89.03%	90.06%	90.98%	89.54%
CNN	92.83%	91.51%	92.76%	93.47%	92.13%
GRU	92.79%	92.06%	93.12%	93.84%	92.59%
Bi-LSTM	93.10%	92.47%	93.53%	94.29%	93.00%
RNN	90.00%	89.47%	90.53%	91.43%	90.00%
LSTM	91.74%	91.20%	92.28%	93.00%	91.74%



```
# Split the data into training and testing sets X_train, X_test, y_train, y_test = train_test_split(X,
y_binary, test_size=0.2, random_state=42)
```

FIGURE 4. Splitting the dataset.

Accuracy assesses the correctness of the classification model by representing the ratio of correctly predicted instances to the total instances in the dataset. A higher accuracy indicates better overall model performance. Recall, also known as True Positive Rate or Sensitivity, measures the model's ability to correctly recognize positive instances by calculating the ratio of correctly predicted positive instances to the actual positive instances. Higher recall signifies the model's proficiency in identifying positive cases. Precision gauges the accuracy of positive predictions made by the model, determining the ratio of correctly predicted positive instances to the total predicted positive instances. A higher precision implies that when the model predicts positive, it is usually correct. Specificity evaluates the model's capability to correctly identify negative instances through the ratio of correctly predicted negative instances to the actual negative instances. Greater specificity indicates the model's effectiveness in identifying negative cases. The F1-Score, which is the harmonic mean of precision and recall, considers both false positives and false negatives, providing a balanced measure of the model's performance. This is particularly valuable when dealing with imbalanced classes.

As shown in the results, the algorithms appear to have achieved a good balance between correctly identifying positive and negative instances, leading to reliable predictions for the given classification task.

As shown in the table, the algorithms generally exhibit good performance, with accuracy ranging from around 88% to 93%. SVM, Random Forest, MLPNN, CNN, GRU, and Bi-LSTM show higher accuracy, indicating their ability to make accurate predictions overall. NB, KNN, and RBFNN have slightly lower accuracy, but they still perform well. Recall values are generally close to accuracy, suggesting that the models are good at identifying positive instances. Precision values are also high, indicating that when the models predict positive, they are usually correct. Specificity values are generally high, showing that the models are also effective at identifying negative instances. F1-Score values are consistent with the high precision and recall, highlighting a balanced performance.

As shown in the table 1 and figure 5, the best performing algorithm for cyberbullying classification training is Bi-LSTM across multiple metrics, including accuracy, recall, precision, specificity, and F1-Score.

BiLSTM has the highest accuracy (93.10%) and a strong F1-Score (93.00%). Additionally, it offers high precision (93.53%) and recall (92.47%). This makes it a promising choice for your cyberbullying detection model. The second-best performing algorithm is CNN, with high accuracy (92.83%) and a competitive F1-Score (92.13%). It maintains good recall (91.51%) and precision (92.76%). Considering CNN's capability in capturing local patterns, it can be a valuable addition to the cyberbullying detection model. The third best performing algorithm is GRU, with with excellent accuracy (92.79%) and F1-Score (92.59%). It offers high precision (93.12%) and recall (92.06%). GRU is known for its efficiency and ability to capture dependencies in sequences, which aligns with text data characteristics.These three algorithms, Bi-LSTM, CNN, and

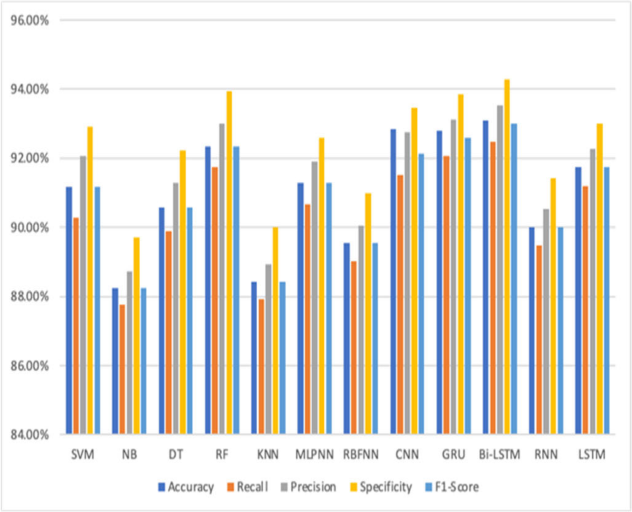


FIGURE 5. Evaluation results for cyberbullying classification.

GRU, show consistent high performance across various evaluation metrics. Therefore, we consider experimenting with them further, in a hybrid approach, to attain the best results for Arabic text cyberbullying detection task.

2) DEVELOPING A HYBRID MODEL FOR ARABIC CYBERBULLYING DETECTION

When comparing CNN, Bi-LSTM, and GRU for text cyberbullying classification, each of these models has its unique characteristics and performance attributes. Following paragraphs provides ablation study on each module.

a: Bi-LSTM

Bi-LSTM is a Recurrent Neural Network (RNN) that handles sequences bidirectionally in order to capture both past and future information. The two-directional processing enables Bi-LSTM effectively capturing contextual nuances, which is crucial for understanding the context of cyberbullying instances. Bi-LSTM handles dependencies across long distances in text, making them well-suited for tasks where understanding the sequential order of words is essential. The architecture of Bi-LSTM is shown in Fig. 6. The embedding layer in Bi-LSTM maps input tokens, such as words or characters, to dense vectors of fixed size, called embeddings to capture the semantic meaning of the tokens during the training process. Note that, as shown in Figures 6, 7, 8, 10, the embedding layer serves as the first layer in all Bi-LSTM, CNN, and GRU models, performing the same function across all architectures.

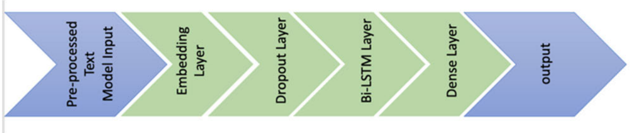


FIGURE 6. Bi-LSTM model architecture.

Additionally, the dropout layer prevents overfitting by randomly setting some neurons to zero during each forward pass, thereby enhancing the robustness of the model. The output from the dropout layer serves as the input to the Bi-LSTM layer, which consists of both forward and backward LSTM layers. These layers handles the sequence of numerical representations and output hidden states at each time step to efficiently handling long-term dependencies used for language processing tasks.

Finally, the output from the Bi-LSTM layer is passed to a dense layer to produce prediction scores for each class, indicating the probability of belonging to each class.

b: GRU

This section provides an overview of the Gated Recurrent Unit (GRU), given the similarity of its architecture with the Bi-LSTM. The functionalities of the embedding, dropout and dense layers have been illustrated upon in the previous section. Fig. 7 presents an overview of the GRU model architecture. GRUs are commonly employed in multi-class classification tasks, where the model is trained on a labeled dataset with samples associated with various categorical outcomes, such as Positive, Neutral, Bullying, or Toxic. GRU units handle input data at each time step in order to generate hidden states that encode information about the input sequence. These hidden states are then input to a fully connected layer, which produces final predictions based on learned weights. Subsequently, these predictions are compared to the actual target labels, and any errors are backpropagated through the network to update the weights, thereby enhancing accuracy over time.

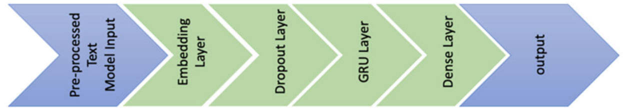


FIGURE 7. GRU model architecture.

c: CNN

CNNs, originally designed for image recognition, have demonstrated their effectiveness in text classification tasks such as cyberbullying detection. They capture local patterns and features within sequences. In the context of text, 1D CNNs use convolutional filters to scan through word sequences, identifying significant word combinations and patterns. This ability enable CNNs recognizing relevant textual features for cyberbullying identification.

The CNN architecture consists of convolutional layers followed by pooling layers. These layers are responsible for extracting spatial patterns and features from the input data. The convolutional layers perform feature extraction by applying convolutional filters over the input tokens or characters. These filters capture local patterns and detect relevant features. Pooling layers then reduce the dimensionality of the extracted features while retaining the most important information. This helps in reducing overfitting and computational complexity. In the context of cyberbullying detection, CNNs capture local patterns and identifying key features indicative of cyberbullying content, such as aggressive language or offensive phrases.

d: CNN-BiLSTM-GRU

In the context of text cyberbullying classification, a hybrid model (Fig. 8) that integrates the strengths of CNN, Bi-LSTM, and GRU could handle their limitations and enhance the performance. This hybrid approach could emphasize the pattern recognition capabilities of CNN, the long-range dependency understanding of Bi-LSTM, and the memory efficiency of GRU to create a complete solution for accurately identifying instances of cyberbullying in text. Consequently, the GRU units process the input data for each time step to produce hidden states, which capture information about the input sequence. These hidden states are then passed to a fully connected layer that outputs the final predictions based on the learned weights. The predictions are compared to the actual target labels, and the error is backpropagated to update the weights, leading to improved accuracy over time. The Bi-LSTM layer, which encompasses both forward and backward LSTM layers, processes the sequences of feature vectors generated by the CNN, capturing the long-term dependencies between the input data. Unlike a regular LSTM, a BLSTM processes the input sequence in forward and backward directions, allowing it to capture information from past and future time steps. The hidden states produced by the Bi-LSTM are then passed to a fully connected layer that outputs the final predictions based on the learned weights. The predictions are then compared to the actual target labels, and the error is backpropagated to update the weights, leading to improved accuracy over time.

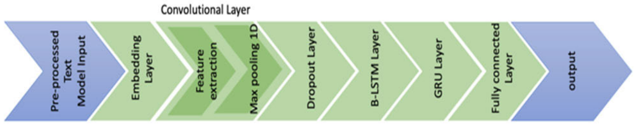


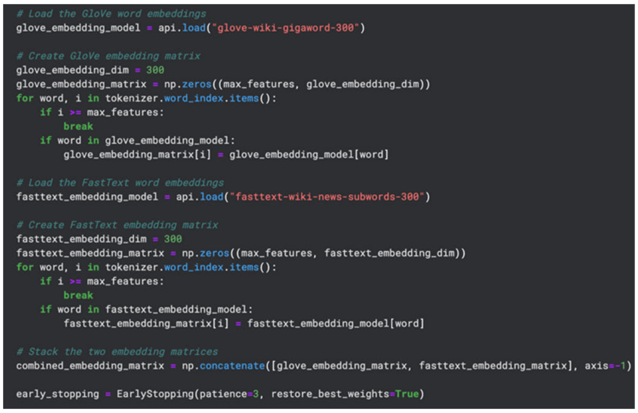
FIGURE 8. Overview of proposed model architecture.

3) ENHANCING THE MODEL WITH STACKED WORD EMBEDDING

To construct an optimized hybrid model, we recognized the critical role that text representation plays in the model performance [20]. Traditional one-hot vector representations have limitations, especially in capturing semantic relationships among words. To overcome these challenges, we adopted the Word Embedding technique, a fundamental step towards creating a more effective cyberbullying detection model.

Word embeddings provide a solution by encoding words as dense vectors in a high-dimensional space, thereby preserving both contextual and semantic information. However, rather than relying on a single word embedding model, we took a strategic approach by stacking two distinct word embeddings: GloVe and FastText. This decision was made with the goal of harnessing the unique strengths of each embedding method to achieve optimal outcomes.

By combining the strengths of GloVe and FastText (as shown in Fig. 9) through stacked word embedding, we aim to create a rich and nuanced text representation that reflects the complex nature of cyberbullying communication. This enhanced text representation will serve as the foundation upon which our optimized hybrid model for cyberbullying detection will be built.



```
# Load the GloVe word embeddings
glove_embedding_model = api.load("glove-wiki-gigaword-300")

# Create GloVe embedding matrix
glove_embedding_dim = 300
glove_embedding_matrix = np.zeros((max_features, glove_embedding_dim))
for word, i in tokenizer.word_index.items():
    if i >= max_features:
        break
    if word in glove_embedding_model:
        glove_embedding_matrix[i] = glove_embedding_model[word]

# Load the FastText word embeddings
fasttext_embedding_model = api.load("fasttext-wiki-news-subwords-300")

# Create FastText embedding matrix
fasttext_embedding_dim = 300
fasttext_embedding_matrix = np.zeros((max_features, fasttext_embedding_dim))
for word, i in tokenizer.word_index.items():
    if i >= max_features:
        break
    if word in fasttext_embedding_model:
        fasttext_embedding_matrix[i] = fasttext_embedding_model[word]

# Stack the two embedding matrices
combined_embedding_matrix = np.concatenate([glove_embedding_matrix, fasttext_embedding_matrix], axis=-1)

early_stopping = EarlyStopping(patience=3, restore_best_weights=True)
```

FIGURE 9. Employing stacked word embedding.

4) SELECTING THE BEST OPTIMIZER AND THE ACTIVATION FUNCTIONS

This phase focuses on optimizing the model by selecting the most suitable combination of optimizers and activation functions as they play a pivotal role in shaping the model's performance.

Optimizers are essential for updating model weights during training. As optimizers for CNN, BiLSTM, and GRU, 1) Adam, 2) Stochastic Gradient Descent (SGD), 3) and RMSProp are employed. Adam is known for its computational efficiency, minimal memory requirements, and adaptability to varying gradient conditions. It's suitable for scenarios with extensive data and parameters. RMSProp controls gradient magnitudes via a decay rate, offering efficient performance. It focuses on the second moment and is faster than Adam. Stochastic Gradient Descent (SGD) is a fundamental optimizer that updates weights based on gradient information. Its performance varies based on learning rate and momentum settings.

Moreover, activation functions introduce non-linearity into neural networks, allowing them to capture complex patterns. For CNN, BiLSTM, and GRU, 1) Tanh (Hyperbolic Tangent), 2) Leaky ReLU, 3) ReLU and 4) Sigmoid are utilized as activation layers. Sigmoid function is ideal for binary classification tasks. It compresses values to the range (0, 1), making it a popular choice for such tasks. Similar to Sigmoid, Hyperbolic Tangent (Tanh) outputs values in the range (-1, 1), addressing the vanishing gradient problem. Rectified Linear Unit (ReLU) replaces negative inputs with zero, mitigating vanishing gradient issues and accelerating convergence. Leaky ReLU is a modified version of ReLU that prevents ''dead neurons'' by allowing a slight gradient for inactive units.

A total of 12 combinations were created by pairing the activation functions with the optimizers mentioned above. To determine the best combination for the proposed hybrid model, we evaluate these configurations based on accuracy. Table 2 below shows a comparison of optimizer-activation function combination on LSTM as baseline model.

TABLE 2. Effect of optimizers and activation functions in cyberbullying detection.

Optimizer	Activation Function	Accuracy
Adam	Tanh	91.01%
Adam	Leaky ReLU	91.19%
Adam	ReLU	91.32%
Adam	Sigmoid	92.83%
SGD	Tanh	89.61%
SGD	Leaky ReLU	89.79%
SGD	ReLU	90.06%
SGD	Sigmoid	89.36%
RMSProp	Tanh	90.72%
RMSProp	Leaky ReLU	90.98%
RMSProp	ReLU	91.26%
RMSProp	Sigmoid	90.53%

The findings suggest that the best results among the 12 configurations were achieved when using the combination of Sigmoid activation with the Adam optimizer.

Apart from employing the Sigmoid activation function, the proposed hybrid model also integrates ReLU as an activation layer within its hidden layers as it helps mitigate the vanishing gradient problem and speeds up training. This dual usage is driven by the fact that the simplicity of the Sigmoid function and its derivative expedites model creation, yet it carries a notable drawback: information loss due to its limited derivative range. Consequently, as the network's depth increases, information becomes progressively condensed and forfeited with each layer, resulting in significant data depletion throughout the architecture. ReLU is a nonlinear activation function that avoids the back-propagation challenges encountered with the sigmoid function. Furthermore, when dealing with larger artificial neural networks, the process of model creation using ReLU is considerably swifter compared to employing sigmoid functions. This rationale underpins the preference for ReLU over sigmoid as the activation function for the hidden layers.

Finally, the different combinations of the Bi-LSTM, CNN and GRU are investigated as shown in Table 3. Hyperparameter adjustments were carried out alongside a comparative evaluation to identify the optimal settings. The comparative analysis unmistakably indicates that the CNN-Bi-LSTM-GRU model with Adam optimizer and Sigmoid activation function stands out with superior performance.

TABLE 3. Effect of activation functions and optimizers on cyberbullying detection models.

Model Combination	Activation Function	Optimizer	Accuracy (Before Hypertuning)	Accuracy (After Hypertuning)
CNN-GRU	Sigmoid	Adam	90.23%	90.83%
CNN-BiLSTM	Sigmoid	Adam	90.06%	90.83%
GRU-BiLSTM	Sigmoid	Adam	89.99%	90.83%
CNN-Bi-LSTM-GRU	Sigmoid	Adam	92.79%	93.96%

Thus, this combination of CNN-Bi-LSTM-GRU model with Adam optimizer and Sigmoid activation function is adopted to formulate the hybrid model.

5) MODEL ARCHITECTURE: CNN-BI-LSTM-GRU HYBRID

This section explores the architecture of the proposed hybrid model that integrates elements from CNN, Bi-LSTM and GRU deep learning models to build an enhanced cyberbullying classification. The proposed model was implemented and trained using Keras, an API that is built on top of the TensorFlow framework to provide a high-level interface for neural network development [39]. The implementation process was conducted in PyCharm, with optimal parameter settings.

In the training process, key parameters were configured to enhance the model performance and grid search was employed to fine-tune the hyperparameters. A grid of hyperparameter values, such as dropout rate, number of filters, kernel size, and learning rate was defined. Dropout rates ranging from 0.1 to 0.5, filter sizes ranging from 16 to 64, kernel sizes from 3 to 7, and learning rates from 0.001 to 0.01 were explored. Each combination of hyperparameters was evaluated using 10-fold cross-validation on the training dataset. During each fold, the model was trained on 90% of the training data and validated on the remaining 10%. The performance metric, including accuracy or F1-score, was computed for each fold, and the average performance across all folds was calculated for each hyperparameter combination. The hyperparameter combination yielding the highest average performance metric was selected as the optimal set of hyperparameters for the final model (see Table 4). This approach improves the performance on unseen test data and ensures that the model will be robust and generalizable across different subsets of the training data.

Table 5 summarizes the key parameters used in training the hybrid CNN-Bi-LSTM-GRU model. A dropout rate of 0.3 was applied to prevent overfitting, while the dense layer, set to 1, utilized the softmax activation function to generate class probabilities. For the CNN component, 32 filters with a kernel size of 5 were specified to extract meaningful features indicative of cyberbullying instances. In the Bi-LSTM layer, 128 hidden nodes captured long-term dependencies bidirectionally, and the GRU layer with 64 hidden nodes efficiently processed input sequences. Utilizing the Adam optimizer with a learning rate of 0.005, the model dynamically adapted

TABLE 4. The hyperparameters tested during the grid search process.

Hyperparameter	Value(s) Tested	Accuracy, F-Score	Best Value Selected
Dropout Rate	0.1	0.84, 0.89	0.3
	0.2	0.83, 0.91	
	0.3	0.94, 0.92	
	0.4	0.80, 0.89	
	0.5	0.89, 0.91	
Number of Filters	16	0.84, 0.89	32
	32	0.93, 0.91	
	48	0.84, 0.89	
	64	0.80, 0.88	
Kernel size	3	0.84, 0.89	5
	5	0.92, 0.90	
	7	0.84, 0.89	
Learning rate	0.001	0.83, 0.89	0.005
	0.005	0.92, 0.91	
	0.01	0.86, 0.89	

learning rates for individual parameters during training. Trained for 15 epochs with a batch size of 32, the model leveraged the Relu activation function in hidden layers to introduce non-linearity and learn complex patterns from input data.

TABLE 5. Key parameters used in training the hybrid CNN-Bi-LSTM-GRU model.

Parameter	Value
Dropout rate	0.3
Dense layer	1
Number of filters	32
Kernel size	5
Number of hidden nodes (Bi-LSTM)	128
Number of hidden nodes (GRU)	64
Optimizer	Adam
Activation function (hidden layer)	ReLU
Activation function (final output)	Sigmoid
Number of epochs	15
Batch size	32
Learning rate	0.005

Fig. 10 illustrates the integrated components of our hybrid model:

- Stacked Word Embedding: Leveraging GloVe and FastText embeddings, this component captures semantic relationships among words, enriching the model's understanding of text.
- Convolutional Model: Incorporating a CNN, our model adeptly extracts essential features from text, aiding in pattern recognition.
- Hybrid Architecture: Combining Bi-LSTM and GRU layers, equipped with dropout and batch normalization, comprehends contextual dependencies bidirectionally. A 1D convolutional layer with max-pooling enriches feature extraction across multiple scales, culminating in fully connected dense layers with ReLU activation for high-level feature extraction and a final dense layer with sigmoid activation for binary classification.
- Fully Connected Model: Interprets extracted features, establishing a connection between learned features and the final output.

The input layer defines sequence length, while the embedding layer generates 100-dimensional representations aligned with vocabulary size, enhancing semantic understanding. Further architecture includes a Conv1D layer with 32 filters and a kernel size determined by simultaneous word processing, followed by a MaxPooling1D layer to distill essential features. The integrated Flatten layer facilitates concatenation and output transformation, while dropout and weight regularization enhance model robustness.

With a batch size of 128 and Adam optimizer, hyperparameter optimization through grid search and 10-fold cross-validation ensures effective training. Dropout layers play a key role in addressing overfitting and promoting model generalization. The proposed architecture emphasizes the model's capacity to effectively classify input documents as cyberbullying or non-cyberbullying instances, employing robust feature extraction and regularization mechanisms.

V. RESULTS AND DISCUSSION

A. EVALUATING THE PROPOSED HYBRID MODEL

The results of evaluating the proposed hybrid CNN-Bi-LSTM-GRU model on the dataset previously illustrated in section III is presented in this section. The strengths of the three models: GRU, CNN, and BiLSTM are utilized. GRU enables learning long-term dependencies, CNN enables learning local patterns in and Bi-LSTM supports a bidirectional long short-term memory architecture enabling capturing both forward and backward relationships within textual data. Table 6 summarizes the performance of the proposed model, including: Precision, F1-Score, Recall, Accuracy, and Specificity.

TABLE 6. Results of evaluating the proposed hybrid model.

	hybrid CNN-Bi-LSTM-GRU
Accuracy	98.83%
Recall	99.55%
Precision	98.15%
Specificity	98.09%
F1-Score	98.13%

Results demonstrate the efficiency of the proposed hybrid CNN-BiLSTM-GRU model in cyberbullying detection with an impressive accuracy of 98.83%. The model is capable of distinguishing cyberbullying instances and non-cyberbullying ones. Results show a high recall of 99.55% which indicates the capability of the model to correctly identify actual cyberbullying cases. Additionally, the precision of 98.15% indicates the model's ability in correctly labeling instances as cyberbullying, minimizing

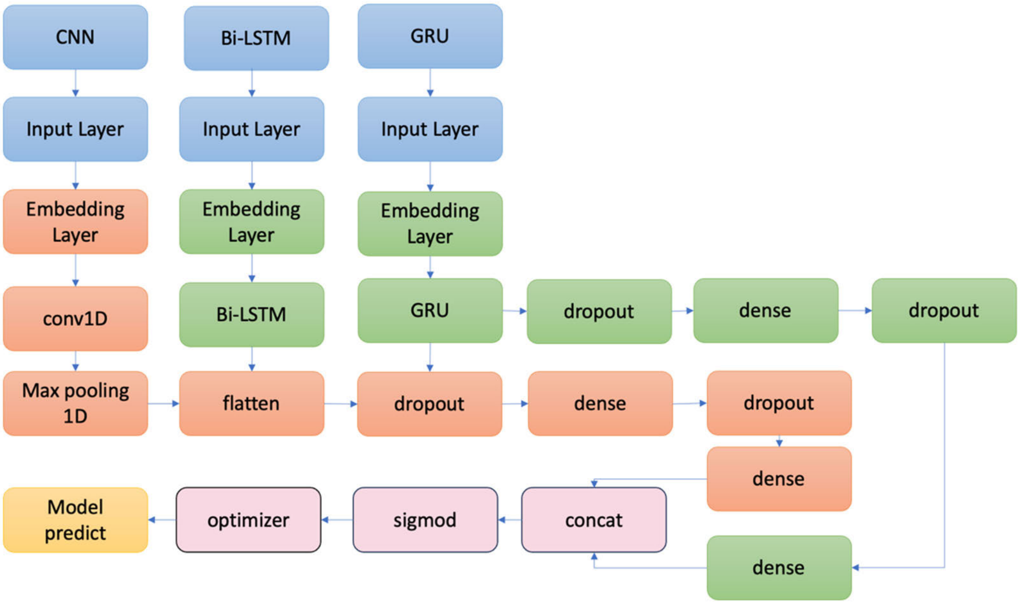


FIGURE 10. The architecture of the proposed hybrid model.

false positives. The ability of the proposed model to maintain a specificity of 98.09% demonstrates its effectiveness in accurately identifying non-cyberbullying instances. The F1-Score of 98.13% provides a balanced evaluation of the model's precision and recall, proving its overall effectiveness in handling both classes. In general, results highlight the potential of the hybrid proposed model as a valuable tool for cyberbullying detection.

A comprehensive breakdown of the results is presented in table 7. As shown in the table, 130,970 instances were correctly identified as positive cyberbullying, 127,681instances were recognized as not-cyberbullying, 589 instances incorrectly recognized as cyberbullying and 2473 instances were wrongly not detected by the algorithm as cyberbullying.

TABLE 7. Confusion matrix of the proposed model.

			Actual Class	
			True	Negative
Proposed hybrid CNN-Bi-LSTM-GRU	Predicated Class	True	TP 130,970	FN 589
		Negative	FP 2473	TN 127,681

According to the results, it appears that the proposed model shows promise in its capability of identifying instances related to cyberbullying. The high number of True Positives (TP) and True Negatives (TN) suggests that the proposed model is capable of correctly classifying both positive (cyberbullying) and negative (non-cyberbullying) instances with a considerable degree of accuracy.

B. COMPARISON OF EXISTING ARABIC CYBERBULLYING DETECTION APPROACHES AND PERFORMANCE METRICS

Table 8 below shows a summary of experimental results on cyberbullying detection in Arabic contents.

In [15], CNN as a DL approach is applied to detect cyberbullying in Arabic tweets. The dataset employed in this paper comprises Arabic cyberbullying tweets collected from Twitter. The authors utilized grid search to fine-tune the hyperparameters of the CNN model including the number of layers, the number of filters per layer, and the learning rate. The tweets in the dataset were represented using word embeddings generated from a pre-trained word embedding model. Results showed that the accuracy of the CNN model is 90.2% and its F-score is 0.91.

In [18], the authors employed SVM as a ML approach to recognize cyberbullying in Arabic tweets. The dataset utilized in their research consists of Arabic cyberbullying tweets collected from Twitter. To tune the hyperparameters of the SVM model including the kernel type, the penalty parameter, and the C parameter, grid search was employed. The tweets in the dataset were represented as Bag of Words (BoW) vectors by counting the appearance of words in each tweet. Accuracy of 88.1% and an F-score of 0.89 were achieved on testing set of unseen tweets.

Similarly, the authors in [33] utilized the SVM as a ML approach to identify Arabic cyberbullying in Twitter streams. The dataset employed is a collection of Arabic Twitter streams containing 100,000 tweets, where 50,000 are labeled as cyberbullying and 50,000 are labeled as non-cyberbullying. A grid search to tune the hyperparameters, including kernel type, the penalty parameter, and the C parameter of the SVM model was utilized. The

TABLE 8. Summary of experimental results: Cyberbullying detection in arabic content.

Reference number	dataseset	method	Hyperparameter tuning	Text representation	accuracy	F1-score
[43]	1,00 tweets from Twitter	LOR(unigrams)	-	Unigram and bigram	-	60
[31]	1,000 tweets from Twitter		-	n-gram	-	92
[45]	1,100 tweets from Twitter	FastText	-	Character n-gram	92.2	90
[49]	4,505 tweets from Twitter	GA-SVM	GA	Fine-tuned GloVe, Fine-tuned AraVec	88.2	87.8
[48]	5,340 tweets from Twitter	mBert with CNN	Grid search	AraVec, n-gram, FastText, mBert		87
[18]	10,000 Arabic tweets from Twitter	SVM	Grid Search	BoW vectors	88.1%	0.89
[15]	10,000 Arabic tweets from Twitter	CNN	Grid Search	word embeddings	90.2%	0.91
[42]	10,00 tweets from Twitter	AraBERT	-	TF-IDF, Ara Vec	93	88
[44]	15,050 comments from YouTube	SVM	-	n-gram	84	81
[41]	20,001 tweets from Twitter	MLP	-	encode with TF-IDF using n-gram	92	90
[46]	24,596 tweets from Twitter	FastText	Grid search	Word2Vec	-	-
[47]	30,354 comments from Instagram	CNN	-	Box, bi-gram and tri-gram	-	97.4
Our proposed model	105,371 records from Kaggle	CNN-BiLSTM-GRU	Grid search	Fine-tuned GloVe and FastText word	98.83%	98.13%

model achieved an accuracy of 87.2% and an F-score of 0.88.

In [41], a dataset of 20,001 tweets was used to perform the experiment adopting a Multilayer Perceptron (MLP) model that achieved an accuracy of 92% and an F1-score of 90%. Similarly, in [42], a dataset of 10,000 tweets was subjected to an analysis using the AraBERT model, achieving an accuracy of 93% and an F1-score of 88%. In contrast, in [43], a dataset of 1,000 tweets was employed for cyberbullying detection using the LOR model with unigrams. However, results show a lower performance, with no specified accuracy and an F1-score of 60%.

Transitioning to other platforms, a dataset of 15,050 YouTube comments was explored using a SVM model, yielding an accuracy of 84% and an F1-score of 81% [44]. Furthermore, a dataset of 1,000 tweets from Twitter was assessed using an unspecified method, achieving an accuracy of 92%, but with no F1-score provided [31]. FastText, applied to a dataset of 1,100 tweets from Twitter, led to an accuracy of 92.2% and an F1-score of 90% by utilizing character n-gram features [45]. In [46], FastText was also employed on a dataset of 24,596 tweets from Twitter with the use of Grid search and Word2Vec, but the specific performance metrics were not disclosed. Moving to Instagram, a dataset of 30,354 comments was analyzed using a Convolutional Neural Network (CNN) model that considered box, bi-gram, and tri-gram features, resulting in F-score of 97.4% [47]. Twitter data was again used in [48], where 5,340 tweets underwent cyberbullying detection through a hybrid model of mBERT with CNN, assisted by Grid search for n-gram, AraVec, mBERT, and FastText representations, achieving an accuracy of 87%. Finally, in a separate study documented in [49], a dataset of 4,505 tweets from Twitter was explored using a GA-SVM approach, incorporating fine-tuned AraVec and GloVe embeddings. This approach achieved an accuracy of 88.2% and an F1-score of 87.8%.

The proposed hybrid model, which integrates three advanced neural network models: GRU, CNN, and BiLSTM, is applied to a dataset consisting of 266,747 records. A grid search technique is utilized for hyperparameter tuning in addition to fine-tuned GloVe and FastText word embeddings as its text representation approach. The model achieved accuracy of 98.83% and an F1-score of 98.81%. Results demonstrate the model's capacity in effectively recognizing cyberbullying within the given dataset. This outcome emphasizes the value of integrating multiple advanced neural network structures and finely tuned word embeddings to enhance the accuracy of cyberbullying detection algorithms.

In comparison to the existing cyberbullying detection models in Arabic content, the proposed model offers multiple key improvements. The novel architecture that integrates GRU, CNN, and BiLSTM models utilize their strengths to capture complex patterns in textual data effectively. This integration enhances the accuracy of cyberbullying detection by enabling the model to catch subtle nuances in cyberbullying content.

Moreover, the proposed model adopts advanced text representation using fine-tuned GloVe and FastText word embeddings, which play a key role in providing rich semantic information from words. This sophisticated representation allows the model to better understand the nuanced context of cyberbullying content compared to methods relying on other representations such as n-grams or Bag-of-Words (BoW) vectors.

Additionally, the hyperparameter tuning process conducted through grid search ensures optimization for performance. By systematically exploring a range of hyperparameter values, the proposed model achieves high accuracy and F1-score.

Furthermore, the utilization of a large dataset comprising 266,747 records enhances the model's generalization capabilities and facilitates robust learning of representations. Compared to many existing methods, which operate on smaller and less diverse datasets, our model benefits from ample data volume, ultimately improving its overall performance in cyberbullying detection tasks.

VI. CONCLUSION

This research aims at developing an effective Arabic cyberbullying detection model by integrating the strengths of DL algorithms. Results highlighted that the CNN, BiLSTM, and GRU models demonstrated superior accuracy scores, prompting the selection of a hybrid approach incorporating these models. Each of the three integrated models: CNN, BiLSTM, and GRU brings unique attributes to enhance text cyberbullying classification. CNN, originally designed for image recognition, proved their effectiveness in capturing local patterns and features within text sequences. BiLSTM, a bidirectional recurrent neural network, can understand contextual nuances due to its ability to process sequences in both directions. GRU exhibited memory efficiency while still maintaining robust performance. While constructing an optimal hybrid model, activation functions and optimizers were rigorously evaluated. The combination of the Sigmoid activation function with the Adam optimizer achieved the best results among the examined configurations. This combination, along with the inclusion of ReLU as an activation layer in hidden layers, was adopted for the proposed hybrid model to effectively balance between information retention and computational efficiency.

The proposed hybrid model employed stacked word embeddings, convolutional and recurrent layers, as well as fully connected components. The model's architecture was constructed to utilize the advantages of CNNs, BiLSTM, and GRU while addressing their individual limitations. The integration of these models resulted in a comprehensive solution for accurately identifying Arabic cyberbullying.

The performance evaluation of the hybrid CNN-BiLSTM-GRU model on the collected dataset demonstrated its effectiveness in cyberbullying detection with an impressive accuracy of 98.83%, high recall, precision, specificity, and F1-Score.

ACKNOWLEDGMENT

The authors would like to thank Palestine Technical University—Kadoorie (PTUK) for their support.

REFERENCES

[1] K. Maity, S. Bhattacharya, S. Saha, and M. Seera, "A deep learning framework for the detection of Malay hate speech," *IEEE Access*, vol. 11, pp. 79542–79552, 2023, doi: 10.1109/ACCESS.2023.3298808.

[2] M. Fazil and M. Abulaish, "A hybrid approach for detecting automated spammers in Twitter," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 11, pp. 2707–2719, Nov. 2018, doi: 10.1109/TIFS.2018.2825958.

[3] E. Sarac Essiz and M. Oturakci, "Artificial bee colony–based feature selection algorithm for cyberbullying," *Comput. J.*, vol. 64, no. 1, pp. 305–313, Nov. 2019, doi: 10.1093/comjnl/bxaa066.

[4] P. K. Roy, A. K. Tripathy, T. K. Das, and X.-Z. Gao, "A framework for hate speech detection using deep convolutional neural network," *IEEE Access*, vol. 8, pp. 204951–204962, 2020, doi: 10.1109/ACCESS.2020.3037073.

[5] S. Salawu, Y. He, and J. Lumsden, "Approaches to automated detection of cyberbullying: A survey," *IEEE Trans. Affect. Comput.*, vol. 11, no. 1, pp. 3–24, Jan. 2020, doi: 10.1109/TAFFC.2017.2761757.

[6] S. Wang, X. Zhu, W. Ding, and A. A. Yengejeh, "Cyberbullying and cyberviolence detection: A triangular user-activity-content view," *IEEE/CAA J. Autom. Sinica*, vol. 9, no. 8, pp. 1384–1405, Aug. 2022, doi: 10.1109/JAS.2022.105740.

[7] M. Al-Hashedi, L.-K. Soon, H.-N. Goh, A. H. L. Lim, and E.-G. Siew, "Cyberbullying detection based on emotion," *IEEE Access*, vol. 11, pp. 53907–53918, 2023, doi: 10.1109/ACCESS.2023.3280556.

[8] R. Zhao and K. Mao, "Cyberbullying detection based on semantic-enhanced marginalized denoising auto-encoder," *IEEE Trans. Affect. Comput.*, vol. 8, no. 3, pp. 328–339, Jul. 2017, doi: 10.1109/TAFFC.2016.2531682.

[9] T. H. Teng and K. D. Varathan, "Cyberbullying detection in social networks: A comparison between machine learning and transfer learning approaches," *IEEE Access*, vol. 11, pp. 55533–55560, 2023, doi: 10.1109/ACCESS.2023.3275130.

[10] M. T. Hasan, M. A. E. Hossain, M. S. H. Mukta, A. Akter, M. Ahmed, and S. Islam, "A review on deep-learning-based cyberbullying detection," *Future Internet*, vol. 15, no. 5, p. 179, May 2023, doi: 10.3390/fi15050179.

[11] A. Muneer and S. M. Fati, "A comparative analysis of machine learning techniques for cyberbullying detection on Twitter," *Future Internet*, vol. 12, no. 11, p. 187, Oct. 2020, doi: 10.3390/fi12110187.

[12] M. Raj, S. Singh, K. Solanki, and R. Selvanambi, "An application to detect cyberbullying using machine learning and deep learning techniques," *Social Netw. Comput. Sci.*, vol. 3, no. 5, p. 401, Jul. 2022, doi: 10.1007/s42979-022-01308-5.

[13] W. A. Al-Khater, S. Al-Maadeed, A. A. Ahmed, A. S. Sadiq, and M. K. Khan, "Comprehensive review of cybercrime detection techniques," *IEEE Access*, vol. 8, pp. 137293–137311, 2020, doi: 10.1109/ACCESS.2020.3011259.

[14] T. R. Soomro and M. Hussain, "Social media-related cybercrimes and techniques for their prevention," *Appl. Comput. Syst.*, vol. 24, no. 1, pp. 9–17, May 2019.

[15] B. Haidar, M. Chamoun, and A. Serhrouchni, "Arabic cyberbullying detection: Using deep learning," in *Proc. 7th Int. Conf. Comput. Commun. Eng. (ICCCE)*, Sep. 2018, pp. 284–289, doi: 10.1109/ICCCE.2018.8539303.

[16] H. B. Aji and E. B. Setiawan, "Detecting hoax content on social media using bi-LSTM and RNN," *Building Informat., Technol. Sci.*, vol. 5, no. 1, pp. 114–125, Jun. 2023, doi: 10.47065/bits.v5i1.3585.

[17] A. M. Alduailaj and A. Belghith, "Detecting Arabic cyberbullying tweets using machine learning," *Mach. Learn. Knowl. Extraction*, vol. 5, no. 1, pp. 29–42, Jan. 2023, doi: 10.3390/make5010003.

[18] D. Mouheb, R. Albarghash, M. F. Mowakeh, Z. A. Aghbari, and I. Kamel, "Detection of Arabic cyberbullying on social networks using machine learning," in *Proc. IEEE/ACS 16th Int. Conf. Comput. Syst. Appl. (AICCSA)*. Abu Dhabi, United Arab Emirates: IEEE, Nov. 2019, pp. 1–5, doi: 10.1109/aiccsa47632.2019.9035276.

[19] D. Mouheb, M. H. Abushamleh, M. H. Abushamleh, Z. A. Aghbari, and I. Kamel, "Real-time detection of cyberbullying in Arabic Twitter streams," in *Proc. 10th IFIP Int. Conf. New Technol., Mobility Secur. (NTMS)*, Jun. 2019, pp. 1–5, doi: 10.1109/NTMS.2019.8763808.

[20] M. A. Al-Ajlan and M. Ykhlef, "Optimized Twitter cyberbullying detection based on deep learning," in *Proc. 21st Saudi Comput. Soc. Nat. Comput. Conf. (NCC)*, Apr. 2018, pp. 1–5, doi: 10.1109/NCG.2018.8593146.

[21] T. Mahlangu and C. Tu, "Deep learning cyberbullying detection using stacked embbedings approach," in *Proc. 6th Int. Conf. Soft Comput. Mach. Intell. (ISCMI)*. Johannesburg, South Africa: IEEE, Nov. 2019, pp. 45–49, doi: 10.1109/ISCMI47871.2019.9004292.

[22] K. S. Alam, S. Bhowmik, and P. R. K. Prosun, "Cyberbullying detection: An ensemble based machine learning approach," in *Proc. 3rd Int. Conf. Intell. Commun. Technol. Virtual Mobile Netw. (ICICV)*. Tirunelveli, India: IEEE, Feb. 2021, pp. 710–715, doi: 10.1109/ICICV50876.2021.9388499.

[23] Y. Luo, X. Zhang, J. Hua, and W. Shen, "Multi-featured cyberbullying detection based on deep learning," in *Proc. 16th Int. Conf. Comput. Sci. Educ. (ICCSE)*. Lancaster, U.K.: IEEE, Aug. 2021, pp. 746–751, doi: 10.1109/ICCSE51940.2021.9569270.

[24] J. Yadav, D. Kumar, and D. Chauhan, "Cyberbullying detection using pre-trained BERT model," in *Proc. Int. Conf. Electron. Sustain. Commun. Syst. (ICESC)*. Coimbatore, India: IEEE, Jul. 2020, pp. 1096–1100, doi: 10.1109/ICESC48915.2020.9155700.

[25] Md. T. Ahmed, M. Rahman, S. Nur, A. Islam, and D. Das, "Deployment of machine learning and deep learning algorithms in detecting cyberbullying in Bangla and romanized Bangla text: A comparative study," in *Proc. Int. Conf. Adv. Electr., Comput., Commun. Sustain. Technol. (ICAECT)*. Bhilai, India: IEEE, Feb. 2021, pp. 1–10, doi: 10.1109/ICAECT49130.2021.9392608.

[26] M. Mahat, "Detecting cyberbullying across multiple social media platforms using deep learning," in *Proc. Int. Conf. Advance Comput. Innov. Technol. Eng. (ICACITE)*. Greater Noida, India: IEEE, Mar. 2021, pp. 299–301, doi: 10.1109/ICACITE51222.2021.9404736.

[27] C. Iwendi, G. Srivastava, S. Khan, and P. K. R. Maddikunta, "Cyberbullying detection solutions based on deep learning architectures," *Multimedia Syst.*, vol. 29, no. 3, pp. 1839–1852, Jun. 2023, doi: 10.1007/s00530-020-00701-5.

[28] K. Dubey, R. Nair, Mohd. U. Khan, and Prof. S. Shaikh, "Toxic comment detection using LSTM," in *Proc. 3rd Int. Conf. Adv. Electron., Comput. Commun. (ICAECC)*. Bengaluru, India: IEEE, Dec. 2020, pp. 1–8, doi: 10.1109/ICAECC50550.2020.9339521.

[29] A. T. Aind, A. Ramnaney, and D. Sethia, "Q-bully: A reinforcement learning based cyberbullying detection framework," in *Proc. Int. Conf. for Emerg. Technol. (INCET)*. Belgaum, India: IEEE, Jun. 2020, pp. 1–6, doi: 10.1109/INCET49848.2020.9154092.

[30] Y. Yadav, P. Bajaj, R. K. Gupta, and R. Sinha, "A comparative study of deep learning methods for hate speech and offensive language detection in textual data," in *Proc. IEEE 18th India Council Int. Conf. (INDICON)*. Guwahati, India: IEEE, Dec. 2021, pp. 1–6, doi: 10.1109/INDICON52576.2021.9691704.

[31] 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, doi: 10.1016/j.dss.2018.06.009.

[32] E. Lee, F. Rustam, P. B. Washington, F. E. Barakaz, W. Aljedaani, and I. Ashraf, "Racism detection by analyzing differential opinions through sentiment analysis of tweets using stacked ensemble GCR-NN model," *IEEE Access*, vol. 10, pp. 9717–9728, 2022, doi: 10.1109/ACCESS.2022.3144266.

[33] A. G. D'Sa, I. Illina, and D. Fohr, "BERT and fastText embeddings for automatic detection of toxic speech," in *Proc. Int. Multi-Conf., Org. Knowl. Adv. Technol. (OCTA)*. Tunis, Tunisia: IEEE, Feb. 2020, pp. 1–5, doi: 10.1109/OCTA49274.2020.9151853.

[34] L. Jiang and Y. Suzuki, "Detecting hate speech from tweets for sentiment analysis," in *Proc. 6th Int. Conf. Syst. Informat. (ICSAI)*. Shanghai, China: IEEE, Nov. 2019, pp. 671–676, doi: 10.1109/ICSAI48974.2019.9010578.

[35] H. Mohaouchane, A. Mourhir, and N. S. Nikolov, "Detecting offensive language on Arabic social media using deep learning," in *Proc. 6th Int. Conf. Social Netw. Anal., Manage. Secur. (SNAMS)*. Granada, Spain: IEEE, Oct. 2019, pp. 466–471, doi: 10.1109/SNAMS.2019.8931839.

[36] M. Khairy. (2021). *Arabic-Abusive-Datasets*. Accessed: Jan. 4, 2024. [Online]. Available: https://github.com/omammar167/Arabic-Abusive-Datasets

[37] H. Hermessi. *Arabic Levantine Hate Speech Detection*. Accessed: May 9, 2024. [Online]. Available: https://www.kaggle.com/datasets/haithemhermessi/arabic-levantine-hate-speech-detection

[38] M. Saad. (2021). *Arabic Sentiment Twitter Corpus Positive and Negative Tweets Collected From Twitter*. Kaggle. Accessed: Aug. 23, 2023. [Online]. Available: https://www.kaggle.com/datasets/mksaad/arabic-sentiment-twitter-corpus

[39] R. ALBayari and S. Abdallah, "Instagram-based benchmark dataset for cyberbullying detection in Arabic text," *Data*, vol. 7, no. 7, p. 83, Jun. 2022, doi: 10.3390/data7070083.

[40] A. Alakrot, L. Murray, and N. S. Nikolov, "Dataset construction for the detection of anti-social behaviour in online communication in Arabic," *Proc. Comput. Sci.*, vol. 142, pp. 174–181, 2018, doi: 10.1016/j.procs.2018.10.473. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050918321756

[41] S. Sadiq, A. Mehmood, S. Ullah, M. Ahmad, G. S. Choi, and B.-W. On, "Aggression detection through deep neural model on Twitter," *Future Gener. Comput. Syst.*, vol. 114, pp. 120–129, Jan. 2021, doi: 10.1016/j.future.2020.07.050.

[42] A. Keleg, S. R. El-Beltagy, and M. Khalil, "ASU_OPTO at OSACT4—Offensive language detection for arabic text," in *Proc. 4th Workshop Open-Source Arabic Corpora Process. Tools Shared Task Offensive Lang. Detection*, Marseille, France, May 2020, pp. 66–70. [Online]. Available: https://aclanthology.org/2020.osact-1.10

[43] H. Mubarak, K. Darwish, and W. Magdy, "Abusive language detection on Arabic social media," in *Proc. 1st Workshop Abusive Lang.*, Aug. 2017, pp. 52–56, doi: 10.18653/v1/w17-3008.

[44] A. Alakrot, M. Fraifer, and N. S. Nikolov, "Machine learning approach to detection of offensive language in online communication in Arabic," in *Proc. IEEE 1st Int. Maghreb Meeting Conf. Sci. Techn. Autom. Control Comput. Eng. MI-STA*. Tripoli, Libya: IEEE, May 2021, pp. 244–249, doi: 10.1109/MI-STA52233.2021.9464402.

[45] H. Mubarak and K. Darwish, "Arabic offensive language classification on Twitter," in *Social Informatics* (Lecture Notes in Computer Science), vol. 11864, I. Weber, K. M. Darwish, C. Wagner, E. Zagheni, L. Nelson, S. Aref, and F. Flöck, Eds., Cham, Switzerland: Springer, 2019, pp. 269–276, doi: 10.1007/978-3-030-34971-4_18.

[46] V. K. Jha and V. Vijayan, "DHOT-repository and classification of offensive tweets in the Hindi language," *Proc. Comput. Sci.*, vol. 171, pp. 2324–2333, 2020, doi: 10.1016/j.procs.2020.04.252. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050920312448

[47] H. Karayiğit, Ç. İ Acı, and A. Akdağlı, "Detecting abusive Instagram comments in Turkish using convolutional neural network and machine learning methods," *Expert Syst. Appl.*, vol. 174, Jul. 2021, Art. no. 114802, doi: 10.1016/j.eswa.2021.114802.

[48] S. Alsafari, S. Sadaoui, and M. Mouhoub, "Hate and offensive speech detection on Arabic social media," *Online Social Netw. Media*, vol. 19, Sep. 2020, Art. no. 100096, doi: 10.1016/j.osnem.2020.100096.

[49] F. Shannaq, B. Hammo, H. Faris, and P. A. Castillo-Valdivieso, "Offensive language detection in Arabic social networks using evolutionary-based classifiers learned from fine-tuned embeddings," *IEEE Access*, vol. 10, pp. 75018–75039, 2022, doi: 10.1109/ACCESS.2022.3190960.

EMAN-YASER DARAGHMI received the B.S. degree in communication and information technology from Al-Quds Open University, in 2008, and the M.S. degree in computer science and the Ph.D. degree in computer science and engineering from National Chiao Tung University, Taiwan, in 2011 and 2015, respectively. She is currently the Dean of the Integrated Education College, Palestine Technical University—Kadoorie (PTUK), Tulkarm, where she is also an Associate Professor with the Department of Computer Science. Her current research interests include artificial intelligence, machine learning, distributed and cloud computing, and blockchain. She is passionate about how artificial intelligence can support day-to-day activities, such as digital assistants. Her research automatic medicine reminder app is a smart medicine reminder assistant that is designed to help patients in taking their medications at the correct time and in the correct amount and hence reducing the time to recover from their diseases. Also, she works on utilizing the artificial intelligence techniques to predict the user health condition via analyzing the user iris image (deep learning). Moreover, her project with expert systems aims at developing a web-based, with Arabic interface expert system that integrates all the techniques that an intelligent doctor may use in diagnosis a patient in one expert tool. She serves as a technical program committee member for several conferences and a reviewer for highly distinguished journals.

SAJIDA QADAN is currently pursuing the master's degree with the Department of Cybercrimes and Digital Evidence Analysis, PTUK. Her research interests include deep learning, machine learning, and cybercrimes.

YOUSEF-AWWAD DARAGHMI received the bachelor's degree in electrical engineering from An-Najah National University, Palestine, in 2002, and the master's degree in computer science and information engineering and the Ph.D. degree in computer science and engineering from National Chiao Tung University, Taiwan, in 2007 and January 2014, respectively. He is currently an Associate Professor with the Computer Systems Engineering Department, Palestine Technical University—Kadoorie. His research interests include blockchain, intelligent transportation systems, and vehicular ad-hoc networks. He received the Best Paper Award from ITST, in 2012. He serves as a TPC member for several conferences and a reviewer for highly distinguished journals.

RAMI YOUSUF received the bachelor's degree in electrical engineering from An-Najah National University, in 1999, the master's degree in scientific computing from Birzeit University, in 2005, and the Ph.D. degree in computer science—artificial intelligence from University Kebangsaan Malaysia (UKM), Malaysia, in 2019. He is currently an Assistant Professor with the Computer Engineering Department, Palestine Technical University—Kadoorie, Tulkarm, Palestine. He is also the Dean of Palestine Technical College. He has developed many professional disciplines that keep up with the local market and its need for professionals specialized in all technological and industrial fields. His primary research interests include artificial intelligence applications, remote control, expert systems, neural networks, machine learning, and deep learning.

OMAR CHEIKHROUHOU received the B.S., M.S., and Ph.D. degrees in computer science from the National School of Engineers of Sfax, Tunisia, in March 2012. He is currently an Assistant Professor with the Higher Institute of Computer Science of Mahdia, University of Monastir, Tunisia. He is also a member of the CES Laboratory (Computer and Embedded System), University of Sfax, National School of Engineers of Sfax. His Ph.D. deals with security in wireless sensor networks and more precisely in "Secure group communication in wireless sensor networks." His current research interests include wireless sensor networks, the IoT security, cybersecurity, blockchain, multi-robot system coordination, and the Internet of Drones. He has several publications in several high-quality international journals and conferences. He has received some awards, including the "Governor Prize" from the Governor of Sfax, in 2005. He was ranked among the top 2% of the most widely cited scientists in the world.

MOHAMMED BAZ received the Ph.D. degree in applications of statistical inference on designing communication protocols for low-power wireless networks from the University of York, in 2015. He is currently an Associate Professor with the Computer Engineering Department, College of Computers and Information Technology, Taif University. He is the author of a number of published articles in recognized conferences. He has been a member of multiple committees related to academic fields and a participant in research projects. Moreover, he has taught several courses and supervised several capstone projects. He acted as a Reviewer for a number of the IEEE journals, including IEEE Transactions on Vehicular Technology, IEEE Access, and the IEEE Wireless Communications Letters.