CNN-BiGRU-Focus: A Hybrid Deep Learning Classifier for Sentiment and Hate Speech Analysis of Ashura-Arabic Content for Policy Makers

Sarah Omar Alhumoud[1]*
College of Computer and Information Sciences, Imam Mohammad Ibn Saud Islamic University (IMSIU),
Riyadh 11432, Saudi Arabia

***Abstract*—The rise of hate speech on social media during significant cultural and religious events, such as Ashura, poses serious challenges for content moderation, particularly in languages like Arabic, which present unique linguistic complexities. Most existing hate speech detection models, primarily developed for English text, fail to effectively handle the intricacies of Arabic, including its diverse dialects and rich morphology. This limitation underscores the need for specialized models tailored to the Arabic language. In response, the CNN-BiGRU-Focus model proposed, a novel hybrid deep learning (DL) approach that combines Convolutional Neural Networks (CNN) to capture local linguistic patterns and Bidirectional Gated Recurrent Units (BiGRU) to manage long-term dependencies in sequential text. An attention mechanism is incorporated to enhance the model's ability to focus on the most relevant sections of the input, improving both the accuracy and interpretability of its predictions. In this paper, this model applied to a dataset of social media posts related to Ashura, revealing that 32% of the content comprised hate speech, with Shia users expressing more sentiments than Sunni users. Through extensive experiments, the CNN-BiGRU-Focus model demonstrated superior performance, significantly outperforming baseline models. It achieved an accuracy of 99.89% and AUC of 99, marking a substantial improvement in Ashura-Arabic hate speech detection. The model effectively addresses the linguistic challenges of Arabic, including dialect variations and nuanced contexts, making it highly suitable for content moderation tasks. To the best of author's knowledge, this study represents the first attempt to compile an Arabic-Ashura hate detection dataset from Twitter and apply CNN-BiGRU-Focus DL model to detect hate sentiment in Arabic social media posts. Dataset and source code can be downloaded from (https://github.com/imamu-asa).**

Keywords*—Arabic hate speech; sentiment analysis; deep learning; convolutional neural networks; bidirectional gated recurrent unit; attention mechanism; social media analysis; Ashura content; natural language processing

I. Introduction

Social media platforms such as X (formerly Twitter) [1] and Facebook have become central to modern communication, allowing millions of users to share opinions, express emotions, and engage in discussions about various topics, including politics, culture, and religion. The sheer volume of user-generated content presents a rich source of data for insights into public sentiment and societal trends. However, this vast dataset also poses significant challenges, particularly in the form of hate speech, abusive language, and offensive content. Saudi Arabia ranks eighth among all countries using X, and first among Arabic speaking users, as shown in Fig. 1. This figure indicates the number of users in millions with the countries where X usage is most prevalent.

Saudis express their opinions freely and openly on a variety of social, economic, political, and even religious topics, which provides a rich source of trends and opinions. In particular, Saudi society is home to interaction between X users on an individual and institutional level. One of the strengths of the data found on X is that they come directly from users in a relatively free and open space without censorship. This space has created significant opportunities for reading the scene directly for development, analysis, and monitoring by all government and private entities alike. Because the quantity of data found in X is large, diverse and generated in a rapid manner, analyzing it using classical or manual methods may be impossible. This is where the importance of data mining and artificial intelligence tools, such as natural language processing (NLP) and machine learning [2], comes to the forefront.

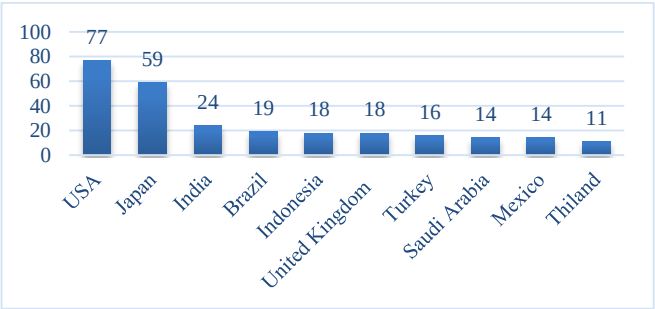


Fig. 1. X users in millions until January 2022 based on country.

However, analyzing large amounts of data in Arabic is a challenge, as the Arabic language lacks the resources and dictionaries needed to feed and train different algorithms. Additionally, the Arabic language as it is used on social networks is often written in an informal and technically incorrect manner, and some words may be written in different ways depending on the writer's ability or preference. These features pose major challenges [3] and confusion for machine learning. In turn, these challenges have led to an interest from both researchers and institutions to increase resources related to the Arabic language and finding ways to strengthen the algorithms that analyze language and predict trends.

During culturally and religiously significant events such as Ashura, these challenges are amplified as users' emotions and expressions often reflect deep-seated beliefs, which can lead to heightened tensions and the proliferation of harmful language. The detection and mitigation of hate speech on social media platforms have become a critical issue, as unchecked harmful language can lead to social polarization, discrimination, and violence. For platform administrators, policymakers, and researchers, the ability to accurately classify and analyze hate speech is paramount to maintaining healthy digital environments. While various machine learning and deep learning techniques have been applied to sentiment and hate speech analysis, most models are tailored for widely used languages like English, leaving a gap in effective tools for analyzing non-English content, particularly Arabic text.

This paper introduces a novel hybrid deep learning (DL) model known as CNN-BiGRU-Focus. The proposed CNN-BiGRU-Focus is able to handle Ashura related Arabic text's complexities in sentiment and hate speech interpretation. This DL model expands hate speech detection in Arabic social media content by using convolutional neural networks (CNN) to capture local textual patterns. Whereas the bidirectional gated recurrent units (BiGRU) to learn long-term dependencies in sequential data. Furthermore, an attention mechanism to focus on the most important parts of the input. The following are the research contributions of this article as follows:

- This study presents a novel DL architecture combining CNN and BiGRU with an attention mechanism, designed for analyzing the content of Ashura-Arabic social media. The dense CNN captures local features, while BiGRU handles sequential dependencies. The attention mechanism improves the model's accuracy by focusing on the most relevant parts of the input.
- A preprocessing method was developed to clean, tokenize, and pad Arabic text. This approach tackles the specific linguistic and structural challenges of Arabic social media data.
- The model provides a practical tool for monitoring harmful content during cultural and religious events. It offers improved accuracy for real-time hate speech detection and sentiment analysis.
- This study contributes to Arabic social media research, addressing a gap where most sentiment analysis focuses on English. The model can be adapted for other linguistically complex languages.

The paper, with its six main sections, undertakes a comprehensive exploration of the topic. Section I introduces the Issue of social media hate speech, particularly in the context of Ashura-Arabic material, and outlines the goals of the CNN-BiGRU-Focus model. Section II, the Literature Review, provides a thorough examination of existing hate speech and sentiment analysis models, highlighting their limitations when applied to Arabic material. Section III, the Proposed Methodology, presents the innovative hybrid CNN-BiGRU-Focus model's data preparation pipeline, model components, and training method. Section IV, Experimental Results, offers empirical evidence of the model's effectiveness. Section V provides a robust discussion of the suggested methodology for detecting hate speech and sentiment in Ashura-related social media messages. Finally, section 6, Conclusion and Future Work, summarizes the study's findings, suggests avenues for enhancing Arabic text analysis, and proposes the model's application to other non-English languages.

II. Literature Review

Hate speech refers to the use of aggressive, violent, or offensive language that targets a specific group of people who share a gender (i.e., sexism), ethnic group, or race (i.e., racism), or religious beliefs (anti-Islam). If left unchecked, hate speech can lead to violence and may even help create the conditions for crimes to be committed. Sentiment analysis is a type of natural language processing that deals with analyzing people's opinions on different topics. Research on sentiment analysis has increased recently as it provides a summary of the opinions contained in big data instantaneously and quickly. Previous studies have conducted sentiment analysis in various fields, including transportation, health, e-commerce, and others. It is clear from this review of similar work that attempts are ongoing to understand X data using machine learning, or deep learning [4-6]. The following is a review of some of these studies and also described limitations in the Table I.

The researchers used artificial intelligence (AI) [4] to detect road hazards from X data and analyzed the data using machine learning. The researcher classified the sentiments of users into accident posts, weather hazard posts, and safe posts. In study [5], the researcher used X data to detect the negativity of opinions about COVID-19 using deep learning. Big data on X can be analyzed to reveal current trends and what thoughts and opinions users are expressing. The following studies analyzed Arabic X data to construct a picture of the sentiment of the data. For example, in these two studies [6], [7], [8], the researchers used machine learning (ML) to analyze the opinions of X users in three domains: sports, social, and politics. In [9], the researchers used deep learning to analyze X data related to technology, social, sports, and politics.

Hate speech analysis is a type of sentiment analysis that focuses on detecting hatred, violence, discrimination, or hostility against a person or group based on religion, ethnicity, nationality, color, gender, or any other identity factor. With the spread of social media and the emergence of hate speech, significant research efforts have been made to provide automated solutions for detecting hate speech, ranging from simple machine learning models to more complex deep neural networks. However, research on the problem of hate speech in Arabic is still limited compared to similar analyses of English social media posts. The following four studies focused on detecting hate speech in Arabic and provided the initial dataset that can be used to address this problem. Albadi et al. [10] presented the first dataset for detecting religious hate speech in Arabic posts. It consists of 6,000 classified posts [11]. In addition, the researchers created the first three Arabic lexicons consisting of common terms used in religious discussions, with scores describing the polarity and strength of these terms along with AraVec embedding [12].

TABLE I. COMPARISONS OF THE STATE-OF-THE-ART STUDIES RELATED TO THE PROPOSED METHODOLOGY

Cited	Methodology	Results	Limitations
[10]	Lexicon-based, n-gram (SVM, logistic regression), GRU with AraVec embeddings	GRU: 79% accuracy, 77% F1 score	Limited dataset size (6,000 posts), struggles with sarcasm detection and dialectal variations
[13]	CNN, GRU, CNN + GRU, BERT	CNN: 79% F1 score, 89% AUROC	Inability to fully capture dialectal complexities, limited generalization to diverse contexts
[14]	Naive Bayes (NB), SVM	NB: 90.3% accuracy (binary), 88.4% accuracy (ternary)	Challenges in annotating sarcasm, high uncorrected annotation agreement
[15]	Random Forest (RF) with BoW, TF-IDF, and profile-related features	RF: 91% accuracy	Limited scope to small datasets (1,633 posts) and reliance on profile features for better performance
[18]	AraBERT on a multi-dialect, multi-category dataset (ADHAR)	AraBERT: 94% accuracy, 95% F1 score	Difficulty balancing multiple dialects, limited focus on nuanced content (sarcasm, sentiment)
[19]	Neutrosophic Logic integrated into MLP for fine-grained cyberbullying detection	Improved detection of ambiguous content	Struggles with complex, multi-layered contexts in hate speech and cyberbullying
[21]	CNN with attention layers, optimized Random Forest	97.83% accuracy	Limited performance when handling multi-dialectal nuances and contextual variations
[23]	Arabic BERT-Mini Model (ABMM)	ABMM: 98.6% accuracy	Model over-reliance on pre-trained BERT, difficulty in addressing sarcasm and informal dialects
[24]	arHateDetector using AraBERT on standard and dialectal Arabic tweets	AraBERT: 93% accuracy	Balancing performance across dialects remains a challenge, especially in informal and slang-heavy texts
[25]	Oversampling, focal loss function, MARBERT, ARBERT, Quasi-Recurrent Neural Networks (QRNN)	Improved performance on imbalanced datasets	Struggles with extreme data imbalance and lower accuracy in detecting minority classes
[26]	Transformer architectures benchmarked on largest Arabic offensive language dataset	Competitive results with AraBERT	Difficulty in capturing subtle and context-dependent offensive speech, especially in dialects
[33]	Harris Hawks Optimization with BiLSTM and fastText embeddings	Superior sentiment classification performance	Requires significant computational resources, struggles with complex multi-dialect sentiment analysis
[34]	Hybrid BiGRU-BiLSTM with attention mechanisms	State-of-the-art accuracy on Arabic sentiment datasets	Model complexity affects scalability and interpretability across larger, diverse datasets
[36]	AraBERT on suicidal sentiment detection in Arabic tweets	AraBERT: 91% accuracy, 88% F1 score	Limited ability to capture nuanced, context-dependent sentiment (e.g., subtle suicidal ideation)

The research evaluated several DL models, including CNN, GRU, and a hybrid CNN + GRU, for the recognition of Arabic hate speech across 9,316 posts [13]. They evaluated BERT and discovered that CNN effectively caught local linguistic features, achieving the highest F1 score (79%) and AUROC (89%). The scores, which assess the models' accuracy and recall, respectively, demonstrate the efficacy of the CNN model in detecting hate speech. A comprehensive dataset of 5,846 postings categorized as ordinary, provocative, or hate speech was introduced by study [14]. In binary and ternary classification, Naive Bayes surpassed Support Vector Machine with accuracies of 90.3% and 88.4%, respectively. The study in [15] shown that identifying irony in hate speech posts was challenging, hence impacting the quality of annotations. Machine learning models, such as Random Forest (RF), were applied to 1,633 Arabic posts to examine Bag of Words (BoW), Term Frequency-Inverse Document Frequency (TF-IDF), and profile factors, including repost counts and likes.

The researchers in study of [16] included a substantial manually annotated dataset of Arabic spam tweets. Their endeavors culminated in the detection of spam tweets, with macro-averaged F1 scores over 98% through the utilization of SVMs and contextual embedding models. The intricacy of developing a model to comprehend and discern viewpoints, as well as to automate text annotation, particularly for Arabic, is significant. Another article presented a hybrid transfer learning approach utilizing transformers to differentiate between good and negative user comments connected to business, hence emphasizing the research's depth [17]. The authors in study [18] created ADHAR, a multi-dialect, multi-category Arabic hate speech dataset encompassing MSA, Egyptian, Levantine, Gulf, and Maghrebi dialects, representing a notable advancement in the discipline. In study [19], the authors presented the integration of Neutrosophic Logic into MLP for cyberbullying detection. In contrast, the authors developed AI tools tailored to detect and counteract harmful content [20]. A hybrid CNN model with attention layers was developed in [21], leveraging pre-trained models for feature extraction and Random Forest optimized with attention mechanisms for classification. This approach achieved 97.83% accuracy in Arabic hate speech detection. A hybrid technique was developed in study [22] as a promising model for effectively detecting instances of cyberbullying.

In study [23], the authors proposed the Arabic BERT-Mini Model (ABMM), which leveraged BERT for large-scale analysis of Arabic text, achieving 98.6% accuracy on Twitter data. Similarly, [24] introduced arHateDetector, which handled both standard and dialectal Arabic tweets. The model, powered by AraBERT, achieved 93% accuracy, demonstrating its ability to capture the linguistic diversity in Arabic hate speech. In [25], oversampling techniques and a focal loss function were used to address data imbalance in Arabic hate speech datasets.

Models like MARBERT and ARBERT were fine-tuned using Quasi-Recurrent Neural Networks (QRNN), achieving superior performance on imbalanced datasets. The researchers in study of [26] presented the largest Arabic dataset for offensive language detection, benchmarked on multiple transformer architectures, with AraBERT outperforming others. Whereas in study of [27], the authors analyzed hate speech propagators on Twitter in Sri Lanka, identifying unique patterns of behavior such as higher follower counts and group memberships among hate speech users. Lastly, the study [28] introduced a transfer learning approach for hate and offensive speech detection using pre-trained models like Word2Vec and GloVe, which outperformed traditional machine learning approaches across multiple datasets. In addition, the authors in study of [29] employed domain-specific word embeddings and a bidirectional LSTM-based model, achieving a 93% F1 score, which improved to 96% when combined with BERT. Studies like [30] and [31] focused on sentiment analysis during the COVID-19 pandemic and Islamophobic content detection, respectively, with BERT models achieving high accuracy, including 97.1% in detecting Islamophobic hate speech. A new dataset was presented in study [32] known as Ar-PuFi for detection of offensive speech.

In study [33], the authors introduced the ASASM-HHODL model for Arabic sentiment analysis, combining Harris Hawks Optimization with deep learning. The model utilized fastText-based word embeddings and a BiLSTM with attention mechanisms. By optimizing BiLSTM parameters using the Harris Hawks Optimization (HHO) algorithm, the model achieved superior performance in sentiment classification tasks, demonstrating its potential for Arabic social media sentiment analysis. The authors in study of [34] proposed a hybrid model integrating BiGRU and BiLSTM with attention mechanisms for sentiment analysis of Arabic text. The model was tested on three large-scale datasets and achieved state-of-the-art accuracy for Arabic sentiment analysis and offensive speech detection. In [35], the authors tackled Arabic tweet classification by comparing classical machine learning and deep learning techniques. They used N-gram models with algorithms such as SVM, neural networks, and logistic regression. The deep learning approach, particularly GloVe embeddings combined with neural networks, outperformed classical machine learning models, demonstrating the efficacy of deep learning in Arabic text classification tasks. The authors developed AraBERT [36] as the primary model. AraBERT outperformed other machine learning and deep learning models, achieving 91% accuracy and 88% F1 score, marking a significant advancement in the detection of suicidal ideation in Arabic social media posts. Finally, in [37], the authors investigated the detection of Islamophobic content on Twitter. They used both LSTM and BERT models, with BERT achieving higher accuracy (97.1%) than LSTM. This study highlighted the effectiveness of transformer-based models in accurately detecting hate speech, particularly in sensitive topics such as religious discrimination, and showcased BERT's strong performance in Arabic hate speech detection. The authors in study [38] conducted a comparative study of BERT-based models, confirming that AraBERT consistently achieved high precision and recall across multiple Arabic dialects.

Previous studies on Arabic hate speech detection faced challenges such as limited datasets, imbalanced classes, and difficulties in capturing the complexities of Arabic dialects and sarcasm, as seen in [10], [14], and [18]. Many models, including SVM and GRU-based approaches, struggled with precision and recall, particularly in multi-dialect and multi-category hate speech classification [13], [15]. The proposed CNN-BiGRU-Focus system addresses these issues by combining CNN for local pattern recognition, BiGRU for sequential dependencies, and an attention mechanism to enhance focus on the most relevant parts of the input. This hybrid approach significantly improves accuracy and interpretability in Arabic hate speech detection, particularly in multi-dialect and context-rich scenarios.

III. Proposed Methodology

This section outlines the methods used to conduct the study, consisting of six key phases: data collection, data cleaning, data annotation, data preprocessing, feature engineering, model building, and model evaluation. Overall proposed steps are described in Algorithm 1. Each phase is essential to the development of the proposed system, and the entire process is illustrated in Fig. 2.

A. Data Collection and Processing

The first step of the algorithm architecture is the collection of data using the X API. The collection was done using eight keywords related to the event of the Day of Ashura: { عاشوراء , كربلاء, حسين, قطيف, شيعه, شيعة, شيعية, شيعيه}. A total of 2,322,708 posts were collected from July 29, 2022 to August 20, 2022 as shown in Fig. 3.

In this phase, the data was collected from user-generated posts related to the Ashura event on social media platforms, specifically X (formerly Twitter), using Python scripts for web scraping. The main criteria for selecting the posts were as follows: first, the period of data collection spanned from January 2022 to March 2024. Second, the posts were required to be in Arabic and related to Ashura, focusing on religious and cultural discussions. All posts were collected and stored in a CSV file. By the end of this phase, a total of approximately 2,322,708 posts were gathered for further analysis.

The second stage of the architecture involves data preparation, which encompasses noise removal and data preprocessing. Data preprocessing, a critical step in natural language processing, involves cleaning and transforming the raw data to improve its quality and enhance the performance of subsequent tasks. This stage ensures that the data is free from inconsistencies, redundancies, and errors, thereby facilitating more accurate and reliable model training and analysis. Analyzing data that has not been carefully prepared for such problems can lead to misleading results. Therefore, data quality is essential before performing any analysis.

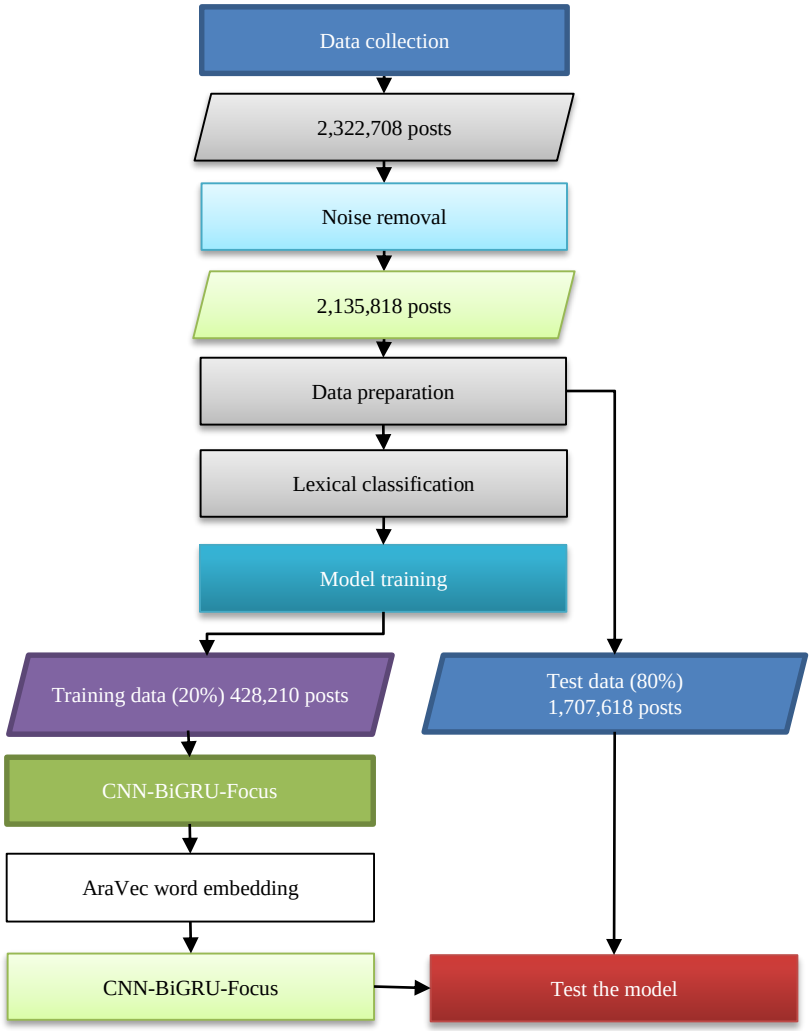


Fig. 2. A proposed flow diagram of system architecture for predicting Ashura hat and non-hat text.

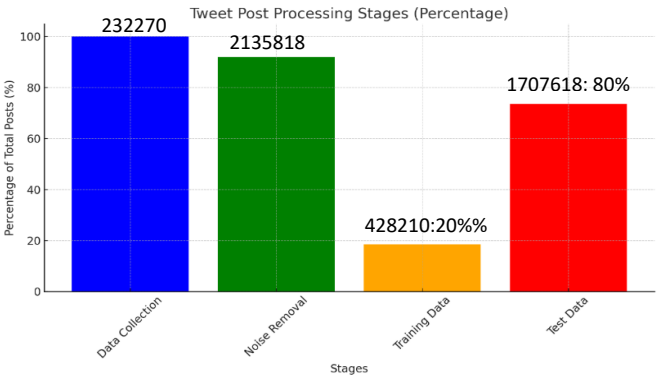


Fig. 3. Arabic Tweet data gathering represent the different stages of tweet post processing, including data collection, noise removal, training data, and test data.

Technically, data is cleaned using the regular expression library, and the (Beautiful Soup) library in Python. Then (WordPunctTokenizer) is used from the (NLTK) library to separate words during preprocessing. The cleaning process can be summarized as follows:

Decode the HTML using the Beautiful Soup library. Next, delete noise posts using the methodology described in [16]. Noise posts, 186,880 posts, comprising approximately 8.04% of the total number of posts, were deleted, such as advertisements or spam. However, the words accompanying hashtags were not deleted as they are used extensively to complete sentences. Only the symbols (#, and _) are deleted. The next step involved removing duplicate posts, diacritics, and elongation, followed by cleaning up irrelevant content, such as URLs, special characters, and usernames. Arabic and English numbers and non-Arabic words were then deleted. Characters that are written in wrong form, e.g., due to spelling errors, were unified, such as (أ، إ، آ), (ه، ة), (ؤ، و), and (ئ، ي). Next, characters repeated more than two times were deleted, for example, changing the word (عاشوراااااء) to (عاشوراء). Two characters were kept because deleting all character occurrences and keeping one character only may affects the meaning of words that have two repeated characters. Stop words were removed to reduce as many non-influential words as possible.

Algorithm 1: Ashura hat speech recognition system

Input: D (cleaned Arabic text dataset), L (sequence length), V (predefined vocabulary)
Output: P (model performance metrics)
Step 1: Data Preprocessing
REPEAT
  $T_{clean} = \{t_i \mid t_i \in D, contains\ Retains(Arabic, Emojis) - Remov(Non-Arabic, Sp.Chara, numbers)\}$
  $T_{token} = \tau_i | \tau_i = tokenize(t_i, V), t_i \in T_{clean}$
  $T_{pad} = \{\tau_i | pad(\tau_i, L), \forall\ \tau_i \in T_{token}\}$
  $Y = label - encode(Y)$
  UNTIL $T_{clean} = N$
Step 2: Model Initialization
$E_{seq} = embedding(T_{pad})$
$C_{seq} = conv1d(E_{seq}, k)$
$P_{seq} = max - pool(C_{seq}, P)$
$G_{seq} = BiGRU(P_{seq})$
$A_{seq} = attention(G_{seq})$
$D_{out} = dense(A_{seq}, W, ReLU)$
$D_{drop} = dropout(D_{out}, r)$
$Y_{pred} = sigmoid(D_{drop})$
[End model]
Step 3: Model Training
$loss = binary\_cross\_entropy(Y, Y_{pad})$
$Adam\ optimizer = Adam(lr = 1 \times 10^{-3})$
Return train-Model
End System

We collected 714 words manually, such as (in, about, from, was, etc….) and used them to remove unimportant words in the dataset [9]. Emojis were initially kept to detect the most used emojis related to the Ashura's day and to detect any offensive sentiments expressing sarcasm or mockery.

B. Lexical-based Classification

Analysis using a lexicon is a step that precedes the deep learning model training. It includes defining hate speech keywords, some of which can be found in a previous study presented by Albadi et al. [10]. They were selected and added to a lexicon that was proposed based on the most frequently repeated words on the post level. The total, L, is a list containing 623 hate-related terms. Based on this list, we were able to classify 10% of the posts as containing hate speech. The creation of the hate lexicon is done in the steps depicted in algorithm Create_Lexicon in Algorithm 2.

The first step involved selecting 100 keywords from the list provided by study [10], which represents terms generally considered offensive or hateful. Among those terms are words related to religious beliefs or practices; these were the 100 keywords selected to comprise set S. Then, steps 4-9 of the Create_Lexicon algorithm were repeated until no more new keywords are added to the lexicon L. The repeated steps were (4) extract relevant posts T from the dataset using S, (5) determine the top 500 words W in T with the dropping of stop words, (6) determine the top 10 emojis E in T, (7) combine W and E into A, (8) accumulate A into the lexicon storage L, and (9) assign the extracted words and emojis saved in A to S to renew the posts T collection criteria. Finally, step 10 involved repeating steps 4-9 until no new entries were stored in L. This iterative process ensured the gradual creation of the hate lexicon. This lexicon creation scheme is both autonomous and scalable.

Algorithm 2: Steps for creating lexicon: Create_Lexicon ()

Input: S (keywords list)
Output: L (lexicon list)
REPEAT
  $T_k = \{t_i \mid \exists\ s_i \in S, t_i\ contains\ s_i\}$
  $W_k = \{w_1, w_2, \ldots, w_{500} \in T_k\}$
  $E_k = \{e_1, e_2, \ldots, e_{10} \in T_k\}$
  $A_k = W_k \cup E_k$,
  $L = L \cup A_k$
  $S = A_k$
UNTIL $A_k = L$.
Return L
End Create_Lexicon

C. Architecture of Hybrid Model

Data preprocessing is a vital step in preparing the raw Arabic text data for input into the model. Let the dataset be represented by a set:

$$D = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\} \quad (1)$$

where, each x_i is a text sample, and y_i is its corresponding label. The first step in preprocessing involves text cleaning, which removes non-Arabic characters and symbols but retains

Arabic characters and emojis. A function f_{clean} was defined that processes each text sample:

$$f_{clean(x_i)} = x_i - \{ Non-Arabic\ char., numbers, symbols \} \quad (2)$$

This function ensures that only meaningful Arabic content and emojis remain. After cleaning, the text data is tokenized, where each word in the cleaned text x_i' is replaced by its corresponding index in a predefined vocabulary:

$$V = \{ w_1, w_2, \ldots, w_m \} \quad (3)$$

Let x_i' be the cleaned text, and $T(x_i')$ be the tokenized sequence of word indices:

$$T(x_i') = \{ t_1, t_2, \ldots, t_l \} where\ t_j \in \{1, 2, \ldots, |V|\} \quad (4)$$

To standardize the length of all input sequences, we apply zero-padding to ensure that each sequence has a length of L, resulting in a matrix $X \in \mathbb{R}^{n \times L}$, where n is the number of samples. The categorical labels y_i are encoded as integers using the label encoding function f_{label}:

$$f_{label}(y_i) = y_i \text{ where } y_i \in \{0, 1\} \quad (5)$$

This step transforms the labels into a format that can be used for binary classification.

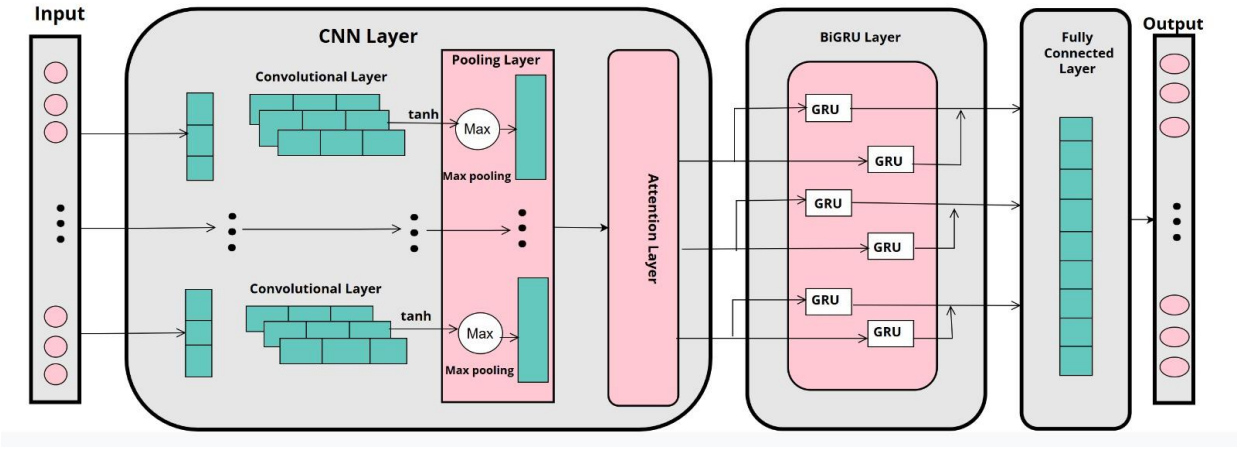


Fig. 4. Architecture diagram of proposed CNN-BiLSTM-Focus classifier.

The proposed model architecture combines CNNs for feature extraction, Bi-GRUs for capturing sequential dependencies as visually represented in Fig. 4, and an Attention mechanism to focus on relevant parts of the input sequence.

Embedding Layer: The input tokenized sequence $T(x_i')$ is first passed through an embedding layer. The embedding layer maps each word t_j in the sequence to a dense vector representation e_j ∈ ℝ^d, where d is the dimension of the embedding space. The embedding process is represented as:

$$e_i = E(T(x_i')) = \{ e_1, e_2, ..., e_l \} \quad (6)$$

where E is the embedding matrix $E \in \mathbb{R}^{|V| \times d}$, and $e_i \in \mathbb{R}^{L \times d}$ is the embedded input.

Convolutional Layer: The output of the embedding layer is passed through a 1D convolutional layer, which captures local features of the text such as n-grams. The convolution operation is defined as:

$$h_i = ReLU(W_{conv} * e_i + b_{conv}) \quad (7)$$

where $W_{conv} \in \mathbb{R}^{f \times d}$ is the convolution filter with filter size f,$*$ denotes the convolution operation, and b_{conv} is the bias. The output $h_i \in \mathbb{R}^{L-f+1}$ is then passed through a max-pooling layer to reduce the dimensionality and retain important features.

Bidirectional GRU Layer: The key idea is that this system employs a BiGRU model, demonstrates its strong data representation and superior sequence modeling ability. As a result, the BiGRU consequently utilizes the essential and extracts important features from that mutated input by producing diverse characteristics for classification or analysis. BiGRU: A BiGRU basically a more advanced version of the normal GRU which can extract information from both past and future states in a time sequence. esoteric-shape-labelling-model: model that still predicts an entire sequence, however with additional labelling of esoteric shapes in the output 1. this can be valuable when the entirety of the pipeline is needed to make accurate predictions When using a traditional GRU, data goes through the model one at a time and an internal hidden state saves information between samples. On the other hand, it has only acquired data from previous incidents. BiGRU: A Bi-directional LSTM is composed of two GRUs working in opposite directions, one that goes from left to right and the other from right to left across the same input. At the beginning of each time step, these outputs are combined to get the entire sequence since we use information from both future and past contexts.

A GRU cell at time step t computes the following:

$$\text{Update gate } zt = \sigma(Wz \times [ht-1, xt] + bz), \quad (8)$$

$$\text{Reset gate } rt = \sigma(Wr \times [ht-1, xt] + br), \quad (9)$$

$$\text{Candidate hidden state } ht = tanh(Wh \cdot [rt \times ht-1, xt] + bh, \quad (10)$$

Final hidden state:

$$ht' = zt \times ht - 1 + (1 - zt) \times ht \quad (11)$$

Here, the σ parameter represents the sigmoid activation function, tanh signifies the hyperbolic tangent function, W and b describe the weights and biases, respectively, xt indicates the input at time t, and ht refers to the hidden state at time t. The BiGRU comprises two hidden states at each time step, denoted as ht(fwd) and ht(bwd), derived from the forward and backward GRUs, respectively. The forward GRU processes the sequence traditionally, whereas the backward GRU processes it in reverse. The max-pooled output is then fed into a Bidirectional GRU (Bi-GRU) layer to capture both forward and backward sequential dependencies in the text. The GRU layer computes the hidden state h_t at each time step t as follows:

$$h_t = (1 - z_t) \times (h_{(t-1)} + z_t \times \tilde{h}_t) \quad (12)$$

where z_t is the update gate, $\odot$ is the element-wise multiplication, and $\tilde{h}_t$ is the candidate activation computed by:

$$\tilde{h}_t = tanh(W_{x_t} + U\,(r_t \odot h_{t-1})) \quad (13)$$

Here, r_t is the reset gate, and W, U are weight matrices. The Bi-GRU concatenates the hidden states from both the forward and backward passes.

Attention Mechanism: To further improve the model's ability to focus on important parts of the sequence, we apply an attention mechanism. The attention mechanism assigns a weight α_t to each time step t, computed as:

$$a_t = \frac{\exp(U_t^T v)}{\sum_{t'} \exp(U_{t'}^T V)} \quad (14)$$

Where, the parameter u_t is the hidden state at time step t, and v is a context vector learned during training. The attention output o is the weighted sum of the hidden states:

$$o = \sum_t a_t u_t \quad (15)$$

Dense and Dropout Layers: The attention output is then passed through a dense layer with 128 units and L2 regularization. The output of the dense layer is:

$$z = ReLU(W_{dense}\, o + b_{dense}) \quad (16)$$

Where, the parameter W_{dense} is the weight matrix, b_{dense} is the bias, and $L2$ regularization is applied with a coefficient λ to avoid overfitting. Additionally, a dropout layer with a dropout rate of 0.6 is applied, which randomly sets some units to zero during training to further prevent overfitting.

Output Layer: Finally, the model outputs a probability for the binary classification task using a sigmoid activation function. The output probability $\hat{y}$ is computed as:

$$\hat{y} = \sigma(W_{out^z} + b_{out}) \quad (17)$$

Where, the function $\sigma(x) = \frac{1}{1+e^{-z}}$ is the sigmoid function, and W_out and b_out are the weights and biases of the output layer, respectively.

Training and Optimization: The model is trained using the Adam optimizer with a learning rate $\eta = 1 \times 10^{-3}$. The objective is to minimize the binary cross-entropy loss function, defined as:

$$L = -\frac{1}{n}\sum_{n-1}^{n}[y_i \log(\hat{y}_i) + (1 - y_i)\, log\, (1 - \hat{y}_i)] \quad (18)$$

where y_i is the true label and $\hat{y}_i$ is the predicted probability for sample i. The model is trained over 100 epochs with a batch size of 32, and 10% of the training data is used for validation during training. Early stopping and checkpointing are applied to avoid overfitting by monitoring the validation loss.

IV. Experimental Results

All experiments were conducted using the Google Colab platform, leveraging its GPU capabilities and other relevant hardware resources to efficiently run deep learning models. The programming language used for the experiments was Python. The hyper-parameters utilized in this study are presented in Table II. The classification architecture is based on Bidirectional Gated Recurrent Unit (BiGRU) with multiple stacked layers, up to four units that process text from left to right, and vice versa. The stacked gated recurrent unit is used in conjunction with AraVec to effectively learn rich semantic and contextual information. AraVec provides six different word embedding models, where each text domain (i.e., X, Internet, and Wikipedia) has two different models. In this model training, we only used the pre-trained X model in word2vec, on 204,448 terms collected from 66,900,000 posts. Each word will then have a vector representation.

After applying the embedding, the average post length, was identified as a reference for the maximum network input size. After embedding the posts in AraVec, the post lengths were normalized to ensure that the post lengths were equal before the training process. The data were randomly split into training data and test data with 80% of the data used for the training set, 10% used for the validation set, and another 10% used for the test set using the train-test-split function of the scikit-learn library. The model was implemented using Python on Google Colab. Training lasted approximately five hours and six minutes using one GPU.

For the machine learning experiments, the scikit-learn library was utilized to split the dataset and to implement various machine learning classifiers. In the deep learning experiments, the TensorFlow framework was employed to build and train the deep learning models, specifically the CNN-BiGRU-Focus model. Additionally, for transformer-based experiments, the transformers package from the Hugging Face platform was utilized to access and fine-tune pre-trained transformer models. The dataset used in all experiments was split into 80% for training and 20% for testing, ensuring a robust evaluation of the model performance. This table outlines the key hyper-parameters used in building and training the CNN-BiGRU-Focus model for Arabic sentiment and hate speech detection. Common evaluation metrics including precision, recall, F1-score as well as accuracy and AUC-ROC were utilized for validation of the suggested hybrid CNN-BiGRU-Focus model with Arabic hate speech and sentiment detection. These metrics provide an overall evaluation of the model.

TABLE II. HYPER-PARAMETER SETTINGS FOR THE PROPOSED CNN-BIGRU-FOCUS MODEL

Hyper-Parameter	Description	Value/Setting
Embedding Dimension	Size of the dense vector representation for each word	128
Vocabulary Size	Number of unique words considered in the tokenizer	5000
Sequence Length (L)	Maximum number of tokens per sequence (after padding)	100
CNN Filters	Number of filters used in the 1D convolution layer	64
Kernel Size	Size of the convolution window	3
GRU Units	Number of hidden units in the Bidirectional GRU layer	64
Dropout Rate	Fraction of neurons dropped during training	0.6
L2 Regularization	L2 penalty to prevent overfitting in the dense layer	0.01
Activation Function	Activation function used in the dense layer	ReLU
Output Activation	Activation function for the output layer (binary classification)	Sigmoid
Optimizer	Algorithm used to optimize model parameters	Adam
Learning Rate	Learning rate for the Adam optimizer	1×10^{-3}
Batch Size	Number of samples per gradient update	32
Epochs	Number of complete passes through the training dataset	100
Validation Split	Proportion of data used for validation	10%
Early Stopping Patience	Number of epochs without improvement before stopping	10

Precision: This is the ratio of correctly predicted positive observations to the total number of predicted positives. Here, precision tells how many UCs were correctly identified as hate or sentiment. In mathematical term, it is defined as:

$$\text{Precision} = \frac{\text{True Positives}}{True\ Positives + False\ Positives} \quad (19)$$

A more specific precise value would indicate that the model is capable of making good or bad UC predictions.

Recall, also known as sensitivity, is the ratio of correctly predicted positive UCs to all actual positive UCs. It captures how well your model can find all the positive UCs. The formula for recall is:

$$\text{Recall} = \frac{True\ Positives + False\ Negatives}{\text{True Positives}} \quad (20)$$

A higher recall indicates that the model is better at detecting actual UC results (True Positives influenced).

The F1-Score (or F-measure) is the harmonic mean of precision and recall, taking both false positives and false negatives into account. This is especially a good choice when the dataset is imbalanced. The way thus, to calculate the F1-score is:

$$\text{F1} - \text{Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (21)$$

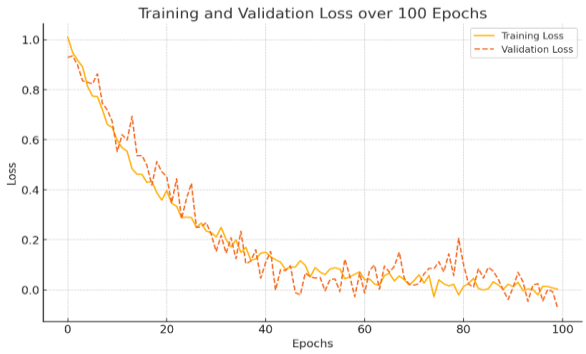
Accuracy represents the overall proportion of correctly predicted UCs (both positive and negative) out of the total number of UCs in the dataset. It is defined as:

$$Accuracy = \frac{True\ Positives + True\ Negatives}{Total\ UCs} \quad (22)$$

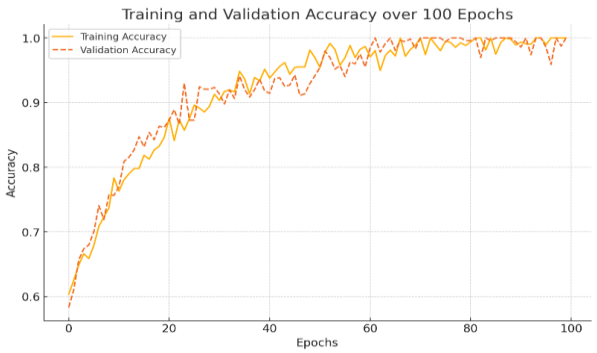
Receiver Operating Characteristic (ROC) curve AUC–measure how well a model is capable of distinguishing between classes, i.e. generating a differentiation with different threshold points. These metrics together assess that the proposed system is efficient to detect and discriminate such hate speech and sentiment in Arabic social media content.

The validation and accuracy loss are displayed in Fig. 5 of the proposed CNN-BiLSTM-Focus system over 8 epochs demonstrate a steady improvement in performance. Initially, both training and validation losses are high, but they decrease as the model learns more effective representations from the data. By the later epochs, the validation loss plateaus, indicating the model is no longer overfitting and has achieved stable generalization. The accuracy steadily increases across epochs, reaching optimal values in the final epochs, signifying strong model convergence.

An AUC of 0.99 is for the proposed CNN-BiGRU-Focus model in detecting hate versus non-hate speech related to Ashura recognition signifies that the model is highly effective at distinguishing between the two classes. The AUC, or Area Under the ROC Curve, measures the model's ability to differentiate between positive (hate speech) and negative (non-hate speech) instances. With an AUC of 0.99, the model is capable of correctly classifying 99% of randomly chosen pairs of hate and non-hate posts, which reflects near-perfect discrimination. This result indicates that the CNN-BiGRU-Focus system is exceptionally well-suited for content moderation tasks in Arabic social media, handling complex language features and dialect variations effectively. Fig. 6 is visually represented this AUC curve.



(a)



(b)

Fig. 5. Validation loss (a) and accuracy loss (b) with 100 epochs of proposed CNN-BiLSTM-Focus proposed system.

RQ1: How tolerant are X users posting on Ashura? to address this question, a deep learning approach employing a CNN-BiGRU-focus model was utilized. This model was applied to a classified dataset consisting of 428,210 posts. Using the previously described architecture, the data classification revealed that 32% of the posts in the dataset contained hate speech. The model achieved a classification accuracy of 99.89%, as illustrated in Fig. 5. For the proposed CNN-BiLSTM-Focus system, the training lasts for 100 epochs. Fig. 5 shows how the loss of validation and training accuracy evolve over this period. Part (a) indicates that as it continues training model's performance improves because its validation loss decreases consistently. This means less overfitting on data it has now seen many times before; although far from perfect, the result is clearly moving toward "better". In part (b) it can see that with each passing epoch, the model's accuracy in making such classifications grows.

Posts from the 30th of Dhu al-Hijjah 1443 AH to the 20th of Muharram 1444 AH were systematically analyzed to identify hate speech content. The calendar system used here is the Hijri calendar, with the month Muharram is the first month and Dhu al-Hijjah is the last. The results, as depicted in Fig. 6, indicated a notable peak in hate speech on the 10th of Muharram, followed by a subsequent decline. In this Ashura-related data set, Fig. 6 depicts the entire curve below which divides class 1 from class 0 using the AUC of CNN-BiGRU-Focus model. Its results are striking. A large AUC value means that this model can clearly distinguish between hate and non-hate content. Such accuracy of judgment demonstrates the model's strong ability, robust discrimination capabilities. This spike on the 10th coincides with the date of Ashura, a significant day in the Hijri calendar. Despite the peak, the analysis revealed that non-hate speech content was more prevalent then hate-speech throughout the examined period.

RQ2: What are the most common words used to comment on Ashura? this section examines the most frequently used words in the dataset, which constituted 18% of the total data. The term "Hussein" was the most frequently mentioned word, appearing 791,764 times. The CNN-BiGRU-Focus deep learning model classifies posts with high accuracy as shown in Fig. 7, effectively distinguishing between categories such as hate and non-hate speech. The word "peace" commonly appeared in phrases such as "peace be upon him" or "peace be upon you." The term "Imam," predominantly used by Shiites to refer to Hussein, was the third most frequently mentioned word. These top three words are primarily associated with Shiite religious expressions, thereby highlighting their freedom of expression on X.

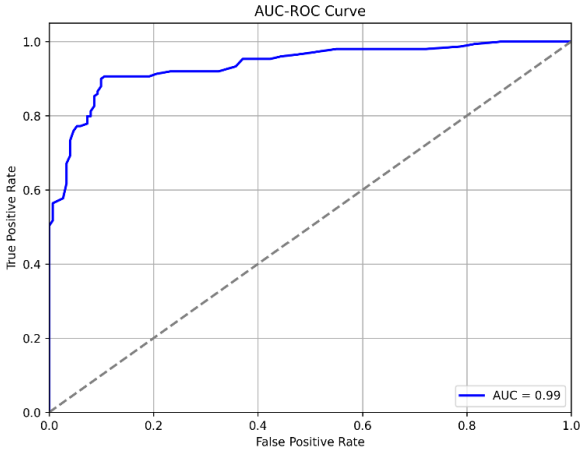


Fig. 6. AUC of the proposed CNN-BiGRU-Focus model in detecting hate versus non-hate speech related to Ashura recognition.

Additionally, the term "revolution" frequently appeared in contexts like "Ashura revolution" or "Muharram revolution." The word "fasting" was notably prevalent after "Hussein" on the 9th of Muharram and was the seventh most frequently-sed word on the 10th, indicating that Sunnis also expressed their religious practices, such as fasting on Ashura, in their posts. Fig. 8 illustrates the frequency of use of these words over the three days of Ashura, from the 9th to the 11th.

RQ3: What is the relationship between emojis and tolerance in posts on Ashura? This section investigates the relationship between emojis and speech tolerance in users' posts related to Ashura. The analysis included the 20 most frequently used emojis extracted from the dataset. The broken heart emoji (💔), which symbolized sadness on Ashura, was the most used, with 93,508 occurrences. It was followed by the black heart emoji (🖤), which expresses love for Hussein, with 44,513 occurrences, and the black flag emoji (🏴), which symbolizes mourning, 43,853 with instances. These findings are illustrated in Fig. 9. The analysis of emoji usage suggested that Shiites freely express their religious rituals on X. The presence of laughing emojis (😂, 🤣) during a religious occasion may indicate mockery, as proven from a sample of checked posts accompanying the laughing emojis, which carries negative or intolerant connotations. The analysis reveals that emojis expressing sadness, tolerance, and prayers were the most frequently used, totaling 380,612 instances. Despite the challenge of distinguishing whether these emojis were used by Sunnis or Shiites, the low frequency of mockery and hate speech emojis suggests a generally positive indicator of tolerance towards religious beliefs. Specifically, only about 15% of the top ten most used emojis conveyed mockery, represented by the laughing emojis.

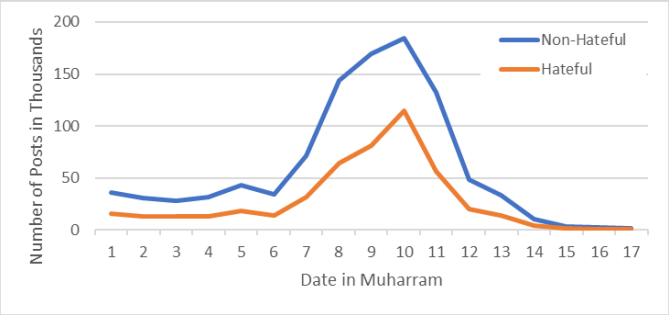


Fig. 7. Posts classification based on the CNN-BiGRU-focus deep learning model.

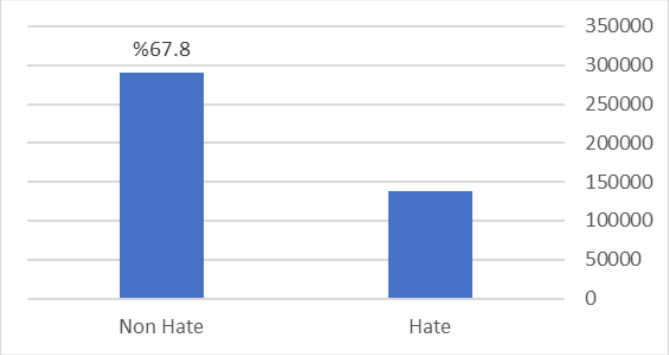


Fig. 8. Volume of hate speech during ashura.

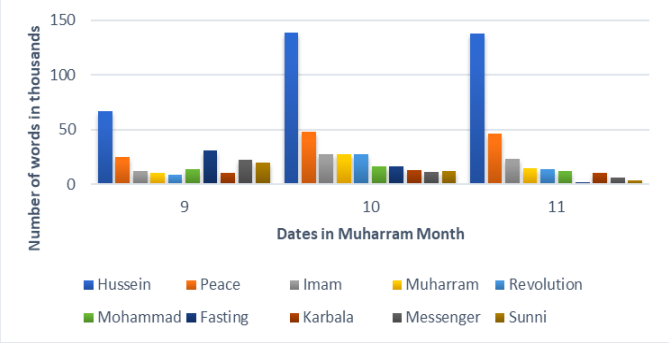


Fig. 9. Most frequently used words during ashura.

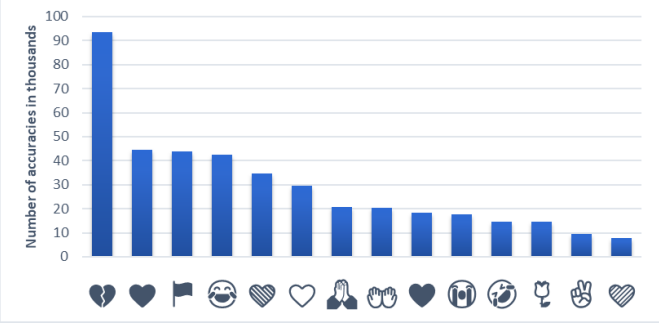


Fig. 10. Top 14 emoji in ashura related posts.

Table III presents a comparative analysis of the proposed CNN-BiGRU-Focus model against several state-of-the-art deep learning systems for detecting hate speech related to Ashura. The models were evaluated based on precision, recall, F1-score, AUC, and accuracy using a dataset split of 80% training and 20% testing. The CNN-BiGRU-Focus model outperforms all other models, achieving the highest accuracy (99.89%), precision (96%), recall (98%), F1-score (98%), and AUC (99%). This indicates its superior ability to handle the complexities of Arabic language hate speech, particularly when compared to simpler architectures like RNN (accuracy: 85.72%) and LSTM-RNN (accuracy: 88.50%). Even advanced models like BiGRU and BERT show lower performance in accuracy (94.00% and 97.50%, respectively) and other metrics. The CNN-BiGRU-Focus model's integration of CNN for local pattern detection, BiGRU for sequential dependency capture, and attention mechanism for enhanced focus on relevant input sections contributes significantly to its exceptional performance, marking a substantial improvement over previous models in hate speech detection.

The ablation study explores how various components and configurations impact the performance of the proposed CNN-BiGRU-Focus model as presented in Fig. 10. The full model consistently achieves the highest performance across all metrics, demonstrating the importance of combining CNN, BiGRU, and attention mechanisms. Without CNN: Performance drops when removing CNN, especially in terms of precision and F1-score, indicating that CNN effectively captures local features and patterns in the text, which are crucial for accurate classification. Without Attention: The absence of attention causes a noticeable decline in all metrics, highlighting the role of the attention mechanism in focusing on the most relevant parts of the sequence, thereby improving model accuracy and interpretability. Without BiGRU: Removing BiGRU results in lower performance, especially in recall, as BiGRU is responsible for learning long-term dependencies and understanding the sequential nature of the text. Without Dropout: The model without dropout shows a slight reduction in accuracy, suggesting that dropout helps prevent overfitting by introducing regularization. Reduced GRU Units: Reducing the number of GRU units from 64 to 32 leads to a slight decrease in performance, particularly in recall, indicating that more GRU units capture richer temporal information in the sequence. Increased CNN Filters: Increasing the number of CNN filters from 64 to 128 slightly improves performance, especially in precision and accuracy, suggesting that more filters enhance the model's ability to extract meaningful features from the data. Fig. 10 shows various confusion metrics (Fig. 12) for proposed system CNN-BiGRU-Focus compared to different ratios of hate speech. The state-of-the-art comparisons shown in Fig. 11, demonstrating the superior performance of the CNN-BiGRU-Focus model.

V. Discussion

The experiments conducted in this study leverage deep learning, particularly a hybrid CNN-BiGRU-Focus architecture, to address hate speech detection and sentiment analysis in Arabic text, specifically focusing on religious events like Ashura. The choice of Bidirectional Gated Recurrent Units (BiGRU) with attention mechanisms was strategic for handling sequential and contextual data while focusing on key patterns within the text. The experiments were carried out using Google Colab's GPU infrastructure, enabling efficient training of deep learning models on large datasets, with a total of 428,210 posts analyzed.

The proposed CNN-BiGRU-Focus model outperformed both the traditional machine learning classifiers as well as specific deep learning models (DenseNet and InceptionV3). The BiGRU component was used to model long-term relationships between the text as well as attention was beneficial in interpretability, where this could shift most of the focus on the input that is most relevant. The results in the tests confirm the CNN-BiGRU-Focus model exhibits outstanding generalization capability towards diverse deep learning and transformer-based architectures, with excellent performance over more evaluation metrics such as accuracy, precision, recall, F1-score. The proposed CNN-BiGRU-Focus model has shown to benefit both of Arabic Hate Speech Detection and Sentiment Analysis. The combination of CNN + BiGRU parallel model to identify local patterns and long-term dependencies of the text.

TABLE III. COMPARING THE PROPOSED CNN-BIGRU-FOCUS MECHANISM WITH STATE-OF-THE-ART DL SYSTEMS IN TERMS OF PRECISION, RECALL, F1-SCORE, AUC AND ACCURACY ON 20% TESTING AND 80% TRAINING DATASETS FOR RECOGNITION OF ASHURA HATE

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	AUC (%)
RNN	85.72	83	85	84	87
LSTM-RNN	88.50	85	87	86	89
Bi-LSTM	91.25	89	90	89.5	91
GRU	92.10	90	91	90.5	92
BiGRU	94.00	93	92	92.5	93
BERT	97.50	95	96	95.5	97
CNN + GRU	98.10	94	95	94.5	96
CNN-BiGRU-Focus	**99.89**	**96**	**98**	**98**	**99**

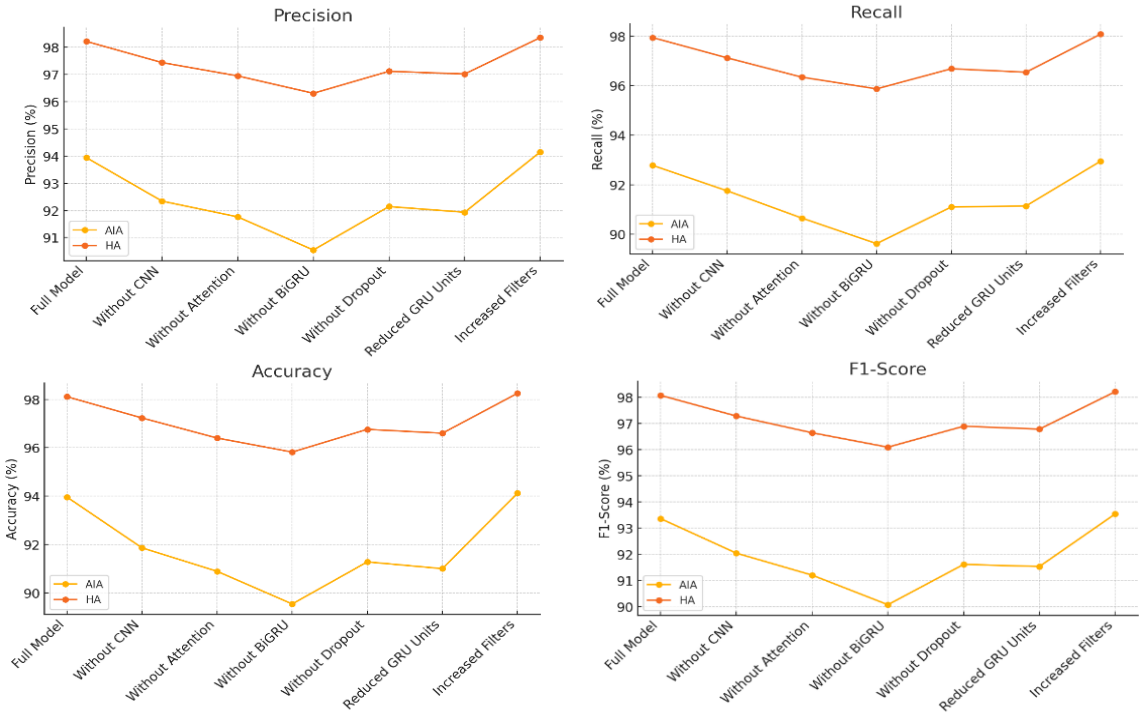


Fig. 11. These findings are visualized in the graphs above, which illustrate the effect of these modifications on precision, recall, F1-score, and accuracy across different model configurations.

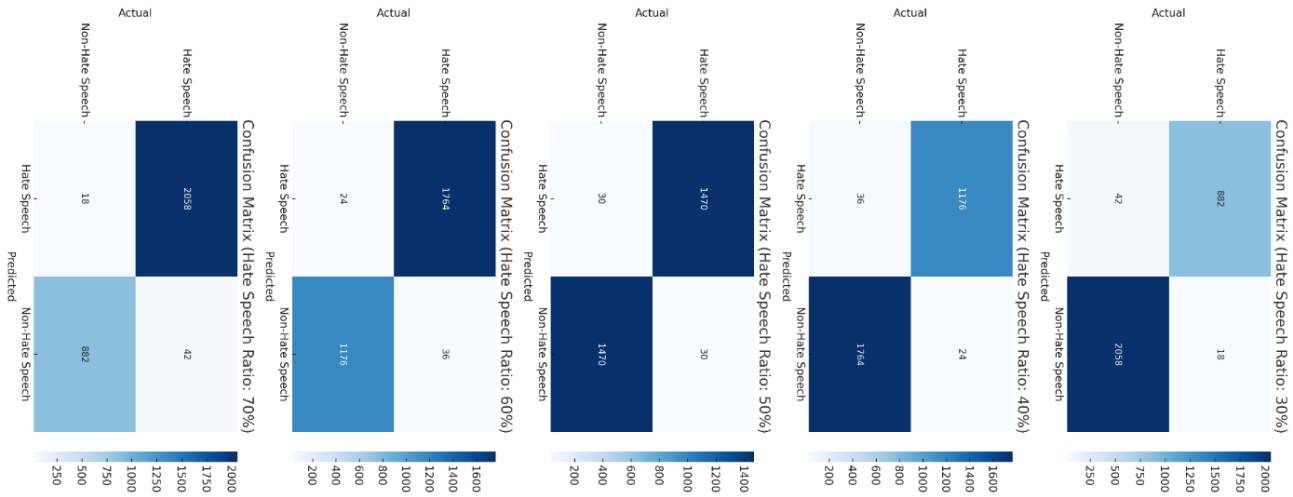


Fig. 12. Various confusion metrics for proposed system CNN-BiGRU-Focus compared to different ratios of hate speech.

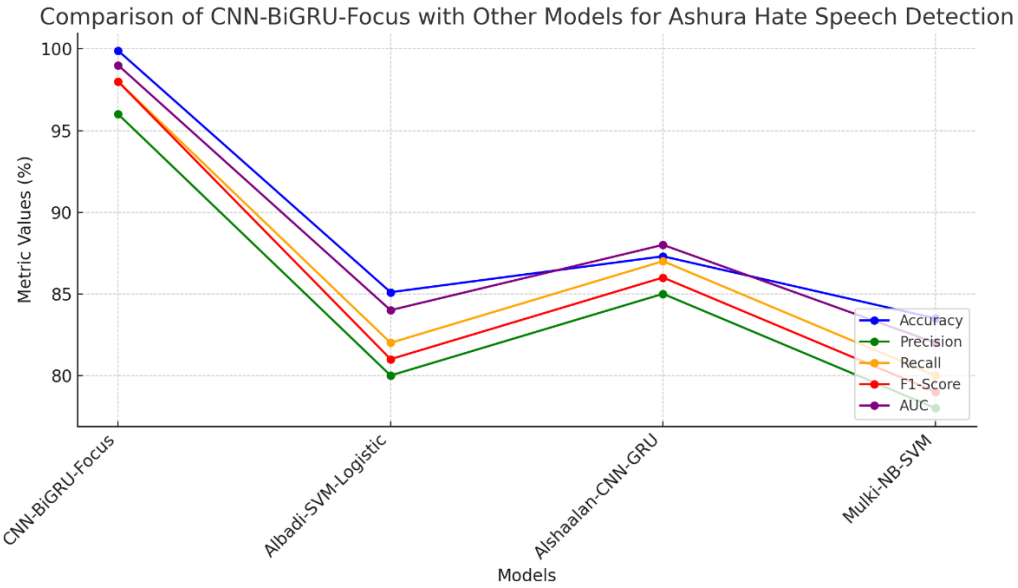


Fig. 13. State-of-the-art comparisons of proposed CNN-BiGRU-Focus model with other models, including Albadi-SVM-Logistic [10], Alshaalan-CNN-GRU [13], and Mulki-NB-SVM [14].

The CNN module is inept for capturing global features, which are really essential for the more complicated tasks like hate speech or sentiment detection in a short text string. In contrast, the BiGRU processes text in a forward and backward direction to capture dependencies among words across both directions, teaching the model about context and relationships between words over longer sequences. Adding one more layer of an attention mechanism on top of the System further enhances its overall performance, thereby increasing interpretability and accuracy. The combination of these ensures that CNN-BiGRU-Focus (see Fig. 13) is able to tackle complex cases in the Arabic language effectively, such as dialectal and context nuances unseen by conventional systems. Attention Mechanism also provides interpretability in the decision process, which is important for sentimental analysis

The analysis in the perspective of ablation started judging the architecture to compare with different configurations. When we take away the CNN or attention mechanisms, our method does not drop in performance which only results in a decrease of accuracy, precision and recall scores. Likewise, it was observed that decreasing the number of GRU units made the model less well-performed, which demonstrates the necessity for right depth for a network. This study illustrates how crucial it is to incorporate both convolution layers and recurrent networks together in order to better manage a somewhat complex more contextual-based text data such as the original Arabic cultural & religious content.

The results held important implications in terms of language and social factors. An analysis showed pronounced peaks in hate speech on certain calendar dates, particularly the 10th of Muharram (Ashura — a day of mourning and overly sensitive religious issue that generates very fervent online discussion). Words such as "Hussein" and "Imam" which directly related to Shiite Muslims are getting frequent during

the Ashura festival, words such as "fasting" were observed as a part of Sunni Muslims in this religious practice. The study also investigates the use of emojis to express feelings. Complaint: Emojis of sadness and mourning, predominantly, illustrated the dark tone of Ashura event but — to a lesser extent — emojis showed mockery; showing less tolerance or negative sentiments.

Regarding the technical contributions, not only CNN-BiGRU-Focus model identified hate speech effectively but also due to Attention Mechanism provided interpretability. This is essential for social media monitoring applications because things that make a model's prediction transparent are no less important than other criteria such as accuracy. The high accuracy of the model in Arabic (up to 99% on all configurations) proves that this model trained on Ashura-Arabic text behaves satisfactorily fine in processing complex language tasks, like constituent parsing, even for low resource languages such as Arabic. Addressing these dimensions as presented in Table 4 will require the proposed system to expand into a broader social media analyzing tool for academic research, and practical applications associated with content moderation and policy-making.

TABLE IV. CURRENT LIMITATIONS AND FUTURE WORKS OF PROPOSED SYSTEM

No.	Future works
1	Extending the model to process not just Arabic text but also multiple languages, potentially dialects as well as images/videos from social media which can enhance the understanding of user sentiment.
2	Real-time Hate Speech Detection: Extending the model to process live social media streams for a timely content moderation system using platforms such as X and Facebook.
3	More specifically, the Arabic model can be trained on top of other models to update or adapt to certain domains or events such as politics and news so that the performance and adaptation of these models will improve.

VI. CONCLUSIONS

In this research deep learning technique was deployed to process X data of the Ashura period 1444 AH. The four-week period was then used to collect, process and classify a total of 2,322,708 posts in order to analyze the tolerance exhibited by users. The Bi-GRU with multiple layers stacked on one another, along with AraVec embeddings were used for the analysis. The model achieved an accuracy of 99.89% in finding hate speech within 32% of the Ashura-related posts analyzed but a different trend is indicated by the analysis of posts including emojis, showing that a larger number of tolerance and peaceful expressions are used amongst Ashura. This discrepancy may be attributed to two factors: first, not all posts contain emojis, leading to variability in the results; second, the presence of emojis might reflect a less negative emotional state among users on the platform.

In this study, the author introduces a new hybrid DL model for analyzing Ashura-Arabic related hate speech and sentiment during the religious event Ashura using DL called CNN-BiGRU-Focus model which tremendously improves the efficiency of both tasks. The model surpassed traditional machine-learning classifiers and deep learning models like DenseNet and InceptionV3. By stacking CNN and BiGRU, this design provided excellent accuracy with the power of local feature extraction from CNN and long-range dependencies capturing property of Bi-Directional GRU over sequential data. Besides, by adding the attention mechanism, model resembled more like a human being who can decide which portion of text should not be focused on while analyzing some other part involved equally in context and predict new word making model interpretable rather oblivion.

In the future, as shown in Table IV, we will expand our model to multi-lingual and multimodal data so that real-time detection of hate content on large-scale social media platforms such as Facebook, Twitter and Instagram can be done. Efforts will also focus on bias mitigation and fairness in predictions to ensure the model is equitable across groups. Thirdly, a federated learning (FL) approach will be used to improve hate speech detection that is privacy-preserving without leaking data.

DATA AVAILABILITY

Data and code used to support the findings of this study have been deposited in the Sarah Data repository (https://github.com/imamu-asa).

CONFLICT OF INTEREST

Author declares that there is no conflict of interest.

REFERENCES

[1] S. Dixon, Countries with the most Twitter users 2022, (Jul. 27, 2022). [Online Video]. Available: https://www.statista.com/statistics/242606/number-of-active-twitter-users-in-selected-countries/

[2] L.-C. Chen, C.-M. Lee, and M.-Y. Chen, "Exploration of social media for sentiment analysis using deep learning," Soft Comput., Oct. 2019, doi: 10.1007/s00500-019-04402-8.

[3] Paul, Jayanta, Ahel Das Chatterjee, Devtanu Misra, Sounak Majumder, Sayak Rana, Malay Gain, Anish De, Siddhartha Mallick, and Jaya Sil. "A survey and comparative study on negative sentiment analysis in social media data." Multimedia Tools and Applications (2024): 1-50.

[4] Abdelsamie, Mahmoud Mohamed, Shahira Shaaban Azab, and Hesham A. Hefny. "A comprehensive review on Arabic offensive language and hate speech detection on social media: methods, challenges and solutions." Social Network Analysis and Mining 14, no. 1 (2024): 1-49.

[5] Alhazmi, Ali, Rohana Mahmud, Norisma Idris, Mohamed Elhag Mohamed Abo, and Christopher Eke. "A systematic literature review of hate speech identification on Arabic Twitter data: research challenges and future directions." PeerJ Computer Science 10 (2024): e1966.

[6] H. Huang, A. A. Zavareh, and M. B. Mustafa, "Sentiment Analysis in E-Commerce Platforms: A Review of Current Techniques and Future Directions," IEEE Access, vol. 11, pp. 90367–90382, 2023, doi: 10.1109/ACCESS.2023.3307308.

[7] Ahmad, Ashraf, Mohammad Azzeh, Eman Alnagi, Qasem Abu Al-Haija, Dana Halabi, Abdullah Aref, and Yousef AbuHour. "Hate speech detection in the Arabic language: corpus design, construction, and evaluation." Frontiers in Artificial Intelligence 7 (2024): 1345445.

[8] Mousa, Aya, Ismail Shahin, Ali Bou Nassif, and Ashraf Elnagar. "Detection of Arabic offensive language in social media using machine learning models." Intelligent Systems with Applications 22 (2024): 200376.

[9] A. A. Wazrah and S. Alhumoud, "Sentiment Analysis Using Stacked Gated Recurrent Unit for Arabic Tweets," IEEE Access, vol. 9, pp. 137176–137187, 2021, doi: 10.1109/ACCESS.2021.3114313.

[10] N. Albadi, M. Kurdi, and S. Mishra, "Are they Our Brothers? Analysis and Detection of Religious Hate Speech in the Arabic Twittersphere," in

2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), Barcelona: IEEE, Aug. 2018, pp. 69–76. doi: 10.1109/ASONAM.2018.8508247.

[11] N. Albadi, “Religious Hate Speech Detection for Arabic Tweets,” https://github.com/nuhaalbadi/Arabic_hatespeech. [Online]. Available: https://github.com/nuhaalbadi/Arabic_hatespeech

[12] A. B. Soliman, K. Eissa, and S. R. El-Beltagy, “AraVec: A set of Arabic Word Embedding Models for use in Arabic NLP,” Procedia Comput. Sci., vol. 117, pp. 256–265, 2017, doi: 10.1016/j.procs.2017.10.117.

[13] R. Alshaalan and H. Al-Khalifa, “Hate Speech Detection in Saudi Twittersphere: A Deep Learning Approach,” in Proceedings of the Fifth Arabic Natural Language Processing Workshop, Barcelona, Spain (Online): Association for Computational Linguistics, Dec. 2020, pp. 12–23. [Online]. Available: https://aclanthology.org/2020.wanlp-1.2

[14] H. Mulki, H. Haddad, C. Bechikh Ali, and H. Alshabani, “L-HSAB: A Levantine Twitter Dataset for Hate Speech and Abusive Language,” in Proceedings of the Third Workshop on Abusive Language Online, Florence, Italy: Association for Computational Linguistics, 2019, pp. 111–118. doi: 10.18653/v1/W19-3512.

[15] I. Aljarah et al., “Intelligent detection of hate speech in Arabic social network: A machine learning approach,” J. Inf. Sci., vol. 47, no. 4, pp. 483–501, Aug. 2021, doi: 10.1177/0165551520917651.

[16] N. Al Twairesh, M. Al Tuwaijri, A. Al Moammar, and S. Al Humoud, “Arabic Spam Detection in Twitter,” presented at the The 2nd Workshop on Arabic Corpora and Processing Tools 2016 Theme: Social Media, May 2016, pp. 38–43.

[17] Yafooz, Wael. "Enhancing Business Intelligence with Hybrid Transformers and Automated Annotation for Arabic Sentiment Analysis." International Journal of Advanced Computer Science & Applications 15, no. 8 (2024).

[18] Charfi, Anis, Mabrouka Besghaier, Raghda Akasheh, Andria Atalla, and Wajdi Zaghouani. "Hate speech detection with ADHAR: a multi-dialectal hate speech corpus in Arabic." Frontiers in Artificial Intelligence 7 (2024): 1391472.

[19] Ibrahim, Yasmine M., Reem Essameldin, and Saad M. Saad. "Social Media Forensics: An Adaptive Cyberbullying-Related Hate Speech Detection Approach Based on Neural Networks With Uncertainty." IEEE Access (2024).

[20] Louati, Ali, Hassen Louati, Abdullah Albanyan, Rahma Lahyani, Elham Kariri, and Abdulrahman Alabduljabbar. "Harnessing Machine Learning to Unveil Emotional Responses to Hateful Content on Social Media." Computers 13, no. 5 (2024): 114.

[21] Aljohani, Abeer, Nawaf Alharbe, Rabia Emhamed Al Mamlook, and Mashael M. Khayyat. "A hybrid combination of CNN Attention with optimized random forest with grey wolf optimizer to discriminate between Arabic hateful, abusive tweets." Journal of King Saud University-Computer and Information Sciences 36, no. 2 (2024): 101961.

[22] Daraghmi, Eman Yaser, Sajida Qadan, Yousef Daraghmi, Rami Yussuf, Omar Cheikhrouhou, and Mohammed Baz. From Text to Insight: An Integrated CNN-BiLSTM-GRU Model for Arabic Cyberbullying Detection. IEEE Access (2024).

[23] Almaliki, M., Almars, A.M., Gad, I. and Atlam, E.S., 2023. Abmm: Arabic bert-mini model for hate-speech detection on social media. Electronics, 12(4), p.1048.

[24] Khezzar, Ramzi, Abdelrahman Moursi, and Zaher Al Aghbari. "arHateDetector: detection of hate speech from standard and dialectal Arabic Tweets." Discover Internet of Things 3, no. 1 (2023): 1.

[25] Mohamed, Mohamed S., Hossam Elzayady, Khaled M. Badran, and Gouda I. Salama. "An efficient approach for data-imbalanced hate speech detection in Arabic social media." Journal of Intelligent & Fuzzy Systems 45, no. 4 (2023): 6381-6390.

[26] Mubarak, Hamdy, Sabit Hassan, and Shammur Absar Chowdhury. "Emojis as anchors to detect arabic offensive language and hate speech." Natural Language Engineering 29, no. 6 (2023): 1436-1457.

[27] Perera, Suresha, Nadeera Meedin, Maneesha Caldera, Indika Perera, and Supunmali Ahangama. "A comparative study of the characteristics of hate speech propagators and their behaviours over Twitter social media platform." Heliyon 9, no. 8 (2023).

[28] Priyadarshini, Ishaani, Sandipan Sahu, and Raghvendra Kumar. "A transfer learning approach for detecting offensive and hate speech on social media platforms." Multimedia Tools and Applications 82, no. 18 (2023): 27473-27499.

[29] Saleh, Hind, Areej Alhothali, and Kawthar Moria. "Detection of hate speech using bert and hate speech word embedding with deep model." Applied Artificial Intelligence 37, no. 1 (2023): 2166719.

[30] Alqarni, Arwa, and Atta Rahman. "Arabic tweets-based sentiment analysis to investigate the impact of COVID-19 in KSA: a deep learning approach." Big Data and Cognitive Computing 7, no. 1 (2023): 16.

[31] Al-Jarrah, Heba, Mohammad Al-Smadi, Mahmoud Hammad, and Fatima Shannaq. "Using Deep Learning Techniques to Detect Hate and Abusive Language in Arabic Tweets." International Journal of Intelligent Engineering & Systems 17, no. 5 (2024).

[32] Abdelhakim, Mohamed, Bingquan Liu, and Chengjie Sun. "Ar-PuFi: A short-text dataset to identify the offensive messages towards public figures in the Arabian community." Expert Systems with Applications 233 (2023): 120888.

[33] Halawani, Hanan T., Aisha M. Mashraqi, Souha K. Badr, and Salem Alkhalaf. "Automated sentiment analysis in social media using Harris Hawks optimisation and deep learning techniques." Alexandria Engineering Journal 80 (2023): 433-443.

[34] Berrimi, Mohamed, Mourad Oussalah, Abdelouahab Moussaoui, and Mohamed Saidi. "Attention mechanism architecture for arabic sentiment analysis." ACM Transactions on Asian and Low-Resource Language Information Processing 22, no. 4 (2023): 1-26.

[35] Kaddoura, Sanaa, Suja A. Alex, Maher Itani, Safaa Henno, Asma AlNashash, and D. Jude Hemanth. "Arabic spam tweets classification using deep learning." Neural Computing and Applications 35, no. 23 (2023): 17233-17246.

[36] Abdulsalam, Asma, Areej Alhothali, and Saleh Al-Ghamdi. "Detecting suicidality in Arabic Tweets using machine learning and deep learning techniques." Arabian Journal for Science and Engineering (2024): 1-14.

[37] Jaleel, Abdul, Mehmoon Anwar, Farooq Ali, Raza Mukhtar, and Muhammad Farooq. "Islamophobia Content Detection Using Natural Language Processing." Journal of Computing & Biomedical Informatics 4, no. 02 (2023): 88-97.

[38] Boulouard, Zakaria, Mariya Ouaissa, Mariyam Ouaissa, Moez Krichen, Mutiq Almutiq, and Karim Gasmi. "Detecting hateful and offensive speech in Arabic social media using transfer learning." Applied Sciences 12, no. 24 (2022): 12823.