

HOSTED BY



ELSEVIER

Contents lists available at ScienceDirect

# Journal of King Saud University - Computer and Information Sciences

journal homepage: [www.sciencedirect.com](http://www.sciencedirect.com)

Full length article

## A hybrid combination of CNN Attention with optimized random forest with grey wolf optimizer to discriminate between Arabic hateful, abusive tweets

Abeer Aljohani <sup>a,\*</sup>, Nawaf Alharbe <sup>a</sup>, Rabia Emhamed Al Mamlook <sup>b,c</sup>, Mashael M. Khayyat <sup>d</sup><sup>a</sup> Department of Computer Science, Applied College, Taibah University, Medina 42353, Saudi Arabia<sup>b</sup> Department of Business administration Trine University Indiana, United States of America<sup>c</sup> Department of Mechanical and Industrial Engineering University Zawia Tripoli, Libya<sup>d</sup> Department of Information Systems and Technology Faculty of Computer Science and Engineering, University of Jeddah, 23442, Jeddah, Saudi Arabia

### ARTICLE INFO

#### Keywords:

Arabic natural language processing  
Deep learning  
Machine learning  
Grey wolf optimizer

### ABSTRACT

Arabic hateful speech recognition has long been a major area of focus in Natural Language Processing (NLP) research. In light of recent advancements in transformer models and deep learning, researchers are now turning to transfer learning techniques based on existing models such as BERT for Arabic hateful speech recognition. To detect Arabic hateful contexts, using advanced machine learning algorithms and NLP techniques is essential. These techniques can help to detect different forms of hateful contexts in Arabic by analyzing the text for lexical, semantic, and syntactic features. In this research, we proposed a new hybrid approach that combines deep and machine learning models to detect hateful and abusive content in Arabic. The proposed model consists of a combination of convolutional neural networks and attention layers that are trained to differentiate between normal, abusive, and hateful contexts in Arabic. In the first step, we used a pre-trained model to extract features from the hateful Arabic context. After that, we used an optimized random forest combined with particle swarm optimization and grey wolf optimizer to classify the extracted features. Finally, we evaluated the performance of the model to detect hateful Arabic contexts. To evaluate the proposed method we used 5846 and 6023 tweets with 3 categories of hateful, abusive, and normal Arabic contexts. The experimental result indicates 97.16% accuracy, 97.15% F1-score, 97.17% precision, and 97.13% sensitivity using CNN Attention + optimized random forest by the grey wolf optimizer on 5846 tweets. 97.83% accuracy, 97.83% F1-score, 97.84% precision, and 97.83% sensitivity have been reported CNN Attention + optimized random forest by the grey wolf optimizer on 6023 tweets.

### 1. Introduction

It is easy to access social media. Platforms such as Twitter and Facebook, allow people to share their opinions on politics, economics, sports events, etc. Unfortunately, this freedom to express oneself can also lead to the spread of racism, offensive language, and other forms of verbal abuse, which are perpetuated by differences in culture (Shin and Choi, 2021). Abusing from this platform can happen in various languages and Arabic language is one of the languages that is used for spreading offensive context as well (Litvak et al., 2022). A process of regulation on social media platforms can be used to detect and

prevent offensive attacks against people and organizations. Due to its intricate morphology, the Arabic language makes it difficult to identify harmful contents. There are about 10,000 roots and more than 900-word patterns that form the foundation of nouns and verbs, making the task of detecting hateful phrases a challenging one (Ofer et al., 2021).

Algorithms based on Artificial Intelligence (AI) are key in recognizing and preventing hostile actions from occurring (Gubatan et al., 2021). Natural Language Processing (NLP) is an area of AI devoted to the development of systems that can interpret and process natural language spoken or written by humans. It focuses on the analysis of

\* Corresponding author.

E-mail addresses: [aahjohani@taibahu.edu.sa](mailto:aahjohani@taibahu.edu.sa) (A. Aljohani), [nrharbe@taibahu.edu.sa](mailto:nrharbe@taibahu.edu.sa) (N. Alharbe), [ralmamlook@zu.edu.ly](mailto:ralmamlook@zu.edu.ly), [almamlookr@trine.edu](mailto:almamlookr@trine.edu) (R.E. Al Mamlook), [mkhayyat@uj.edu.sa](mailto:mkhayyat@uj.edu.sa) (M.M. Khayyat).

Peer review under responsibility of King Saud University.



Production and hosting by Elsevier

<https://doi.org/10.1016/j.jksuci.2024.101961>

Received 8 June 2023; Received in revised form 18 December 2023; Accepted 30 January 2024

Available online 12 February 2024

1319-1578/© 2024 The Author(s). Published by Elsevier B.V. on behalf of King Saud University. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

human language, both linguistically and computationally, to enable computers to understand and respond to human language (Chowdhary, 2020). Using AI to detect hate speech in the Arabic language is a challenging task due to the difference between common and standard understanding of what is defined as hate speech. Due to the lack of annotated datasets for hate speech in Arabic, Machine Learning (ML) methods for detecting hate speech in this language are more difficult than in English (Tavakolian et al., 2022b; Al-Hassan and Al-Dossari, 2021). The precise definition of what is considered offensive in the Arabic context varies between European and Middle Eastern countries, creating difficulty in accurately labeling these types of contexts (Wachs et al., 2021; Aldjanabi et al., 2021). Deep Learning (DL) is an advanced form of ML algorithm that has seen increased use in the area of detecting hate speech (Aldjanabi et al., 2021; Al-Hassan and Al-Dossari, 2021). However, developing a proper DL model for Arabic hateful context detection is difficult due to the lack of a properly labeled dataset (Khalafat et al., 2021). Thus, extracting the best sets of features from the Arabic context will consume time for marginal improvements.

Recently, this issue was addressed with the help of innovative DL models that include either an attention layer or a transformer-based model (Aldjanabi et al., 2021). However, these models could not discriminate between misleading abusive, and offensive information. Thus, the problem of discriminating between similar contexts with different labels stands still. The imbalance distribution of each label plays a key role in poor discrimination performance (Tavakolian et al., 2022b).

To get a better understanding of the mentioned issues, similar studies are examined in the related work section. Possible solutions to each issue are investigated, and a suitable solution is proposed to enhance the efficiency of the existing method of detecting offensive contexts with only a small amount of data. In this research, first a deep learning structure with a combination of various attention layers is proposed to extract information from the preprocessed text. Then a tuned ML with grey wolf optimizer is proposed to improve the distinguishable ability of the proposed method between the offensive and abusive contexts in the Arabic language. To refer to the lack of a proper dataset for Arabic hateful detection a sophisticated preprocessing and augmentation process is proposed. Unlike similar research, the proposed method focuses on increasing both true positive and negative rates. Thus, it can detect Arabic offensive contexts without abnormal false alarms. Objectives of the proposed method are:

- Presenting new hybrid methods for Arabic context detection using fewer tweets.
- The proposed model outperformed similar approaches in the related work for Arabic hateful context detection.
- Optimizing the final classifier using a customized objective function.

This paper is split up into 6 parts. Section 2 looks at the history of Arabic offensive context detection using ML and DL. Section 3 provides information about the dataset and associated statistical information. Section 4 details the hardware arrangement employed in both training and evaluation of the suggested technique. Section 5 outlines the designed hybrid model as well as the method for extracting features with the incorporation of Convolutional Neural Networks (CNN) and attention layers. In Section 5, the proposed model's accuracy and validity are assessed. Section 6 details the experimental findings and demonstrates the model's superiority over other ML models for distinguishing between Arabic contexts and abuse contexts. Lastly, Section 7 summarizes the research's contributions, shortcomings, and future potential.

## 2. Related work

In this section, a detailed explanation of similar research for discriminating between offensive and normal Arabic contexts is presented.

Spreading false or hateful messages in Arabic can be a source of promoting terrorism and other violent acts in real life. The consequences of such propaganda can be devastating for the entire Arabic-speaking population.

Albadi et al. (2019) collected annotated Arabic tweets for religious hatred detection. They used the n-gram (Georgieva-Trifonova and Duraku, 2021) method to encode the Arabic language. They used a combination of pre-trained word embedding with GRU (Liu et al., 2021) as a classifier. Albadi et al. evaluated the proposed method using unseen Arabic tweets and reported a 77% F1 score, 76% precision, and 84% sensitivity.

Aldjanabi et al. (2021) proposed a combination of Arabic Bidirectional Encoder Representation from Transformer (BERT) (Ghaddar et al., 2021) and attention transformers model (Chefer et al., 2021) for Arabic hatred detection. The proposed method for detecting hatred and offensive language in Arabic was evaluated with data consisting of 31870 samples. The accuracy achieved was reported to be 87.46% and the F1 score was 87.18%. Aldjanab et al. used different sources of hate and abusive content in the Arabic language to test their model.

Al-Hassan and Al-Dossari (2021) proposed a system that combined CNN with Long Short Term Memory (LSTM) (Zha et al., 2022) and Gated Recurrent Unit (GRU) for hatred detection in Arabic tweets. The proposed method achieved an accuracy of 72%, a precision of 75.1%, sensitivity of 73%, and an F1 score of 73% when evaluated using 11,000 labeled Arabic tweets..

Khalafat et al. (2021) used already available Machine Learning models such as Support Vector Machine (SVM) (Otchere et al., 2021), Naive Bayesian (NB) (Chen et al., 2020), and K Nearest Neighbor (KNN) (Cunningham and Delany, 2021) for hatred detection in Arabic language. In this research Khalafat et al. encoded the Arabic texts via n-gram encoding and tested their proposed method on the Arabic lexicon dataset. Upon evaluation, their results indicated that the SVM method surpassed prior similar research attaining an F1 score of 72%, sensitivity of 75%, precision of 69%, and accuracy of 68%.

Husain (2020) Compare the performance of ensemble machine learning models like RF with traditional models like SVM to detect Arabic-language hate speech. In this research, 7835 tweets were used as the dataset. The textual information was converted to numerical values via the Term Frequency-Inverse Document Frequency (TF-IDF) technique (Gomes et al., 2023). Husain et al. reported that RF has outperformed traditional machine-learning models for Arabic hatred detection with an 88% F1 score and 89% accuracy.

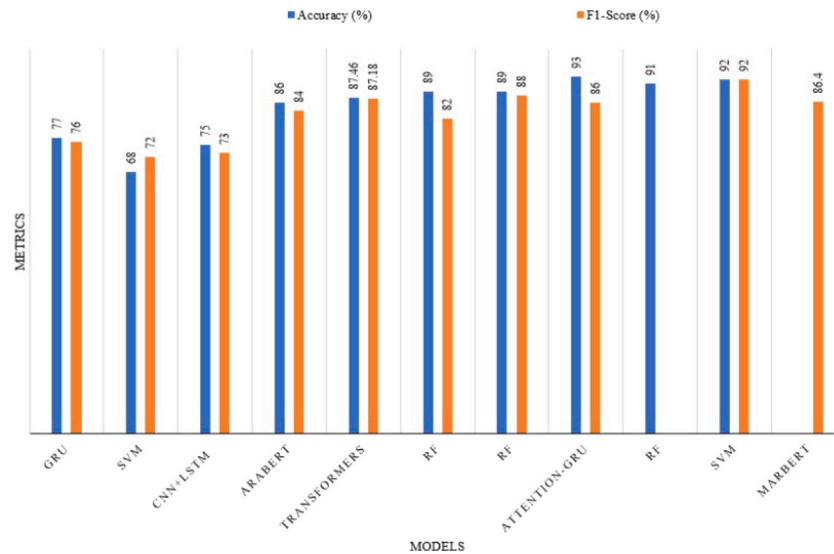
In another article by (Husain and Uzuner, 2021) used the AraBERT (Antoun et al., 2020) for extracting information from the Arabic context and they used transfer learning for the final classification. Husain et al. evaluated the proposed model using the Levantine Hate Speech and Abusive(L-HSAB) dataset. Finally they reported 86% precision 84% recall, and 84% F1-score for discriminating between abusive, normal, and offensive contexts.

Haddad et al. (2020) combined attention with GRU layers to discriminate between offensive and hateful Arabic contexts. They have composed the proposed method using two GRUs and attention to one dense layer at the end for classification. The author evaluated the proposed method using 28k comments from YouTube. In this research Arabic version of the Word2Vec model is used to convert the contextual information to a numerical value. Haddad et al. reported 93% accuracy, 83% recall, and 86% F1-score for discriminating between offensive and normal context.

Mursi et al. (2022) studied the detection of hate speech in the Arabic language by analyzing 100,000 tweets collected from 2014 to 2020. A Multi-Layer Perceptron (MLP) with three hidden layers was developed and compared to an SVM to detect hatred. Word2Vec was employed to encode the Arabic context (Jang et al., 2019). They reported 92% accuracy, 95% sensitivity, and 92% F1 score for hatred context detection using SVM.

**Table 1**  
Recent published research for hatred detection.

Author	Dataset	Preprocessing	Model	Accuracy (%)	F1-Score (%)
Albadi et al(2019)	6.6 K of religious hatred tweets	n-gram	GRU	77	76
Aldjanab et al(2021)	L-SHAB	AraBERT	Transformers Model	87.46	87.18
Al-Hassan et al(2021)	11K tweets	n-gram	CNN+LSTM	75	73
Haddad et al (2020)	28K Youtube comments	Arabic Word2Vec	Attention-GRU	93	86
khalafat et al(2021)	L-SHAB	n-gram	SVM	68	72
Husain et al (2021)	L-SHAB	TF-IDF	AraBERT	86	84
Husain et al (2020)	7.8 K tweets	TF-IDF	RF	89	88
Mursi et al(2022)	100 K hatred tweets	Word2Vec	SVM	92	92
Messaoudi et al (2020)	T-HSAB	TF-IDF	RF	89	82
Alhejaili et al(2022)	T-HSAB	TF-IDF	RF	91	–



**Fig. 1.** Comparison between reviewed research and their reported results.

Messaoudi et al. (2020) combined the BERT with the LSTM layer for extracting features from the Arabic context and creating the final classifier. The proposed model is composed of n-grams to convert the Arabic text into a numerical value. L-HSAB and Tunisian-Hate Speech and Abusive (T-HSAB) were used as the datasets in this research. Messaoudi et al. reported 89% accuracy, and 82% F1-score using the proposed method on the combined L-HSAB and T-HSAB.

Alhejaili et al. (2022) collected new sets of tweets during COVID-19 to train the ML model for discriminating between hateful and normal contexts. In this research, to convert the textual information into numerical values TF-IDF was used. They used various ML models such as SVM, RF, and KNN as the final classifier. They reported 90.8% accuracy using the RF model for discriminating between hateful and offensive contexts.

Based on reviewed articles, most of the researchers have used a version of BERT to extract features from the original Arabic context. AraBERT and salamBERT are the tuned Arabic content models that were used to preprocess and extract time series dependencies relation among the contexts. Based on the reviewed article ML methods especially SVM and RF are the main choices if the number of samples in the dataset is small. Thus, as time passes researchers are exploiting the recent development and novel architectures of DL for hatred detection. A summary of reviewed research is shown in Table 1.

In Table 1 various research with different datasets and corpora have been reviewed. Based on Table 1 the best-achieved results on L-SHAB is 86% and T-HSAB is 91% accuracy using AraBERT and RF consequently. Other researchers have used various datasets with different corpora and sizes and the best results belong to Mursi et al. (2022). that worked on 100 K hatred tweets with SVM. Based on the reviewed articles RF and SVM were the two best models for detecting hatred speech in various Arabic dialects.

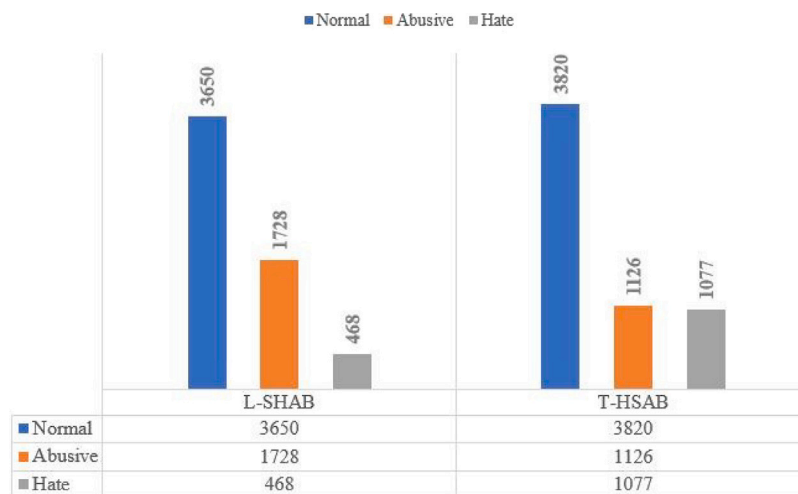
As shown in Fig. 1 SVM was reported as the best ML algorithm for detecting the hatred context in 100,000 tweets. However, the best-reported accuracy was reported using the Attention-GRU model for detecting the hatred context in 28,000 YouTube comments. Also, the reported results by the reviewed researchers indicate more use cases of DL models over ML models for feature extraction from texts. However, the average reported accuracy for hatred context detection using the ML method is slightly higher than ML methods.

### 3. Method

In this section, we go over the steps taken to prepare the dataset and apply the proposed method for extracting features and using them for classification. This includes parsing, removing emojis, punctuation, normalizing, and embedding Arabic contexts. From the embedded context, we then extract the suitable feature set. Finally, we discuss the optimization of the classification model.

#### 3.1. Dataset

In this research we have used two sets of different datasets with the name L-SHAB (Mulki et al., 2019), and T-HSAB (Haddad et al., 2019). Both of these datasets contain information in 3 classes of “Normal”, “Abusive”, and “hateful” contexts. L-SHAB is composed of 5846 tweets with 102 duplicated tweets. T-HSAB is composed of 6023 tweets with 31 duplicated samples. Distribution for the classes in each dataset is shown in Fig. 2. As shown in Fig. 2, the imbalance distribution among normal and hateful contexts is clear in both datasets. The ratio between “Normal” and “Abusive” Contexts are 7.80 in L-SHAB and 3.55 in T-HSAB. The imbalance distribution between the normal and offensive



**Fig. 2.** Distributions of each class in the L-SHAB and T-HSAB.


L-SHAB		T-HSAB	
Hate	Distribution (%)	Hate	Distribution (%)
كلب (Dog)	1	تونسي (Tunisia)	2.01
كلاب (Dogs)	0.98	شعب (People)	0.74
لبناني (Lebanese)	0.55	الاسلام (Islam)	0.43
قطر (Qatar)	0.55	اليهود (Jews)	0.39
سوري (Syrian)	0.39	التونسي (Tunisian)	0.26
Abusive	Distribution (%)	Abusive	Distribution (%)
هوا (Sh*t)	1.58	تونسي (Tunisia)	1.39
كول (Swallow)	1.52	ع (Fu*k You)	0.40
كلب (Dog)	0.97	ميوهن (Faggot)	0.32
حمار (Donkey)	0.59	لعنه (Curse)	0.31
خراس (Chimp)	0.59	كلب (Dog)	0.30

Fig. 3. Distribution on top 5 repeated abusive and hateful contexts in L-SHAB and T-HSAB.

classes will affect the classification performance (Tavakolian et al., 2022a). L-SHAB is collected from the Syrian and Lebanese. T-HSAB is collected from Tunisian nationality. Both datasets are screened to contain certain offensive words in each comment. The Distribution of each word in both L-SHAB and T-HSAB is shown in Fig. 3. As shown in Fig. 3, Most of the offensive contexts are related to nationality and religion. Thus, extracting specific keywords with higher distribution will help the classifier understand the Abusive and hateful contexts better.

### 3.2. Preprocessing

To extract meaningful information from the raw context, a series of steps are conducted. This includes removing new lines and combining all sentences into one text. Then we removed the links related to signs such as "(?=@-https?:/)\$+>". Usually, tweets contain a series of ASCII characters and emojis. Thus, in the next step, we remove these characters from the tweets. For the next step, we will take out any punctuation from the tweets and eliminate any stop words in Arabic that do not provide any additional information in the text. Additionally, we will convert any different formatting of the language to normal Arabic alphabets so that there is consistency with the text. We employed MARBERT to segment the input sentence to capture each word and its constituent components. As observed in related studies, utilizing MARBERT enhances the classifier's performance by improving its comprehension. MARBERT is an extensive revision over the AraBERT which was trained using the 128 GB of text that is composed of both Arabic

dialectics and Modern Standard Arabic (MSA) ([Abdul-Mageed et al., 2020](#)). Opting for MARBERT in place of AraBERT results in distinct tokenization methods for MSA and other Arabic dialects. This separation between MSA and other dialects proves advantageous, particularly considering the inclusion of diverse Arabic dialects within both the T-HSAB and L-SHAB datasets. Since in the Arabic language, there are different appearances of the same letter, we replace this variety with only one shape for each letter. For example the letter “ل” has various shapes like “Beattie and Esmonde-White, 2021). PCA is a technique used to identify the directions (axes) that capture the most variability within a dataset. It does this by finding the directions of maximum variance in the data, and then projecting the data onto the new axes. PCA can also be used to identify the axis that captures the most amount of variance in the training set, as well as a second axis orthogonal to the first one that accounts for the largest amount of the remaining variance in the dataset. To ensure that we are using 99% of useful information in this work we set a criterion of 99% for decreasing the dimension of the input dataset. The result of using PCA is reducing the feature set

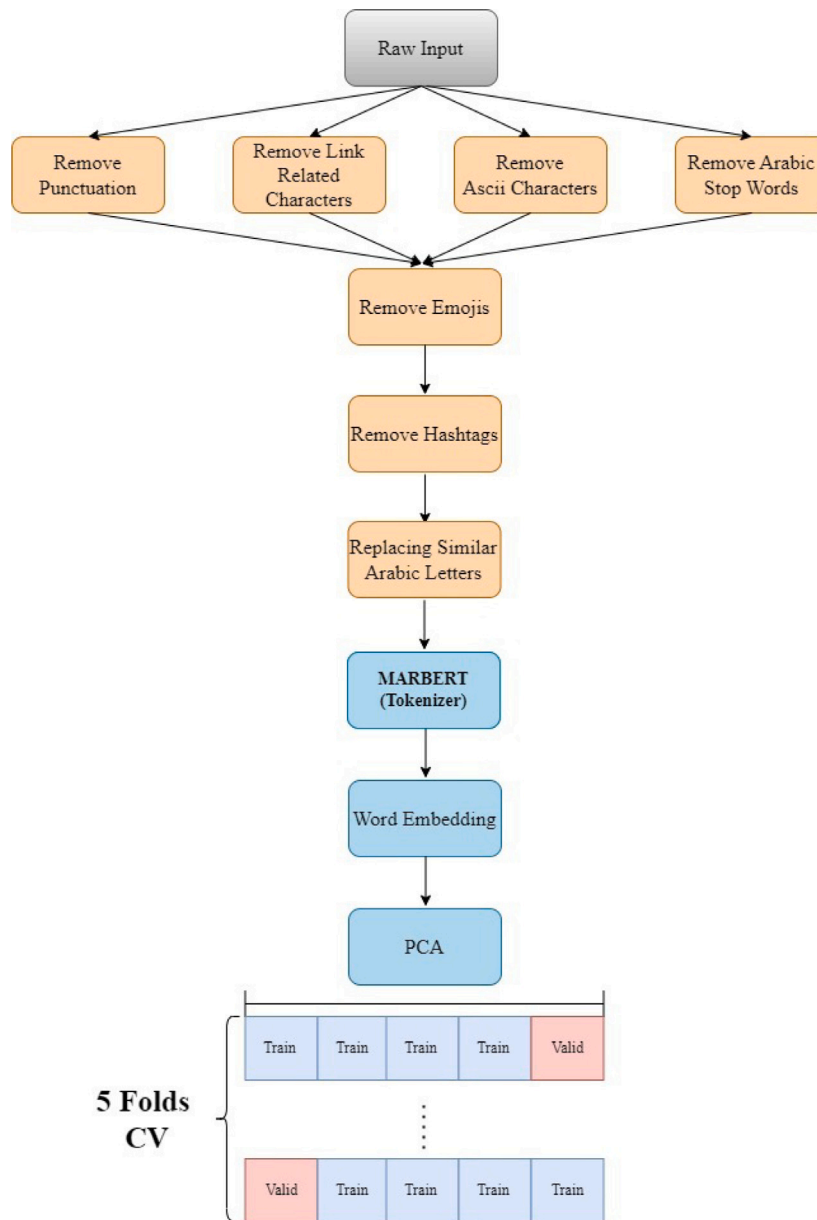


Fig. 4. Architectures of preprocessing and feature extraction.

from 2000 to 1727 features for the L-SHAB dataset. The PCA reduced the feature set from 2000 to 1672 features in the T-HSAB dataset by gathering the proper information, we divide it into train and test sets. We separate the dataset into 80% for training and 20% for testing. 5 folds cross-validation is used to validate the final result. A summary of all processes for preprocessing and feature extraction is shown in Fig. 4.

### 3.2.1. Data augmentation

The primary challenge of an imbalanced dataset is that it has an uneven number of samples in each class. One possible approach to address this challenge is to use a sampling technique to create a pseudo-balanced dataset. The deep learning community has introduced several sampling methods to augment small-sized datasets for classification. Synthetic Minority Over-sampling Technique (SMOTE) (Mansourifar and Shi, 2020) and Modified Synthetic Minority Over-sampling Technique (M-SMOTE) (Nedjar et al., 2022) are some samples of the sampling augmentation methods. We combined the output of under-sampling with the result of the MSMOTE technique which creates additional samples in the minority class, helping to balance the dataset.

The under-sampling removes instances randomly from the majority class to achieve an even class distribution, while MSMOTE creates new samples from the minority class to increase its density. The combination of these two methods produces a dataset with balanced class proportions. MSMOTE is an adapted version of SMOTE that splits the minority class into three groups: safe, border, and latent noise samples. It determines the distinctions between these groups by gauging the distances between all the samples. Using a combination of sampling techniques raises the sample size of the underrepresented classes, which in turn minimizes the disproportionate influence of the predominant classes and generates a moderately balanced data set (Nedjar et al., 2022).

### 3.3. Proposed model

Transfer learning with AraBERT is becoming a popular approach for detecting hateful speech in recently published articles (Husain and Uzuner, 2021). Transfer learning is a popular approach used to train deep learning models, and many researchers have incorporated it into



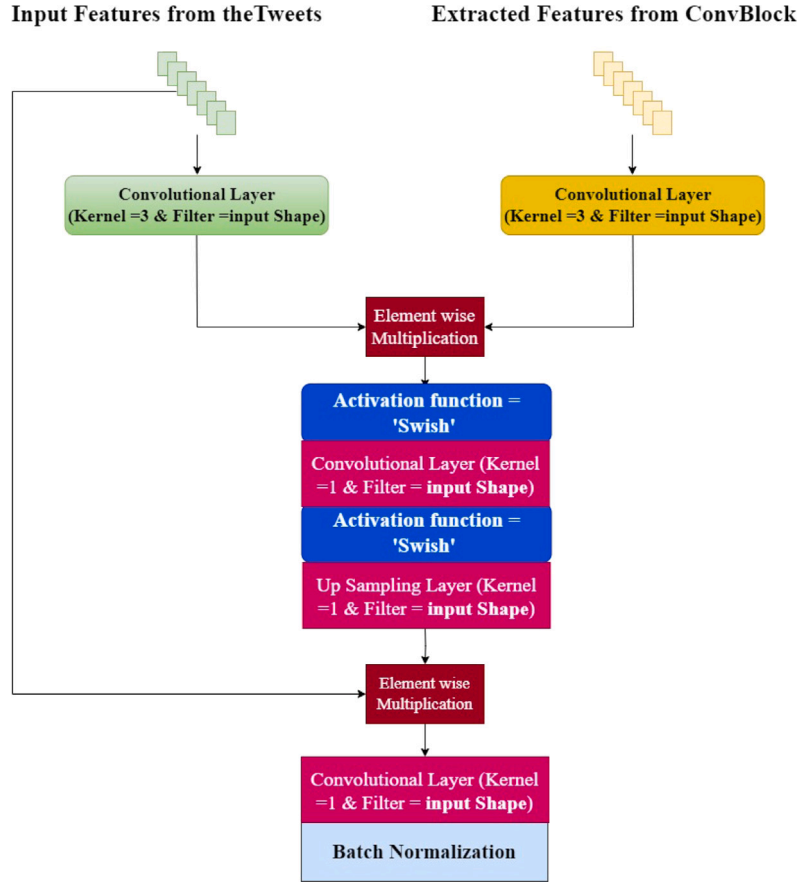


Fig. 5. Architectures of preprocessing attention layer.

their work (Aldjanabi et al., 2021). AI is also evolving rapidly, and transformers and multi-head attention layers have recently become popular for this type of task (Bhojanapalli et al., 2020). Motivated by reviewed articles, we exploited the architecture of artificial networks like convolutional and attention layers (Povey et al., 2018). In this research, we propose a combination of convolutional and attention layers to extract proper features from a long sequence of tweets.

### 3.4. Convolutional layer

Each neuron in the convolutional layer is connected to a subset of the input values, with the same set of weights applied across the entire sequence. This allows the layer to detect consistent patterns in the data, such as edges or corners of images. As CNNs become more intricate and have more layers, the model can understand complex correlations between nearby words (Farag, 2023). The top CNN layers are connected only to neurons in the layer before which are located within a small area. There are various types of convolutional layers that can extract information based on the construction of the input dataset. The one-dimensional convolution layer (Conv1D) can extract tile relation dependencies from textual information. Since the input information is sparse and most of the tweets are short texts, we utilized a shallow convolutional network with a large kernel. To ensure proper circulation of forward signals in the proposed model, we place a batch normalization layer between the convolutional layers (Visca et al., 2022). Instead of using a pooling layer to decrease the length of extracted information, we used the Conv1D with a stride of 2 and kernel size of 4 (Visca et al., 2022).

The mathematical equation behind the proposed model is explained as follows. Given the sets of input feature set as  $\{I_1, I_2, \dots, I_n\}$  the

convolutional layer extracts the feature map using random sets of weight  $\{W_1, W_2, \dots, W_n\}$ .

$$Feature\ Map = \sum_{f=0}^{f-1} \sum_{K=0}^{k-1} W_{f,k} \cdot I_{i,j} \quad (1)$$

the  $k$  is the size of the receptive field, and  $f$  is the number of feature maps in the previous layer. The output of the convolutional ( $O_c$ ) layer enters the batch normalization. By using the convolutional block, the proper feature sets are extracted from the textual sequence. In this research, we use the Swish activation function. Swish is an activation function proposed by Google in 2017 that is a continuous and non-monotonic non-linearity for neural networks (Allu and Padmanabhuni, 2022).

### 3.5. Attention layer

As the depth of the proposed CNN increases, the size of the extracted features decreases while the number of feature maps increases. To ensure important extracted information is not forgotten, the attention mechanism can be utilized to remember and amplify crucial data (Niu et al., 2021).

The attention layer takes the output of two successive convolutional blocks and applies a transformation to it, creating a contextual vector. This vector then determines how much importance to assign to each feature extracted from the blocks to generate the final output. The process of calculating the context vector and importance vector of the attention layer is shown in Fig. 5.

This final feature representation is fed to an attention layer that chooses which features are highly correlated for final classification. As shown in Fig. 5, to combine the extracted features and calculate the context, we use Conv1D. To generate the hidden context representation,

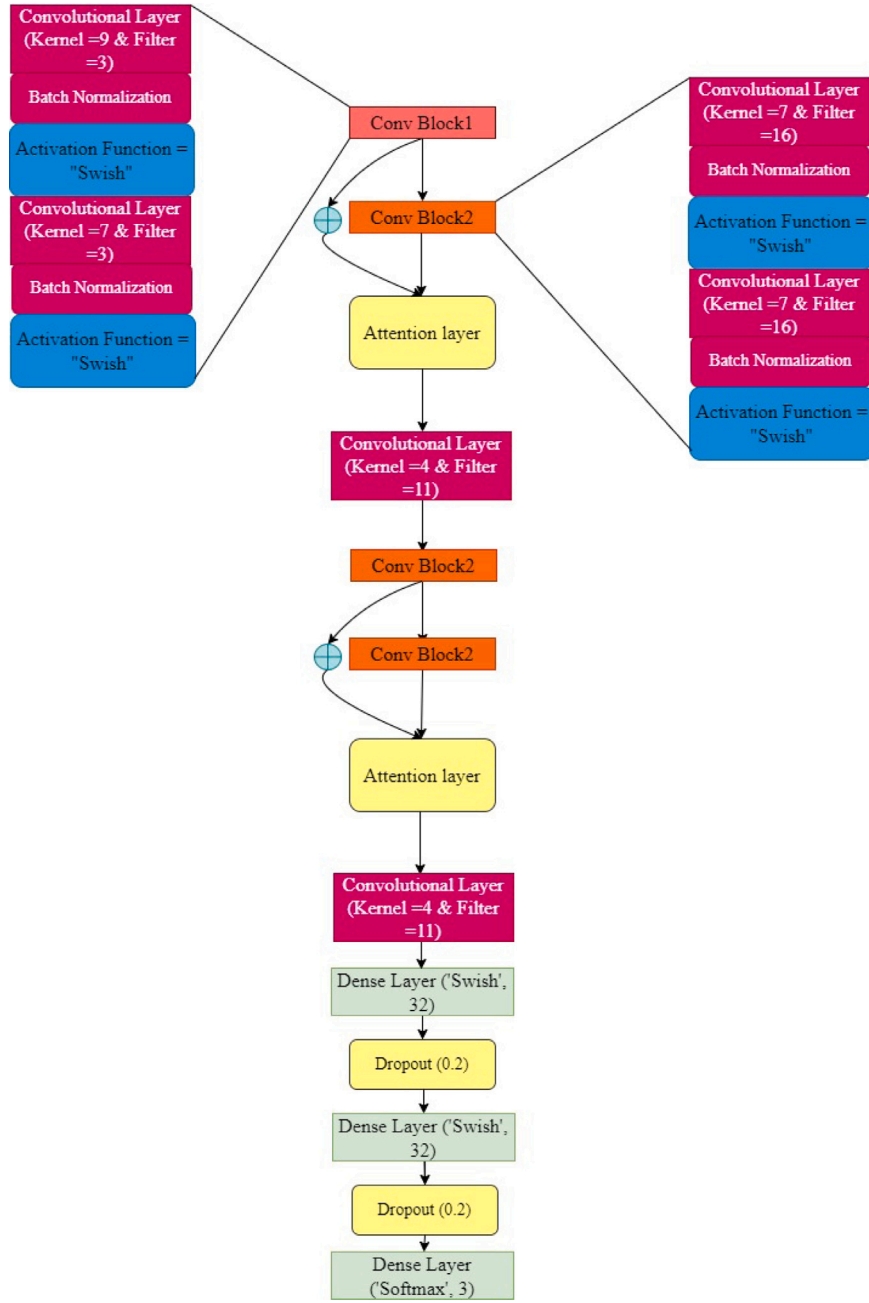


Fig. 6. Architectures of the whole proposed model.

we multiplied the extracted features together point-wise. Afterward, to obtain the attention coefficient, we up-sampled the hidden context and then multiplied the results point-wise to get the attention output. This enabled the model to focus on the most important words in the tweets. By this phase, the most important features are extracted using the proposed method. To decrease the size of extracted features even more we have utilized dense layers to specify the most important features for the final classifier. The number of neurons in the first and second dense layers is 32 and 32 respectively. At the last layer, we used 3 neurons which is equal to the number of classes. The structure of the proposed model is shown in Fig. 6. As shown in Fig. 6, the proposed method is composed of 4 main components with the name of Conv Block1, Conv Block2, attention, and final discriminator. To not lose the flow of the model, batch normalization at the end of Conv Block1 and Conv Block2. Also, Dropout layers have been placed among the dense layers to train the best weight for the final discriminator without overfitting (Santos

and Papa, 2022). The initial kernel size of Conv Block1 is 9 to capture the relation between each word over a large window. In Conv Block2, the filter size has increased from 3 to 16 to extract various features map from Conv Block1.

### 3.6. Optimized classifier

Previously, we described the process of using the combination of convolution and attention layers to extract features from the Arabic contexts. To conduct the process of feature extraction properly, first, we trained the model using an already available dataset. Thus, initially, we train the proposed method to detect hateful contexts. Then, we extract features from the last dense layer before classification. Thus, the number of extracted features using the proposed method is 32.

As mentioned in the previous step, the proposed method uses the extracted features for final classification. However, to use this extracted

feature set properly, the proposed method uses a heuristic optimization algorithm. The main role of the heuristic optimizer in the proposed method is to choose the best feature set for final classification. In this research, we used Particle Swarm Optimizer (PSO) (Tavakolian et al., 2022a), and Grey Wolf Optimizer (GWO) (Safaldin et al., 2021), to optimize the last ML classifier.

### 3.6.1. PSO

The PSO algorithm guides an ensemble of particles with different positions in a search space to move toward a global optimum while still maintaining local diversity. Each particle is associated with a velocity that reflects its exploration capability, and the ensemble of particles makes up a swarm. The particles try to reach the global optimum by gradually updating their velocities and positions, and this process is repeated until the global optimum is found. PSO uses principles of bird flocking to identify the most advantageous point in a cost function. This is done by breaking the cost function down into a set of features, which are then used to locate the optimal solution. PSO uses binary decisions to select individual features in the original feature set to create multiple subsets. These individual features, or particles, act together in the same environment and operate to optimize the objective function that relates to the performance of each subset. The ultimate decision is based on the values of the objective function. The most suitable particle (sub-feature group) with the lowest objective function value (highest performance) is chosen as the most ideal feature set for pothole detection. PSO convergence rate is faster compared to other feature selection approaches (Chtita et al., 2022). The formula for updating each particle location is mentioned as follows:

$$x_{id}^{t+1} = x_{id}^t + v_{id}^{t+1} \quad (2)$$

Where the  $v_{id}^{t+1}$  is the velocity of each particle in the search space. The formula for the calculation of each particle velocity is shown in Eq. (3).

$$v_{id}^{t+1} = W * v_{id}^t + C_1 * R_{1i} * (p_{id} - x_{id}^t) + C_2 * R_{2i} * (p_{gd} - x_{id}^t) \quad (3)$$

In this formula,  $W$  refers to inertia weight,  $C_1$  and  $C_2$  are refer to acceleration constants.  $R_{1i}$  and  $R_{2i}$  are designated based on random values between the range of 0 and 1.  $p_{id}$  and  $p_{gd}$  represent the elements of  $p_{best}$  and  $g_{best}$  in the  $d$ th dimension. The initial population of the PSO is determined randomly. The model which is used for classification is random forest (Priyadarshi et al., 2019). In this research, we focus on increasing both true positive and negative predictions. Thus, we define the objective function as follows:

$$Cost_{Function} = (1 - (0.5 * precision + 0.5 * specificity)) + (1 - \alpha) * \left(1 - \frac{Selected\ Sub - feature}{Length\ of\ feature\ set}\right) \quad (4)$$

Where  $\alpha$  is the hyperparameter value to indicate the important number of selected features over the true prediction rate. In this research, we set the  $\alpha$  to 0.5. As shown in Eq. (3), the proposed method tries to decrease the cost function. To decrease the cost function, the proposed method should increase True positive and True negative predictions. The optimization algorithm is presented below.

The proposed structure of Algorithm 1 is a revision over the original PSO algorithms via the customized cost function. Utilizing algorithm 1, PSO aims to find the best candidate (model) by optimizing the cost function over a defined number of iterations. The speed of optimization relies on the initial position of the agent concerning the random numbers  $C_1$  and  $C_2$ . The maximum number of iterations for PSO is set to 100. The number of particles is set to 30.

### 3.6.2. GWO

GWO mimics the wolf pack hunting behavior. GWO uses three types of individual optimizers to find the optimal solution to reduce the objective function (Xie et al., 2021). In a wolf's pack alpha, beta, and omega in the exact mentioned order are responsible for hunting

#### Algorithm 1: PSO Optimization Algorithm

---

```

Max_iter, Set the maximum iteration number
forall  $P_i \leftrightarrow$  Position;  $V_i \leftrightarrow$  Velocity;
 $F \leftarrow$  Cost Function;
Set  $L_i(t)$ ,  $G_i(t)$  in the swarm;
while iter < Max_iter do
    Update inertia weight;
     $W \leftarrow X_{iter}$ 
    for i = 1: Number of particle do
        Update  $V_i$ ;
         $V_{i,j}(t+1) =$ 
             $W * V_{i,j}(t) + R_1 * C_1 * (l_{i,j}(t) - P_{i,j}(t)) + R_2 * C_2 * (G_{i,j}(t) - P_{i,j}(t))$ 
        ;
        Update  $P_i$ ;
         $P_{i,j}(t+1) = P_{i,j}(t) + V_{i,j}(t+1)$ ;
        Update Cost Function;
         $F \leftarrow$  Cost Function();
        if  $P_i$  better than  $L_i(t)$  then
            Update  $L_i(t)$ ;
             $L_i(t) \leftarrow P_i$ 
        end
        if  $P_i$  better than  $G_i(t)$  then
            Update  $G_i(t)$ ;
             $G_i(t) \leftarrow P_i$ 
        end
    end
    iter = iter + 1;
end

```

---

prey (Makhadmeh et al., 2023). GWO employs a three-agent strategy to identify the most suitable solution; these agents represent the top-ranked, second-ranked, and third-ranked solutions respectively (Alo-mari et al., 2022). To update the location of the optimizer in the next iteration, GWO sums all of the previous optimizer solutions and divides the results by 3. The formula for updating each particle location is mentioned as follows:

$$\vec{D} = |\vec{C} \cdot \vec{X} p(t) - \vec{X}(t)| \quad (5)$$

$$\vec{X}(t+1) = \vec{X} p(t) - \vec{A} \cdot \vec{D} \quad (6)$$

where  $t$  denotes current iteration,  $\vec{A}$  and  $\vec{C}$  are coefficient vectors,  $\vec{X}_p$  is the position vector of the prey and  $\vec{X}$  is the position of wolf. The formula for calculations of Vectors  $\vec{A}$  and  $\vec{B}$  are mentioned as follow:

$$\vec{A} = 2 \vec{a} \cdot \vec{r}_1 - \vec{a} \quad (7)$$

$$\vec{C} = 2 \vec{r}_2 \quad (8)$$

where components of  $\vec{a}$  are linearly decreased from 2 to 0 through iterations.  $r_1$  and  $r_2$  are designated based on random values between the range of 0 and 1. Vector  $\vec{A}$  controls the trade-off between exploration and exploitation while  $\vec{C}$  always adds some degree of randomness. This is necessary because the agents can get stuck in the local optima and most of the metaheuristics have a way of avoiding it. As mentioned GWO uses three individual optimizers to find the final optimized solution. Thus the final solution is the combination of all of the individual ones. The equation to find an individual solution is mentioned as follows:

$$\vec{D}_\alpha = |\vec{C} \cdot \vec{X}_\alpha(t) - \vec{X}(t)| \quad (9)$$

$$\vec{D}_\beta = |\vec{C} \cdot \vec{X}_\beta(t) - \vec{X}(t)| \quad (10)$$

$$\vec{D}_\Omega = |\vec{C} \cdot \vec{X}_\Omega(t) - \vec{X}(t)| \quad (11)$$



$$\vec{X}_1 = \vec{X}_\alpha - \vec{A}_1 \cdot \vec{D}_\alpha \quad (12)$$

$$\vec{X}_2 = \vec{X}_\beta - \vec{A}_2 \cdot \vec{D}_\beta \quad (13)$$

$$\vec{X}_3 = \vec{X}_\omega - \vec{A}_3 \cdot \vec{D}_\omega \quad (14)$$

$$\vec{X}_{p(t)} = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3} \quad (15)$$

The cost function of the GWO is the same as PSO. Thus, the proposed method increases the true negative and true positive prediction rates simultaneously (Chantar et al., 2020; Rashaideh et al., 2018). The optimization algorithm using GWO is presented below.

---

**Algorithm 2: GWO Optimization Algorithm**


---

$Max_{iter}$ , Set the maximum iteration number

$X_\alpha$ , Best search agent

$X_\beta$ , second Best search agent

$X_\omega$ , Third Best search agent

$F \leftarrow Cost_{Function}$ ;

Initialize random sets for  $A$ ,  $a$ , and  $c$ ;

Calculate  $F$  for each agent;

**while**  $iter < Max_{iter}$  **do**

**for**  $i = 1: 3$  **do**

        Initialize  $r_1$  and  $r_2$  randomly;

        Update  $X_1$ ;

$\vec{X}_1 = \vec{X}_\alpha - \vec{A}_1 \cdot \vec{D}_\alpha$ ;

        Update  $X_2$ ;

$\vec{X}_2 = \vec{X}_\beta - \vec{A}_2 \cdot \vec{D}_\beta$ ;

        Update  $X_3$ ;

$\vec{X}_3 = \vec{X}_\omega - \vec{A}_3 \cdot \vec{D}_\omega$ ;

        Update  $X_{p(t)}$ ;

$\vec{X}_{p(t)} = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3}$ ;

        Update  $a$ ,  $A$ ,  $c$ ;

        Calculate  $F_{agents}$ ;

        Update  $X_\alpha$ ,  $X_\beta$ ,  $X_\omega$

**end**

$iter = iter + 1$ ;

**end**

---

As shown in Algorithm 2, GWO is searching for the best strategy to optimize the customized cost function. However, the process of searching is separated between 3 agents that require more time to search the environment for the best solution. Similar to PSO the maximum number of iterations was set to 100.

### 3.7. Classifier

The proposed method uses an ML model in the final hateful context detection. In this research, we used RF (Hu and Szymczak, 2023) for final detection. RF is an ensemble learning technique that combines multiple decision trees to obtain a more accurate prediction than can be made by an individual tree. It works by building multiple decision trees using randomly selected subsets of data from the original dataset, each of which provides an independent prediction. The predictions are then averaged to generate a single prediction that is more accurate than the individual predictions. The number of trees used, the parameters of the trees (such as the maximum depth or maximum number of features), and the method used to combine the predictions all affect the performance of the model.

**Table 2**

Results of screening using the CNN Attention for each class.

Class name	Dataset	Accuracy (%)	Precision (%)	Sensitivity (%)	F1-score (%)
Normal	L-SHAB	96.76	96.79	97.40	96.42
Abusive	L-SHAB	97.01	97.10	97.04	97.02
Hateful	L-SHAB	91.12	91.26	91.05	91.13
Normal	T-HSAB	96.01	96.02	96.01	96.01
Abusive	T-HSAB	97.45	97.49	97.42	97.45
Hateful	T-HSAB	94.03	94.17	94.02	94.08

## 4. Experiment results

### 4.0.1. CNN attention

In this research, we proposed a combination of an optimized DL model with GWO for feature extraction and an ML model for discriminating between hateful, abusive, and normal Arabic context detection. The training section of the proposed method has two phases. First, the suggested CNN attention mechanism is trained using the Nesterov Adaptive Moment Estimation algorithm (Sharma et al., 2022). To validate the proposed method we used 80% of all datasets for training and the remaining 20% for testing. The best results out of the 5-fold cross-validation is reported for accuracy and precision, sensitivity, and F1-score. We limited our training of the CNN attention to 1000 epochs, but we imposed a cutoff for when the process should quit. After 100 epochs of no improvement in training and validation loss, the model would cease training. The initial setting for the learning rate is 0.1 and the value for the learning rate will decrease gradually in every 50 epochs by 0.3 (Park et al., 2021). We set the batch size to 64. Metrics such as accuracy, precision, sensitivity, F1-score, and confusion matrix have been used to validate experimental results in this research. The formula for calculating these criteria is mentioned as follows:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad (16)$$

$$Precision = \frac{TP}{TP + FP} \quad (17)$$

$$Sensitivity = \frac{TP}{TP + FN} \quad (18)$$

$$F1\_score = \frac{2 * TP}{2 * TP + FP + FN} \quad (19)$$

Where  $TP$ ,  $TN$ ,  $FP$ , and  $FN$  refer to the True Positive, True Negative, False Positive, and False Negative prediction. The result of training and validating the proposed method in the Arabic context is shown in Fig. 7. As shown in Fig. 7, the proposed Deep learning method has shown promising results in detecting hateful Arabic contexts. However, the model is overfitted after 30 epochs of training for the L-SHAB dataset. Overfitting occurs for T-HSAB after 35 epochs of training. The validation accuracy before overfitting is 96.48% and 96.74% for L-SHAB and T-HSAB. The accuracy of detecting hateful context is lower due to its lower presence in the dataset. Nevertheless, the experimental results of the DL model demonstrate that CNN attention is successful in boosting the overall accuracy of Arabic hateful, normal, and abusive detection. Detailed results of the DL classifier are shown in Table 2.

As shown in Table 2, The evaluated DL model increased the accuracy of the overall classification, but it appears to have a bias towards false positive rate predictions: the true negative rate is lower than the true positive rate.

### 4.0.2. CNN Attention+PSO

The next step in tuning the proposed DL model belongs to the use of a combination of metaheuristic algorithms with the RF model. As shown in Table 1, the evaluated DL model is overfitted. To avoid overfitting and use only the best features for training the combination of PSO with RF has been used as the final classifier. PSO tries to use the best subset of features for final classification. PSO evaluates the

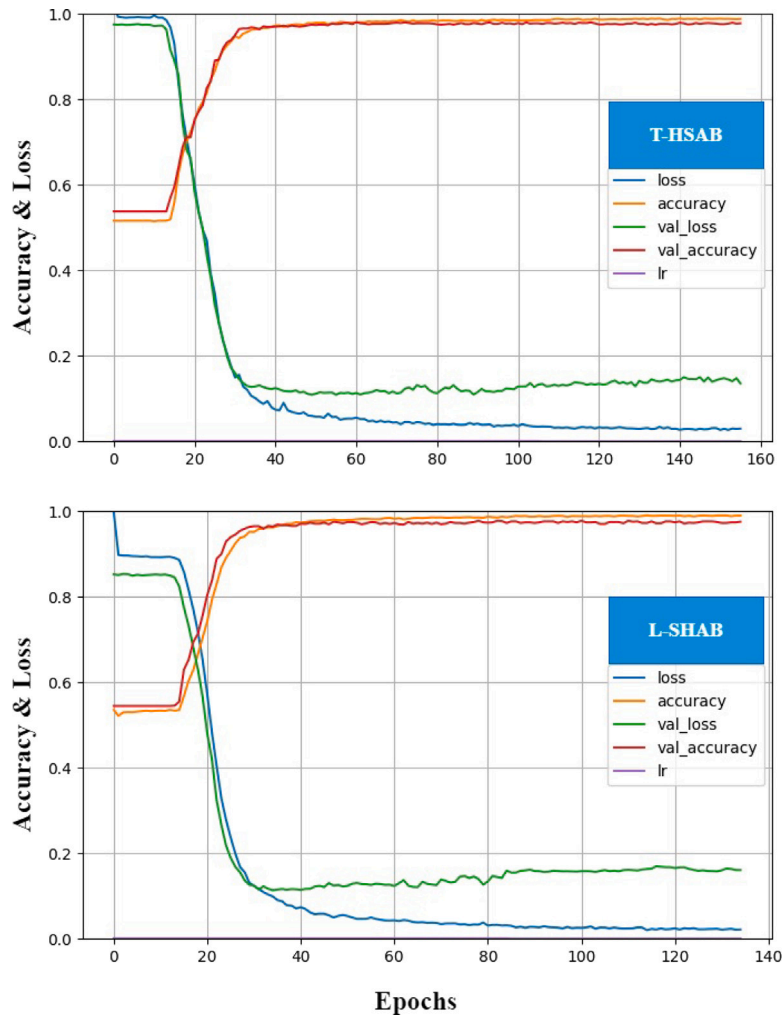


Fig. 7. Learning diagram for CNN Attention model.

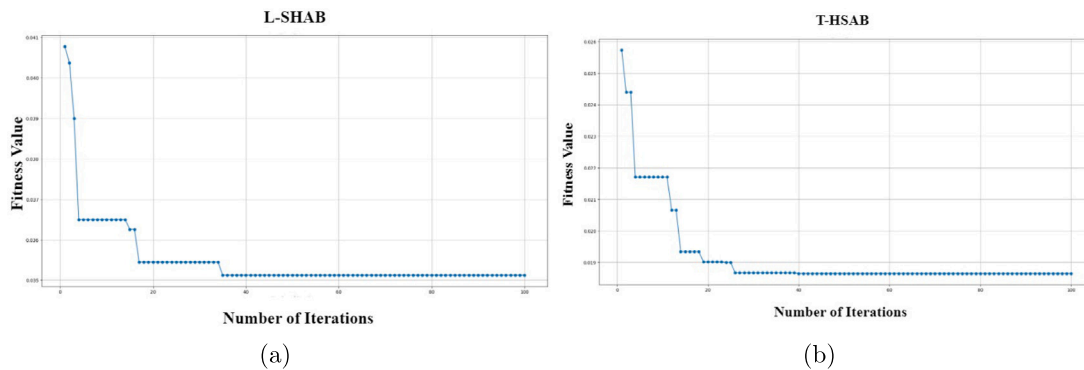


Fig. 8. Result of fitness function using PSO on (a) L-SHAB. (b) T-HSAB.

use of various feature sets and the best combination of sub-features that decrease the cost function is the final result. The result of optimizing RF using various feature sets is shown in Fig. 8. As shown in Fig. 8, The fitness value gradually decreases over each iteration. However, after certain iterations, the process of optimizing the model was halted. The hyper-parameters to train the RF are :  $maximum_{depth}=15$ ,  $maximum_{features}=32$ ,  $minimum_{samplesleaf}=2$ , and number of estimators= 100. Detailed results of the CNN Attention with PSO classifier are shown in Table 3. As shown in Table 3, combining the proposed method using PSO, decreased the unstable performance on precision

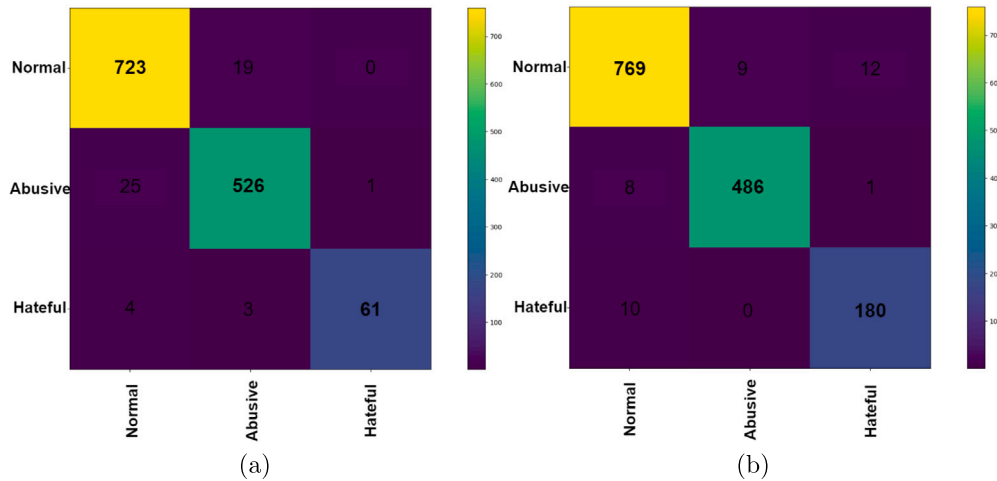
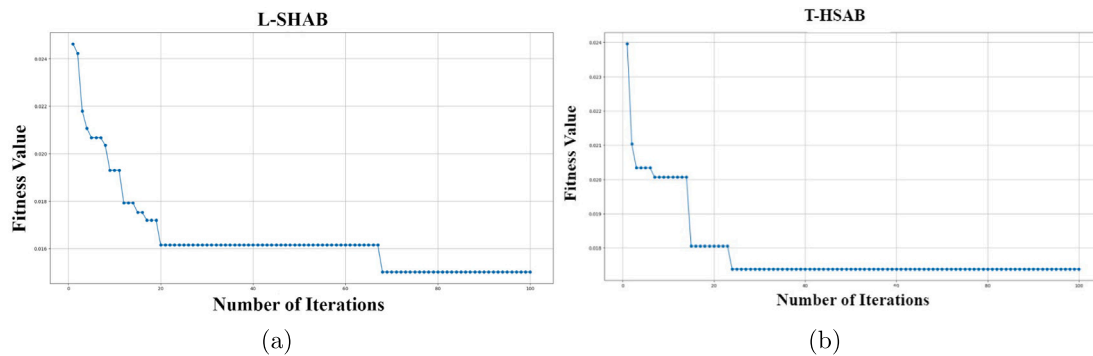
and sensitivity. The detailed result on the confusion matrix for CNN Attention + PSO is shown in Fig. 9.

As shown in Fig. 9, using PSO increased the true perdition rate in Arabic hateful versus abusive contexts discrimination ability. In this case, the specificity ratio(true negative prediction) for a hateful class is increased as well. Also, based on a result of the normalized confusion matrix the result of false positive prediction for the hateful class is less than 6% for the abusive class. The optimization run-time for the proposed method is 7 min and 11 s. It took 4 min and 18 s to pre-train the CNN Attention model on the L-SHAB dataset. It took 3 min and 22 s

**Table 3**

Results of Arabic hateful, abusive, and normal context discrimination using the proposed CNN Attention+PSO model.

Dataset	Model	Accuracy (%)	Precision (%)	Sensitivity (%)	F1-score (%)
L-SHAB	CNN Attention +PSO	96.17	96.18	96.17	96.17
T-HSAB	CNN Attention +PSO	97.30	97.36	97.35	97.35

**Fig. 9.** Result confusion matrix result for CNN Attention +PSO (a) L-SHAB. (b) T-HSAB.**Fig. 10.** Result of fitness function using GWO on (a) L-SHAB. (b) T-HSAB.**Table 4**

Results of Arabic hateful, abusive, and normal context discrimination using the proposed CNN Attention +GWO model.

Dataset	Model	Accuracy (%)	Precision (%)	Sensitivity (%)	F1-score (%)
L-SHAB	CNN Attention + GWO	97.16	97.17	97.13	97.15
T-HSAB	CNN Attention + GWO	97.83%	97.84%	97.83	97.83%

to train the CNN Attention model on the T-HSAB dataset using V-100 GPU.

#### 4.0.3. CNN Attention+GWO

In this section, we explain the process of combining CNN Attention as a feature extractor with optimized RF with GWO to extract proper features and intelligently choose the best feature for final classification. The setting for RF hyper-parameters are:  $maximum_{depth}=15$ ,  $maximum_{features}=32$ ,  $minimum_{samplesleaf}=2$ , and number of estimators=100. The number of iterations to train the final GWO is 100. The result of optimizing the cost function is shown in Fig. 10. As shown in Fig. 10, the optimization duration for L-SHAB is 67 iterations before stopping the optimization process. The optimization duration for T-HSAB is 23 iterations before stopping the tuning process. Detailed results of the CNN Attention with GWO are shown in Table 4.

The detailed result for true positive and negative rate prediction has shown in Fig. 11. As shown in Fig. 11, by using GWO of detecting the

results of detecting the true hateful instances are increased compared to the same result by PSO for the L-SHAB dataset. The performance of detecting the abusive class is improved by the GWO on the T-HSAB dataset as well. By comparing the results in Tables 4 and 5, the overall performance based on accuracy has increased by approximately 1%. The optimization run-time for the proposed method is 41 min and 48 s. As has explained before the pretraining of the CNN Attention model on the L-SHAB dataset was completed in 4 min and 18 s, while the training on the T-HSAB dataset took 3 min and 22 s using V-100 GPU. Comparing the results of the proposed optimized model with GWO and CNN attention, the difference between the true positive detection rate of the hateful and other classes has decreased.

#### 4.1. Feature selection analyzing

We used the output of the last dense layer as the input of the final classifier. Thus, the number of extracted features in the last

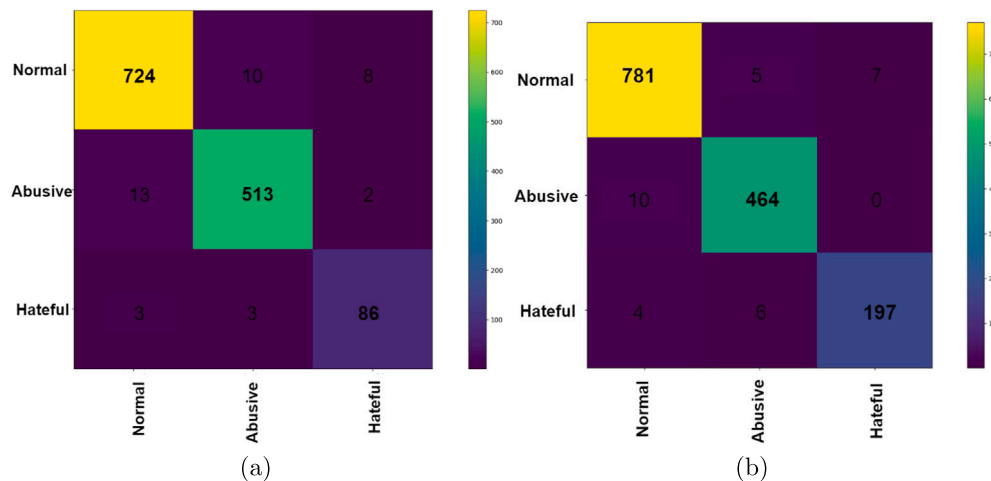


Fig. 11. Result confusion matrix result for CNN Attention + GWO (a) L-SHAB. (b) T-HSAB.

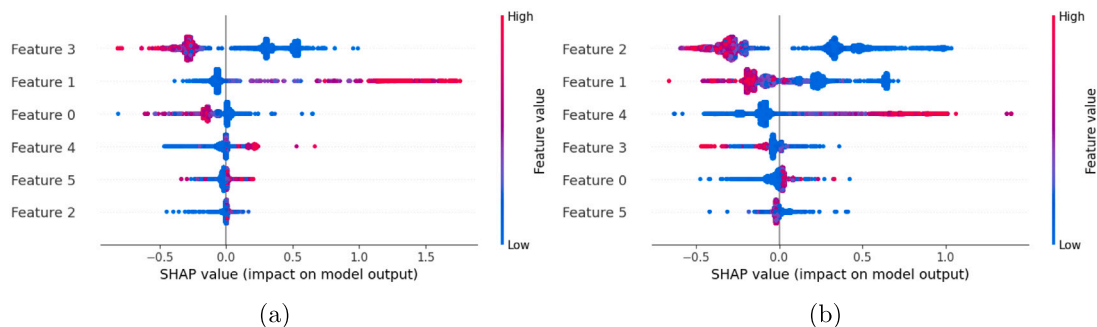


Fig. 12. Result of feature selection using GWO on (a) L-SHAB. (b) T-HSAB.

layer is 32 features. GWO uses the best subset of features out of this feature set. The number of chosen features by GWO using the L-SHAB dataset is 6. The number of chosen features by GWO using the T-HSAB dataset is 6. The importance of each feature inside the chosen subset of features is done using a shapely algorithm (Sundararajan and Najmi, 2020). Shapely value is an algorithm used in feature selection that helps to identify input features or attributes that are most important for predicting the output or target variable. It works by calculating a score for each feature based on its classification accuracy contribution. The higher the score, the more important the feature is for predicting outcomes. The score can range from 0 to 1, with 1 representing the most important feature and 0 representing the least. Shapely value is useful for identifying the most important features in a dataset that can be used in predictive models. The results of the shapely algorithm are shown in Fig. 12. As shown 12, the highest related features for discriminating between normal, hateful, and abusive using the L-SHAB dataset are featured 3, 1, and 0. The highest related features on the T-HSAB dataset are features 4, 2, and 0. Thus, GWO tries to find the best-related feature from the extracted features, usually in the first part or last part of the extracted feature. Thus, GWO is trying to find the best features based on the most important words related to hateful or abusive contexts at the beginning or at the end of the sentences.

## 5. Discussion

In this section, we compare the results of the proposed method with other similar research by discussing its effectiveness in distinguishing between normal and hateful Arabic contexts, a task that is more difficult than distinguishing between normal and hateful English contexts. The reasons behind this matter are:

- Lack of datasets and pretrained models to understand various Arabic nation languages.
- Sophisticated morphology of Arabic contexts with different alphabetic shapes for the same alphabet.
- The cultural difference between Arab nations and other nations.

In this study, we utilized previously existing Arabic tweets from three categories: normal, offensive, and hateful. It was difficult to distinguish between hateful and non-hateful contexts in the Arabic language. As mentioned, one of the problems in the Arabic context is the lack of a proper dataset volume. The original datasets in this research consisted of 5846 and 6023 tweets respectively. To overcome the limited availability of the original dataset, we applied sampling methods for augmentation. This enabled us to address the imbalanced distribution between normal and hateful contexts in the Arabic language effectively. We employed a word embedding model to convert Arabic context to numerical values. To discriminate between normal, abusive, and hateful Arabic contexts, we first trained a combination of CNN and attention layers. We then used this pretrained model as a feature extractor to obtain the necessary features for classification. To detect hateful Arabic contexts, we utilized optimized random forests with particle swarm optimization and grey wolf optimization to evaluate performance. Experimental results have indicated 97.83% accuracy for hateful context detection using CNN Attention + optimized RF with GWO and 97.30% accuracy using CNN Attention + optimized RF with PSO as the best result on T-HSAB datasets. To compare the performance of the proposed method with similar research, we gathered the result of similar research for Arabic context discrimination using the same dataset.

As shown in Table 5, the proposed model outperformed similar research with 6.83% on accuracy and 6.83% on F1-score using the

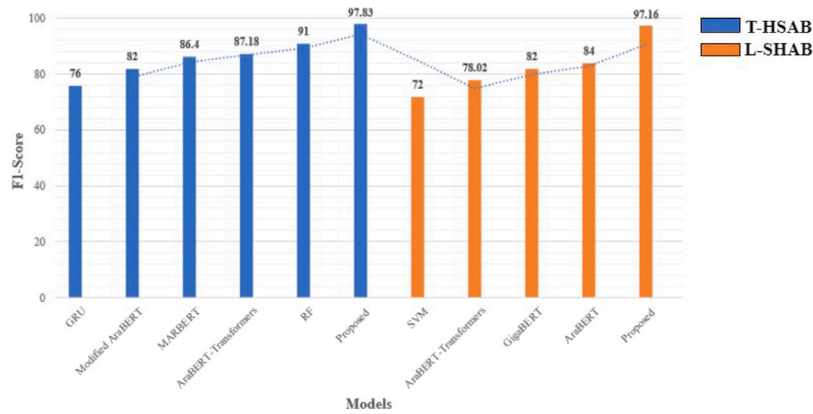


Fig. 13. Comparison between similar research and the proposed method based on F1-score.

**Table 5**  
Comparison between similar hateful Arabic context discrimination.

Reference	Dataset	Model	Accuracy (%)	Sensitivity (%)	F1-Score (%)
<b>Proposed</b>	T-HSAB	<b>CNN Attention+RF+GWO</b>	<b>97.83</b>	<b>97.83</b>	<b>97.83</b>
<b>Proposed</b>	L-SHAB	<b>CNN Attention+RF+GWO</b>	<b>97.16</b>	<b>97.13</b>	<b>97.15</b>
Alhejaili et al(2022)	T-HSAB	RF	91	–	91
Aldjanab et al(2021)	T-HSAB	AraBERT with transformers	87.46	87.42	87.18
Elzayady et al(2023b)	T-HSAB	MARBERT	–	–	86.4
Husain et al (2021)	L-SHAB	AraBERT	86	–	84
Elzayady et al(2023a)	T-HSAB	Modified AraBERT	–	–	82
Abdelhamid et al(2022)	L-SHAB	GigaBERT	–	–	81
Aldjanab et al(2021)	L-SHAB	AraBERT with transformers	78.54	78.5	78.02
Albadi et al(2019)	T-HSAB	GRU	77	75	76
Khalafat et al(2021)	L-SHAB	SVM	68	–	72
Khezzar et al(2023)	arHateDataset	AraBERT	91	–	–
Salomon et al(2022)	Tunisian reference dataset	AraBERT	–	–	99
Almaliki et al(2023)	Arabic Hate-Speech	AraBERT–mini model	98.6	–	98.6

T-HSAB dataset. The improvement by the proposed model is 11.16% and 13.15% based on accuracy and F1-score on the L-SHAB dataset respectively. Also, by checking the evaluated results by similar researchers, using the AraBERT architecture to generate a new model has shown promising results for just hateful context detection (Khezzar et al., 2023; Almaliki et al., 2023). The proposed model employed an objective function to enhance the rate of true negative and true positive predictions simultaneously, thereby resulting in greater sensitivity and F1-score than similar research in terms of accuracy. The proposed method demonstrates improved performance in distinguishing between abusive and normal Arabic contexts, surpassing the performance of previous models which focused on discriminating between hateful and nonhateful Arabic contexts (Aldjanabi et al., 2021). A descriptive comparison between the proposed method and similar research has shown in Fig. 13.

Another improvement in similar research belongs to the process of feature selection in the final step. As we discussed, in Fig. 3, one of the markers for discriminating between hateful and abusive contexts is the specific words that were repeated in the hate and abusive labeled tweets. In this, research, the final optimized classifier is focused on finding this marker and using it for the final discriminator. thus, the improvement in the final stage of classification belongs to the process of using the best features. To ensure the performance of the proposed method for discriminating between harassment, abusive, and normal context we also evaluated the performance of the proposed method on the newly released Let-Mi dataset. (Mulki and Ghanem, 2021). Experimental results on this evaluation are shown in Table 6.

### 5.1. Limitations and future of the work

One drawback of the proposed model is the extra time for optimizing the RF using GWO and PSO. Also, reaching the best solution is not

**Table 6**  
Results of screening using the proposed method for each class on the Let-Mi dataset.

Class name	Dataset	Accuracy (%)	Precision (%)	Sensitivity (%)	F1-score (%)
Normal	Let-Mi	97.41	95.60	97.08	96.34
Active	Let-Mi	96.83	97.11	95.64	96.36

guaranteed by metaheuristics algorithms. Every time the model could get stuck in the local minimum solutions based on the definition of the objective function. Another limitation of the proposed method is the search environment. Increasing the number of extracted features increases the tuning time exponentially. If the output of the CNN Attention model generates more than 100 features, it will take both the PSO and the GWO longer than 5 and 6 h respectively to tune the hybrid algorithm. Furthermore, it will also require more computational power to finalize the classifier.

In the future, we plan to evaluate a well-organized and sizable dataset, which will be based on the modern standard Arabic dialectic. This evaluation will be performed using the proposed method to generate high-performance results in accurately discriminating between hateful, abusive, and normal contexts. The ability of the proposed method to discriminate between racism, hateful, cyberbullying, and abusive contexts will add more value to the proposed method. To ensure the ability of the proposed method to discriminate between various classes of harassment, various metaheuristic methods with different objective functions will be evaluated in the future.

## 6. Conclusion

Arabic hateful context detection can reduce the negative psychological effects of propagating hate and misleading information. Since 2014 spread hateful contexts in the Arabic nations has increased.



The emergence of extremist terrorist organizations such as ISIS has increased the dissemination of false information concerning minorities and different religions. In this study, we proposed a novel approach that combines DL and ML models to detect hateful and abusive content in Arabic contexts. In the beginning, we utilized an elimination process for emojis and Arabic stop words. To take into account the intricate structure of Arabic contexts, we normalized the Arabic alphabet in this study. A model made up of a combination of CNN and attention layers, trained to distinguish between normal, abusive, and hateful contexts in the Arabic language, is proposed. We made use of a pre-trained model to extract the relevant features for our final classification, and then optimized RF using PSO and GWO to detect hateful Arabic contexts. We evaluated the performance of the optimized RF for our purpose. Experimental results on T-HSAB have indicated 97.83% accuracy for hateful context detection using CNN Attention + optimized RF with GWO and 97.30% accuracy using CNN Attention + optimized RF with PSO. The experimental results on L-SHAB have indicated 97.16% accuracy using CNN Attention + optimized RF with GWO and 96.17% accuracy using CNN Attention + optimized RF with PSO. Compared to similar research the proposed model has outperformed the other DL and ML models such as SVM, KNN, and AraBERT with transfer learning (see Table 5). In the future, we will evaluate different metaheuristics algorithms to see if they can further reduce the objective function. Furthermore, the proposed method will be evaluated using a better-structured Arabic corpus with a larger volume of data to ensure reliable results. To this end, we will collect an organized dataset based on the modern standard Arabic dialect to ensure that the proposed method produces high-performance results for discriminating between hateful, abusive, and normal contexts.

### CRedit authorship contribution statement

Abeer Aljohani and Nawaf Alharbe carried out the data collection, participated in the proposed solution and drafted the manuscript. Rabia Emhamed and Abeer Aljohani conceived the proposed algorithm and participated in its design and helped to draft the manuscript. Abeer Aljohani and Mashael Khayyatd participated in the design of the study and performed the performance analysis. All authors read and approved the final manuscript.

### Declaration of competing interest

The author declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

### References

- AbdelHamid, M., Jafar, A., Rahal, Y., 2022. Levantine hate speech detection in twitter. *Soc. Netw. Anal. Min.* 12 (1), 121.
- Abdul-Mageed, M., Elmadany, A., Nagoudi, E.M.B., 2020. ARBERT & MARBERT: deep bidirectional transformers for Arabic. *arXiv preprint arXiv:2101.01785*.
- Al-Hassan, A., Al-Dossari, H., 2021. Detection of hate speech in Arabic tweets using deep learning. *Multimedia Syst.* 1–12.
- Albadi, N., Kurdi, M., Mishra, S., 2019. Investigating the effect of combining GRU neural networks with handcrafted features for religious hatred detection on Arabic Twitter space. *Soc. Netw. Anal. Min.* 9 (1), 1–19.
- Aldjanabi, W., Dahou, A., Al-qaness, M.A., Elaziz, M.A., Helmi, A.M., Damašević, R., 2021. Arabic offensive and hate speech detection using a cross-corpora multi-task learning model. In: *Informatics*, vol. 8, (4), MDPI, p. 69.
- Alhejaili, R., Alsaedi, A., Yafooz, W.M., 2022. Detecting hate speech in arabic tweets during COVID-19 using machine learning approaches. In: *Proceedings of Third Doctoral Symposium on Computational Intelligence: DoSCI 2022*. Springer, pp. 467–475.
- Allu, R., Padmanabhuni, V.N.R., 2022. Predicting the success rate of a start-up using LSTM with a swish activation function. *J. Control Decis.* 9 (3), 355–363.
- Almaliki, M., Almars, A.M., Gad, I., Atlam, E.-S., 2023. ABMM: Arabic BERT-mini model for hate-speech detection on social media. *Electronics* 12 (4), 1048.
- Alomari, O.A., Elnagar, A., Afyouni, I., Shahin, I., Nassif, A.B., Hashem, I.A., Tubishat, M., 2022. Hybrid feature selection based on principal component analysis and grey wolf optimizer algorithm for Arabic news article classification. *IEEE Access* 10, 121816–121830.
- Antoun, W., Baly, F., Hajj, H., 2020. Arabert: Transformer-based model for arabic language understanding. *arXiv preprint arXiv:2003.00104*.
- Beattie, J.R., Esmonde-White, F.W., 2021. Exploration of principal component analysis: deriving principal component analysis visually using spectra. *Appl. Spectrosc.* 75 (4), 361–375.
- Bhojanapalli, S., Yun, C., Rawat, A.S., Reddi, S., Kumar, S., 2020. Low-rank bottleneck in multi-head attention models. In: *International Conference on Machine Learning*. PMLR, pp. 864–873.
- Chantar, H., Mafarja, M., Alsawalqah, H., Heidari, A.A., Aljarah, I., Faris, H., 2020. Feature selection using binary grey wolf optimizer with elite-based crossover for Arabic text classification. *Neural Comput. Appl.* 32, 12201–12220.
- Chefer, H., Gur, S., Wolf, L., 2021. Generic attention-model explainability for interpreting bi-modal and encoder-decoder transformers. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. pp. 397–406.
- Chen, S., Webb, G.L., Liu, L., Ma, X., 2020. A novel selective naïve Bayes algorithm. *Knowl.-Based Syst.* 192, 105361.
- Chowdhary, K., 2020. Natural language processing. *Fundam. Artif. Intell.* 603–649.
- Chita, S., Motahhir, S., El Hammoumi, A., Chouder, A., Benyoucef, A.S., El Ghzizal, A., Derouich, A., Abouhawwash, M., Askar, S., 2022. A novel hybrid GWO-PSO-based maximum power point tracking for photovoltaic systems operating under partial shading conditions. *Sci. Rep.* 12 (1), 1–15.
- Cunningham, P., Delany, S.J., 2021. K-nearest neighbour classifiers-a tutorial. *ACM Comput. Surv.* 54 (6), 1–25.
- Elzayady, H., Mohamed, M.S., Badran, K.M., Salama, G.I., 2023a. A hybrid approach based on personality traits for hate speech detection in arabic social media. *Int. J. Electr. Comput. Eng.* 13 (2), 1979.
- Elzayady, H., Mohamed, M.S., Badran, K., Salama, G., Abdel-Rahim, A., 2023b. Arabic hate speech identification by enriching MARBERT model with hybrid features. In: *Intelligent Sustainable Systems: Selected Papers of WorldS4 2022*, Volume 2. Springer, pp. 559–566.
- Farag, M.M., 2023. A tiny matched filter-based CNN for inter-patient ECG classification and arrhythmia detection at the edge. *Sensors* 23 (3), 1365.
- Georgieva-Trifonova, T., Duraku, M., 2021. Research on N-grams feature selection methods for text classification. In: *IOP Conference Series: Materials Science and Engineering*, vol. 1031, (1), IOP Publishing, 012048.
- Ghaddar, A., Wu, Y., Rashid, A., Bibi, K., Rezagholizadeh, M., Xing, C., Wang, Y., Xinyu, D., Wang, Z., Huai, B., et al., 2021. JABER: junior arabic bert. *arXiv preprint arXiv:2112.04329*.
- Gomes, L., da Silva Torres, R., Côrtes, M.L., 2023. BERT-and TF-IDF-based feature extraction for long-lived bug prediction in FLOSS: a comparative study. *Inf. Softw. Technol.* 160, 107217.
- Gubatan, J., Levitte, S., Patel, A., Balabanis, T., Wei, M.T., Sinha, S.R., 2021. Artificial intelligence applications in inflammatory bowel disease: emerging technologies and future directions. *World J. Gastroenterol.* 27 (17), 1920.
- Haddad, H., Mulki, H., Oueslati, A., 2019. T-hsab: A tunisian hate speech and abusive dataset. In: *Arabic Language Processing: From Theory To Practice: 7th International Conference, ICALP 2019*, Nancy, France, October 16–17, 2019, *Proceedings 7*. Springer, pp. 251–263.
- Haddad, B., Orabe, Z., Al-Abood, A., Ghneim, N., 2020. Arabic offensive language detection with attention-based deep neural networks. In: *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*. pp. 76–81.
- Hu, J., Szymczak, S., 2023. A review on longitudinal data analysis with random forest. *Brief. Bioinform.* 24 (2), bbad002.
- Husain, F., 2020. Arabic offensive language detection using machine learning and ensemble machine learning approaches. *arXiv preprint arXiv:2005.08946*.
- Husain, F., Uzuner, O., 2021. Transfer learning approach for arabic offensive language detection system-BERT-based model. *arXiv preprint arXiv:2102.05708*.
- Jang, B., Kim, I., Kim, J.W., 2019. Word2vec convolutional neural networks for classification of news articles and tweets. *PLoS One* 14 (8), e0220976.
- Khalafat, M., Ja'far, S.A., Al-Sayyed, R., Eshtay, M., Kobbaey, T., 2021. Violence detection over online social networks: An Arabic sentiment analysis approach. *ijim* 15 (14), 91.
- Khezzer, R., Moursi, A., Al Aghbari, Z., 2023. ArHateDetector: detection of hate speech from standard and dialectal arabic tweets. *Disc. Internet Things* 3 (1), 1.
- Litvak, M., Vanetik, N., Liebeskind, C., Hmdia, O., Madeghem, R.A., 2022. Offensive language detection in hebrew: can other languages help? In: *Proceedings of the Thirteenth Language Resources and Evaluation Conference*. pp. 3715–3723.
- Liu, Y., Pei, A., Wang, F., Yang, Y., Zhang, X., Wang, H., Dai, H., Qi, L., Ma, R., 2021. An attention-based category-aware GRU model for the next POI recommendation. *Int. J. Intell. Syst.* 36 (7), 3174–3189.
- Makhadmeh, S.N., Al-Betar, M.A., Doush, I.A., Awadallah, M.A., Kassaymeh, S., Mirjalili, S., Zitar, R.A., 2023. Recent advances in Grey Wolf Optimizer, its versions and applications. *IEEE Access*.
- Mansourifar, H., Shi, W., 2020. Deep synthetic minority over-sampling technique. *arXiv preprint arXiv:2003.09788*.

- Messaoudi, A., Haddad, H., Hmida, M.B.H., 2020. iCompass at SemEval-2020 task 12: From a syntax-ignorant n-gram embeddings model to a deep bidirectional language model. In: *Proceedings of the Fourteenth Workshop on Semantic Evaluation*. pp. 1978–1982.
- Mulki, H., Ghanem, B., 2021. Let-mi: an arabic levantine twitter dataset for misogynistic language. *arXiv preprint arXiv:2103.10195*.
- Mulki, H., Haddad, H., Ali, C.B., Alshabani, H., 2019. L-hsab: A levantine twitter dataset for hate speech and abusive language. In: *Proceedings of the Third Workshop on Abusive Language Online*. pp. 111–118.
- Mursi, K.T., Alahmadi, M.D., Alsubaei, F.S., Alghamdi, A.S., 2022. Detecting islamic radicalism arabic tweets using natural language processing. *IEEE Access* 10, 72526–72534.
- Nedjar, I., Mahmoudi, S., Chikh, M.A., 2022. A topological approach for mammographic density classification using a modified synthetic minority over-sampling technique algorithm. *Int. J. Biomed. Eng. Technol.* 38 (2), 193–214.
- Niu, Z., Zhong, G., Yu, H., 2021. A review on the attention mechanism of deep learning. *Neurocomputing* 452, 48–62.
- Ofer, D., Brandes, N., Linial, M., 2021. The language of proteins: NLP, machine learning & protein sequences. *Comput. Struct. Biotechnol. J.* 19, 1750–1758.
- Otchere, D.A., Ganat, T.O.A., Gholami, R., Ridha, S., 2021. Application of supervised machine learning paradigms in the prediction of petroleum reservoir properties: Comparative analysis of ANN and SVM models. *J. Pet. Sci. Eng.* 200, 108182.
- Park, J., Chun, J., Kim, S.H., Kim, Y., Park, J., 2021. Learning to schedule job-shop problems: representation and policy learning using graph neural network and reinforcement learning. *Int. J. Prod. Res.* 59 (11), 3360–3377.
- Povey, D., Hadian, H., Ghahremani, P., Li, K., Khudanpur, S., 2018. A time-restricted self-attention layer for ASR. In: *2018 IEEE International Conference on Acoustics, Speech and Signal Processing. ICASSP, IEEE*, pp. 5874–5878.
- Priyadarshi, N., Padmanaban, S., Holm-Nielsen, J.B., Blaabjerg, F., Bhaskar, M.S., 2019. An experimental estimation of hybrid ANFIS-PSO-based MPPT for PV grid integration under fluctuating sun irradiance. *IEEE Syst. J.* 14 (1), 1218–1229.
- Rashideh, H., Sawaie, A., Al-Betar, M.A., Abualigah, L.M., Al-Laham, M.M., Al-Khatib, R.M., Braik, M., 2018. A grey wolf optimizer for text document clustering. *J. Intell. Syst.* 29 (1), 814–830.
- Safaldin, M., Otair, M., Abualigah, L., 2021. Improved binary gray wolf optimizer and SVM for intrusion detection system in wireless sensor networks. *J. Ambient Intell. Humaniz. Comput.* 12 (2), 1559–1576.
- Salomon, P.O., Kechaou, Z., Wali, A., 2022. Arabic hate speech detection system based on AraBERT. In: *2022 IEEE 21st International Conference on Cognitive Informatics & Cognitive Computing. ICCI\* CC, IEEE*, pp. 208–213.
- Santos, C.F.G.D., Papa, J.P., 2022. Avoiding overfitting: A survey on regularization methods for convolutional neural networks. *ACM Comput. Surv.* 54 (10s), 1–25.
- Sharma, J., Soni, S., Paliwal, P., Saboor, S., Chaurasiya, P.K., Sharifpur, M., Khalilpoor, N., Afzal, A., 2022. A novel long term solar photovoltaic power forecasting approach using LSTM with Nadam optimizer: A case study of India. *Energy Sci. Eng.*
- Shin, S.Y., Choi, Y.-J., 2021. Comparison of cyberbullying before and after the COVID-19 pandemic in Korea. *Int. J. Environ. Res. Pub. Health* 18 (19), 10085.
- Sundararajan, M., Najmi, A., 2020. The many Shapley values for model explanation. In: *International Conference on Machine Learning. PMLR*, pp. 9269–9278.
- Tavakolian, A., Hajati, F., Rezaee, A., Fasakhodi, A.O., Uddin, S., 2022a. Fast COVID-19 versus H1N1 screening using optimized parallel inception. *Expert Syst. Appl.* 204, 117551.
- Tavakolian, A., Hajati, F., Rezaee, A., Fasakhodi, A.O., Uddin, S., 2022b. Source code for optimized parallel inception: A fast COVID-19 screening software. *Softw. Impacts* 13, 100337.
- Visca, M., Powell, R., Gao, Y., Fallah, S., 2022. Meta-Conv1D energy-aware path planner for mobile robots in unstructured terrains. In: *2022 7th International Conference on Robotics and Automation Engineering. ICRAE, IEEE*, pp. 150–157.
- Wachs, S., Mazzone, A., Milosevic, T., Wright, M.F., Blaya, C., Gámez-Guadix, M., Norman, J.O., 2021. Online correlates of cyberhate involvement among young people from ten European countries: An application of the routine activity and problem behaviour theory. *Comput. Hum. Behav.* 123, 106872.
- Xie, Q., Guo, Z., Liu, D., Chen, Z., Shen, Z., Wang, X., 2021. Optimization of heliostat field distribution based on improved Gray Wolf optimization algorithm. *Renew. Energy* 176, 447–458.
- Zha, W., Liu, Y., Wan, Y., Luo, R., Li, D., Yang, S., Xu, Y., 2022. Forecasting monthly gas field production based on the CNN-LSTM model. *Energy* 124889.