Journal of Soft Computing and Computer Applications

Manuscript 1020

Hate Speech Detection Using Optimized Feature Representation via Spiral-Grey Wolf Optimizer-Based Machine Learning Approaches

Noor S. Farhan

Matheel E. Abdulmunim

Hasanen S. Abdullah

ORIGINAL STUDY

Hate Speech Detection Using Optimized Feature Representation via Spiral-Grey Wolf Optimizer-Based Machine Learning Approaches

Noor S. Farhan [a,*], **Matheel E. Abdulmunim** [b], **Hasanen S. Abdullah** [c]

[a] University of Technology - Iraq, College of Computer Science, Al-Sina'a St., Al-Wehda District, 10066 Baghdad, Iraq
[b] University of Technology - Iraq, College of Computer Science, Department of Multimedia and Digital Media, Al-Sina'a St., Al-Wehda District, 10066 Baghdad, Iraq
[c] University of Technology - Iraq, College of Computer Science, Department of Artificial Intelligence, Al-Sina'a St., Al-Wehda District, 10066 Baghdad, Iraq

ABSTRACT

Hate speech detection is crucial as social media diversifies. This research present a lightweight, scalable system using traditional machine learning methods along with a new approach called Spiral-Grey Wolf Optimizer (S-GWO).

S-GWO effectively selects key features that consider both meaning and content from the Term Frequency Inverse Document Frequency (TF-IDF) space, leading to high-quality representation without excessive computing power.

The proposed system was tested on Arabic and another English datasets using six machine learning methods: SVM, RF, LR, KNN, NB, and SGD. It achieved 92% accuracy and F1 score on the Arabic dataset, while reaching 100% accuracy on the English dataset, significantly reducing hate speech and toxicity.

Overall, the enhanced algorithms improve accuracy and efficiency, offering an effective alternative to costly deep learning models even with noisy and unbalanced data.

Keywords: Hate speech, Feature representation, Arabic dialect, Grey wolf optimizer, Spiral motion, Machine learning

Received 20 June 2025; accepted 13 October 2025.
Available online 26 December 2025

* Corresponding author.
E-mail addresses: cs.24.07@grad.uotechnology.edu.iq (N. S. Farhan), matheel.e.abdulmunim@uotechnology.edu.iq (M. E. Abdulmunim), hasanen.s.abdullah@uotechnology.edu.iq (H. S. Abdullah).

https://doi.org/10.70403/3008-1084.1020

1. Introduction

The rapid growth of social media platforms such as Instagram, Twitter, and Facebook has dramatically changed online communication [1]. However, it has also allowed more hateful, bullying, and discriminatory content to be visible online [2]. Spotting hate speech in Arabic is challenging because its structure is complex, the language has an enormous variety, and most writings online are not formal [3]. Most users combine modern standard Arabic with local languages, expressions, and sometimes different spellings, creating difficulties for Natural Language Processing (NLP) [4]. Because there is a lot of messy and unordered user-generated content, specialized methods are needed to filter this data and select useful features for good classification [5].

To classify hate speech effectively, strong feature representations are crucial. Traditional methods often overlook important signals and struggle with dialect variations.

This paper introduces a system combining TF-IDF with the Spiral-Grey Wolf Optimizer (S-GWO) to improve feature quality and variety while preserving essential language details.

The novel feature and main contribution of this paper lies in development of a general and a flexible system focused on enhancing feature representation using an advanced feature selection algorithm, eliminating the need for complex models. It adapts well to various datasets, allowing seamless integration with different machine or deep learning models.

The approach was tested with light classifiers like SVM, Random Forest, and Naïve Bayes on two datasets: one from Instagram featuring multiple Arabic dialects and another from Twitter. Results demonstrate that the system generalizes effectively across languages, platforms, and resources.

The remainder of this paper is organized as follows: Section 2 introduces the problem formulation, Section 3 presents a literature review, and Section 4 describes the optimization strategy. Sections 5 to 9 cover the methodology, architecture, experiments, results, and discussion. Section 10 concludes.

2. Problem formulation

While hate speech detection models have been improved, they still suffer from some intrinsic limitations, particularly in the Arabic language, which is characterized by its structural complexity and extensive dialects, complicating classification.

Most previous research has focused on a single platform or dialect and only on binary classification, limiting its broader application. Additionally, the representations used are inefficient, and deep learning models require significant computational resources.

To address these challenges, this paper employs a GWO algorithm-based improved spiral motion to optimize feature representation, introducing a lightweight and adaptable system. This system is designed to generate high-quality digital representations that are distinctive enough for the type of data, without necessitating retraining or retuning the classifiers, and in turn, achieving better overall accuracy with less computationally expensive techniques.

The system was evaluated on challenging datasets, both in Arabic with multiple dialects, as well as independent English datasets, and for several labels. The experimental results show the effectiveness over the baseline performance in these datasets, as reported in the original papers introducing them.

The proposed system utilizes lightweight machine learning classifiers, eliminating the need for heavy, cumbersome architectures to achieve overall robust performance. This approach offers the potential for portability of the classifier across various language environments and platforms.

Accordingly, the proposed system is capable of effectively performing heterogeneous multi-class domain classification in Arabic and English.

3. Literature review

The following is based on the brief overview of related work, which is sorted thematically. These articles discussed the importance of feature extraction techniques, hate speech detection, and cyberbullying detection in multilingual and socially diverse environments, as well as heuristic optimization methods for optimizing feature selection.

Scientists have examined machine-learning techniques for detecting hate speech and cyberbullying on Arabic social media in several research studies.

Alakrot et al. [6] reached an F1-score of 0.81 by utilizing n-gram features with feature selection that selects an optimal set of features employing Recursive Feature Elimination (RFE) and Least Absolute Shrinkage and Selection Operator (LASSO) regression, also known as L1 regularization, which, to the best of the authors knowledge, is the only method for abusive language identification in Arabic online forums.

Moreover, Almutiry & Abdel Fattah [7] categorized cyberbullying in Arabic using stemming and preprocessing techniques, achieving an SVM AUC of 0.862.

Makram et al. [8] have proposed a hybrid method employing a large-scale pre-trained Arabic language masking model on Twitter to extract the features of the Arabic tweets in the OSACT2022 joint task dataset [4] (also known as MARBERT). These features are then fed into two classical machine learning classifiers (LR and RF). LR performed best: for offensive (accuracy 80, precision/recall/F1 78) and for hate-speech (accuracy 89, precision 72, recall 80, F1 76). Results support the effectiveness of transformer-derived features when coupled with lightweight classifiers. Guellil et al. [9] presented ara-women-hate, a manually annotated collection of YouTube comments in Arabic, Arabizi, French, and English, addressing hate speech directed at women. Researchers used both deep learning models (Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM) network, and bi-directional LSTM (Bi-LSTM) network) and classical machine learning classifiers (LR, SVM) to validate the corpus. CNN with FasText Skip-Gram embedding produced the highest F1-score (up to 86%) on the unbalanced dataset. AlFarah et al. [10] proposed a framework for detecting cyberbullying using five conventional machine learning models on Arabic tweets and YouTube comments. Notably, NB achieved the maximum Area Under the Curve (AUC) of 89% after the dataset was balanced using TF-IDF features and oversampling. More recently, Shannag et al. [11] presented the Multi-Dialectal Arabic Cyberbullying Corpus (ArCybC) and demonstrated that Arabic embeddings, in conjunction with SVM, produced an accuracy of 86.3%, highlighting the challenges associated with data scarcity and annotation in Arabic cyberbullying detection. Shannaq et al. [12] suggested a two-stage offensive language identification system for Arabic tweets that combines optimized AraVec word embeddings with a Genetic Algorithm (GA)-based optimization of SVM and the eXtreme Gradient Boosting (XGBoost) classifiers. Using SVM with AraVec-SkipGram embeddings, the system achieved a peak accuracy of 88.2% and an F1-score of 87.8%. Both feature enrichment and classifier optimization for Arabic offensive content identification were successfully handled in this work. Ahmad et al. [13] introduced a large-scale Jordanian hate speech dataset (403,688 tweets). They ran comparative baselines across Word2Vec (W2V)/TF-IDF/Arabic Bidirectional Encoder Representations from Transformers (AraBERT) with seven machine learning classifiers and fine-tuned Arabic transformers. W2V features paired with CatBoost were the most stable among machine learning models, while fine-tuned MARBERT yielded the best F1 (~0.61). The study also

highlighted persistent sensitivity to class imbalance in the multiclass setting. Finally, Alghamdi et al. [14] presented AraTar, a corpus designed to support the fine-grained detection of hate speech targets in the Arabic language (11,219 tweets; multi-label), by merging and re-tagging Arabic tweets for two subtasks: hate-type (Task 1) and hate-target (Task 2). Across traditional machine learning, deep models with contextual embeddings, and fine-tuned Language Models (LMs), the best results are achieved by AraBERT—base for Task 1 (micro-F1 = 84.50%) and AraBERT—large for Task 2 (micro-F1 = 86.05%). The study highlights the importance of identifying the target of hate, a topic rarely addressed in Arabic literature.

In terms of feature representation, TF-IDF has been widely used as a popular traditional method. Alsubait & Alfageh [15] compared TF-IDF with Count Vectorizer of Arabic YouTube comments and found better performance with Complement NB (CNB) using TF-IDF (F1 $\approx$ 0.779). Sri [16] suggested using domain-specific word weighting in a modified TF-IDF technique to enhance performance on complex hate speech categories.

Despite advances in feature extraction, the selection of influential features remains a critical factor in the success of a representation. Therefore, optimization techniques have been applied to select the most effective features. Subkhi et al. [17] suggested a hybrid feature selection technique to enhance Arabic text categorization by integrating the Binary Grey Wolf Optimizer (BGWO), Particle Swarm Optimization (PSO), and Sine Cosine Algorithm (SCA), achieving an accuracy of 88.08% using SVM on imbalanced speech data, showing superior performance and convergence compared to individual methods. Similarly, Bajpai et al. [18] employed a Differential Evolution (DE) algorithm to develop an efficient feature selection method for the Internet of Things Intrusion Detection 2020 Dataset (IoTID20). They subsequently labeled the performance with XGBoost, achieving up to 83.72% accuracy by saving time on training.

Although they consider a different application domain, their concentration on improving numeric representations over classification validates the generality of this strategy across applications.

The proposed architecture differs from other related works that focus on a single dialect, platform, or binary label. Instead, the system aims to utilize two database datasets (i.e., Arabic, which encompasses a wide range of dialects, and English) with distinct classes. This framework proposes an improved S-GWO to minimize feature selection, leveraging good power backup with a lightweight model, without relying on deep learning, to address the limitations of generalization and efficiency.

4. Optimization strategy

Optimization is an integral part of machine learning, in which nature-inspired metaheuristic algorithms address complex, high-dimensional problems by selecting effective solutions and tuning parameters to enhance overall model performance [19].

The GWO, a widely known metaheuristic algorithm, is designed by simulating the behavior of grey wolves, which exhibit hierarchical leadership and cooperative hunting [20]. The alpha (α), beta (β), delta (δ), and omega structures of GWO are suitable for balancing the advantages of both exploitation and exploration, and can be adopted to solve various optimization problems. Due to its simplicity, adaptation in parameter tuning, and high convergence rates, it has been popularly used for both feature selection and hyperparameter optimization [21].

However, the original GWO algorithm often falls into stagnation, where the fitness of the alpha wolf remains unchanged for numerous iterations, leading it to become stuck in a local optimum.

To overcome this drawback, a modified version of GWO is proposed in which a spiral updating rule is implemented to add the exploration ability rather than just a stagnant tag (S-GWO), combining three event mechanisms: stagnation detection, adaptive spiral movement, and dynamic dimension updates. This method is discussed below:

4.1. Parameter initialization

There are three phases in the Evolutionary Algorithms (EAs): initialization, iterative updates, and termination.

Initialization in S-GWO:

Initialization is the most vital part of exploration and exploitation in the S-GWO algorithm, aiming to decrease stagnation sensitivity. To do this, the wolves are equally located in the search space, and some parameters need to be adjusted sensibly, like:

- Size of population N
- Lower and Upper limit constraints LP, UP
- Maximum number of iterations T
- Minimum Dmin = 1
- Maximum dimensions change Dmax = 2
- Stagnation limit Smax

Further spiral-specific parameters, such as growth and decay rates (b_explore, b_exploit), and phase bounds, add further tuning of the adaptive movement of the spiral.

This cautious initialization helps prevent early convergence and increases robustness in high-dimensional feature spaces, which is crucial, especially for textual applications such as hate speech detection.

All the parameters are listed in Section 7.2. These parameters regulate the displacement and adaptive update of the spiral directly, as we explained in the following.

4.2. Novelty and search space transformation

As described in the initialization stage, the parameters in Section 7.2 directly regulate the spiral motion and its sensitivity to stagnation. The novelty of the S-GWO lies in the introduction of a dynamic spiral trajectory within the encircling phase, which transforms the wolves' movement from a direct linear convergence toward the leader wolf to a radial-angular movement that combines rotation around the leader with a gradual change in radius. This transformation breaks the linear drag pattern and expands the exploration mechanisms to broader and more diverse ranges, as illustrated in Fig. 1.

This spiral motion can take two adaptive forms depending on the value of the growth factor (b), as shown below and in Fig. 2:

1. Inside-outside ($b > 0$): The radius gradually widens, enhancing exploration and reducing the likelihood of early clustering in non-optimal regions.
2. Outside-inside ($b < 0$): The radius gradually shrinks, concentrating the search in promising regions and intensifying exploitation.

To automate this process, S-GWO determines the spiral direction automatically based on two indicators: the last optimization signal of the fitness function and the search phase, calculated as the ratio of the current iteration to the total number of iterations. The phase threshold is calibrated to suit the nature of the fitness function, and the following rule is then applied: In the early phase, an inside-outside motion is used to promote exploration,

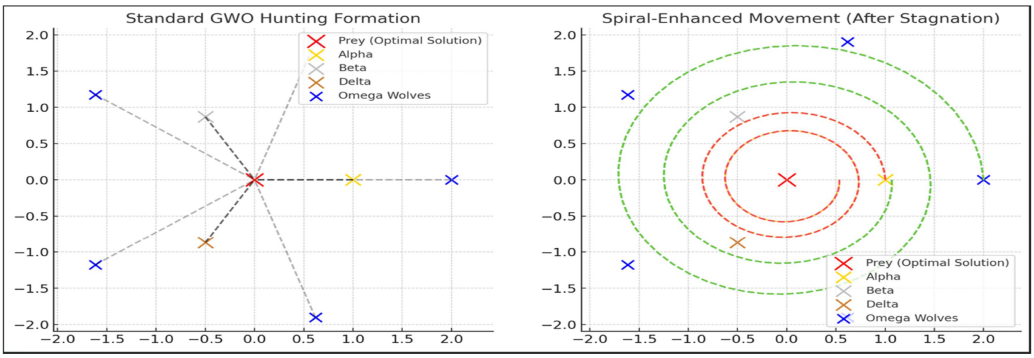

Fig. 1. Standard encircling (left) versus spiral-enhanced motion (right) after stagnation, using an adaptive radius around α.

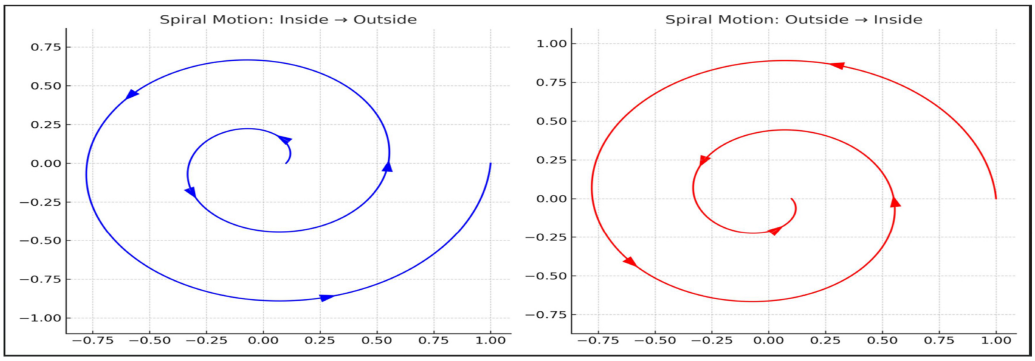

Fig. 2. Spiral directions in S-GWO: Inside→Outside versus Outside→Inside.

and in the late phase, an outside-inside motion is used to encourage exploitation. The spiral stops as soon as improvement occurs, preventing oscillations.

This dual behavior contributes to dynamically reshaping the search space, beginning with a broad exploration phase in the early stages and gradually transitioning to more precise local optimization in the later stages. In doing so, it reduces the risk of stagnation, preserves diversity, and maintains an effective balance between exploration and exploitation.

4.3. Aggregation and behavior monitoring

After initialization and setting up the spiral, S-GWO aggregates the wolf positions at each iteration and evaluates them using the fitness function, updating the hierarchy (α, β, δ). Two tests then govern the process: the first is the stopping criterion, which terminates the optimization process when the maximum number of iterations (T) is reached. The second is the stagnation detection, which tracks whether the wolf's fitness value α remains constant over a predetermined number of consecutive iterations (Smax). This monitoring layer ensures efficient use of updates and forms the basis for the stagnation-based switching described later.

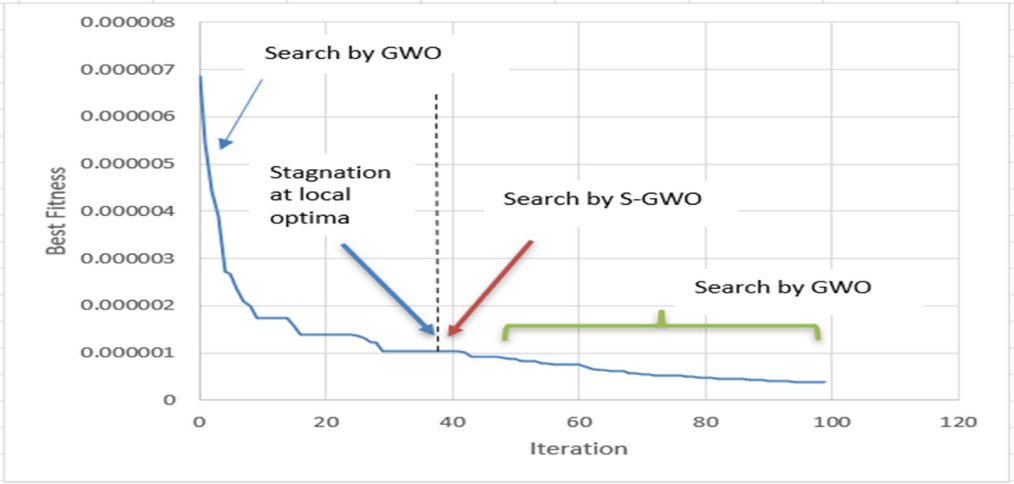

Fig. 3. Proposed scenario of the searching by S-GWO.

4.4. Stagnation-based search switching

Based on the monitoring phase, the S-GWO adaptively switches between standard GWO and spiral-based updates. If no stagnation is observed, the wolves follow the standard GWO rules. However, if the best solution alpha remains unchanged during a predefined stagnation window Smax, the algorithm switches to S-GWO mode, using spiral motion to avoid local optima.

As illustrated in Fig. 3, the process begins with GWO, enters a stagnation phase, and then activates S-GWO to recover progress before resuming standard GWO. This mechanism forms the link between stagnation detection (Section 4.3) and the spiral-based adjustment phase (Section 4.5).

4.5. Spiral-based population modification

Once any stagnation is detected and the algorithm switches to S-GWO, the population update is adjusted using a spiral strategy. This process involves two main steps:

A. Dynamic dimension update

To reduce early scatter in the S-GWO mechanism, only a subset of dimensions is updated for spiral updating at each iteration, rather than modifying all positions. The number of dimensions (W_{Dim}) updated at iteration t is calculated using (Eq. (1)):

$$\mathbf{W_{Dim}} = \mathbf{round}([\mathbf{D_{min}} + (\frac{\boldsymbol{t}}{\boldsymbol{t\ max}})^{2}.(\mathbf{D_{max}} - \mathbf{D_{min}})].\mathbf{D_p}) \quad (1)$$

where t represents the current iteration of the algorithm, t_{max} is the total number of iterations, D_{max} and D_{min} indicate the minimum and maximum ratios of dimensions to be updated, respectively. At the same time, D_p refers to a problem dimension.

Updating starts with a small number of dimensions to achieve a precise local optimization, then gradually expands with iteration to include more dimensions, allowing for a broader exploration of the space, as shown in Fig. 4. (Eq. (1)) specifies only the number of dimensions. Then, the target set is formed and refined in stages: randomly selected in

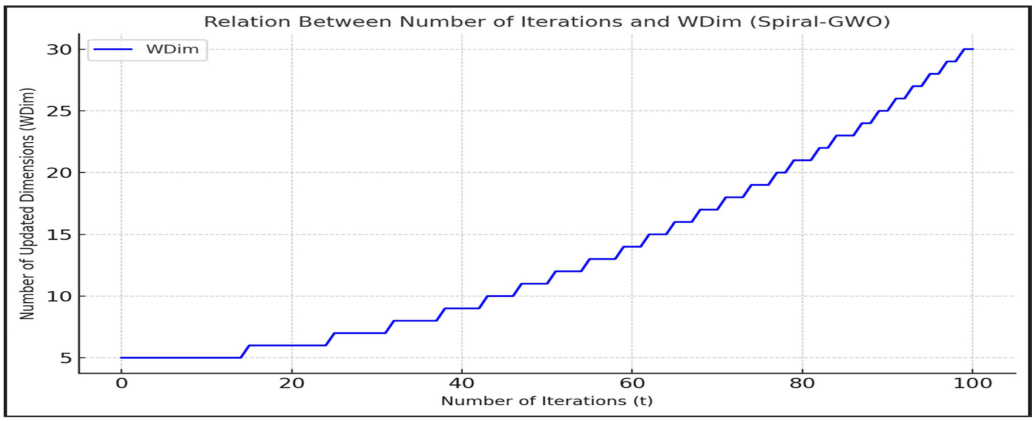

Fig. 4. Example shows the relation between *WDim* and the number of iterations, *Dmax* = 30, *Dmin* = 5, *tmax* = 100.

the early stages to enhance diversity, and later selected dimensions farther from α to focus the search.

B. Update wolf position

Selected dimensions are then updated through the spiral motion (Eq. (2)):

$$\mathbf{X_{new}} = \mathbf{X}_{\alpha} + \mathbf{r}.\mathbf{e}^{(\mathbf{b}.\boldsymbol{\theta})}.\mathbf{cosine}\,(\mathbf{2}\pi\theta) \tag{2}$$

where b (growth factor) is a constant that regulates the tightness of the spiral; $b > 0$ expands (exploration), $b < 0$ contracts (exploitation), r is the distance between the current wolf and the alpha wolf, and $X\alpha$ is the best solution (alpha wolf) at the moment. To enhance stability, the angular component θ is **hybridized** in early iterations; it is randomly sampled from $U(0, 2\pi)$ to promote diversity, while in later iterations, it is scheduled as $\theta = 2\pi(t/T)$ to focus on exploitation.

This mechanism replaces standard enveloping updates during stasis, causing wolves to sweep α in an expanding or contracting vortex, thus exploring near and far regions without losing focus on convergence.

4.6. Scheduling and stopping conditions

S-GWO spirals follow specific scheduling rules to ensure spiral motion is used efficiently, minimizing costs and randomness:

- Activation: Begins only during stagnation.
- Duration: Limited by a maximum number of spiral iterations.
- Termination: Stops immediately if fitness improves.
- Goal stop: Ends when reaching the maximum iterations (T).

4.7. Comparative advantage

Table 1 shows how the S-GWO algorithm outperforms standard GWO, PSO, DE, and GA. Its spiral mechanism improves diversity, reduces stagnation, and balances exploration and exploitation, which is essential for high-dimensional tasks like hate speech detection, where feature sparsity demands careful handling.

Table 1. A comparison of S-GWO with optimization methods.

Algorithm	Search Dynamics	Weakness	S-GWO Advantage
Standard GWO	Leader-based encircling and averaging updates	Stagnates in local optima	Spiral motion dynamically escapes stagnation with adaptive dimensions
PSO	Velocity–position updates	Overshooting optima, weak local refinement	Controlled spiral reduces overshoot and improves local search
DE	Mutation + crossover	Poor diversity in small populations	Spiral preserves diversity with continuous, directed motion
GA	Selection + crossover + mutation	Random search may lack directional guidance	Leader-guided spiral adds deterministic bias toward promising regions

4.8. Integrated workflow of the spiral-grey wolf optimizer algorithm

The S-GWO algorithm operates in complete course ensuring diverse options and monitoring for stagnation. If progress halts, it adjusts to explore new paths. Scheduling rules manage when to start and stop activities, balancing exploration of new solutions with the use of existing ones. Detailed steps are in Algorithm 1.

5. Methodology

This section outlines the main components of the proposed multilingual hate speech detection system.

5.1. Dataset description

This paper used two monolingual hate speech datasets to evaluate the effectiveness of the proposed system across different linguistic contexts and platforms. The first dataset, introduced by [22], includes 47,225 Arabic comments collected from Instagram, covering multiple dialects. The comments were manually annotated into four categories: bullying, toxic, positive, and neutral. The second dataset, described in [23], consists of 24,782 English tweets categorized into three classes: hate speech, offensive language, or neither. The details of both datasets, including the collection, annotation, and distribution procedures, are fully documented in the sources and summarized in Table 2. For system evaluation, both datasets were split into 70% for training and 30% for testing.

5.2. Data preprocessing

Upon selecting suitable datasets for this paper, thorough data preprocessing and cleaning were performed on both Arabic and English corpora. The purpose of this stage is to transform the raw, noisy inputs into structured formats for further machine learning and text classification, a necessary step for achieving better results, as proposed in [24].

The first was a continuous text-cleaning function, which filtered out emojis, hyperlinks, punctuation, and numbers from each language, effectively reducing potential textual noise. In the Arabic version, diacritics were removed to ensure consistency; in the English version, the text was transformed into lowercase, and abbreviated words or expressions were expanded to maintain similarity with those used in previously published sources [25, 26].

Algorithm 1: S-GWO

Input:
Set Max iterations **T**, population size **N**, bounds **[LB, UB]** , Stagnation limit **Smax**, spiral cap $\mathbf{K_{max}}$, Dimension ratios $\mathbf{D_{min}}$, $\mathbf{D_{max}}$, problem dimensionality $\mathbf{D_p}$, Phase boundary ϕ, selection boundary σ, Spiral rates **b_explore (>0)**, **b_exploit (<0)**, Fitness function fitness**(x)**
Output: Best solution **X_best** and its fitness **f _best**
Begin
Step1: Initialization
1.1. Initialize population X_i (i = 1 to N) within bounds
1.2. Evaluate f(X_i); rank and set (X_α, X_β, X_δ); X_best ¬ X_α; f_best ¬ f(X_best)
1.3. Set t ← 0, stagnation_counter ← 0; set GWO parameter a
Step 2: Main loop: while t < T do
2.1. Standard GWO update:
For each X_i update: using GWO rules and applying boundary check
End for
2.2. Evaluate fitness; update X_α, X_β, X_δ
If f(X_α) < f_best: X_best ←X_α; f_best ← f(X_best); stagnation_counter ← 0
Else stagnation_counter ← stagnation_counter + 1
End if
2.3. If stagnation_counter ≥ Smax then
k ← 0; improved ← false
while (k < K_{max}) and (improved = false) do // A. Dimension Selection (Eq. (1))
W_{Dim} = round ([D_{min} + (t/T) ^2*(D_{max} - D_{min})] *D_p)
If t/T < σ, then select W_{Dim} dims randomly
else select W_{Dim} dims with largest |X_i - X_α | // B. Spiral Update (Eq. (2)) — HYBRID θ
End if
If t/T < **Φ** then
$\theta \sim$ U (0, 2 π); b ← b_explore // early phase: random angle, expansion
else
θ ←2 π *(t/T); b← b_exploit // late phase: scheduled angle, contraction
End if
For each $X_i \neq$ X_best:
r = distance (X_i, X_α)
Xnew = X_α + r * exp (b* θ) * cos (2 π θ)
Apply boundary check; evaluate f(Xnew)
If f(Xnew) < f_best:
X_best ← Xnew; f_best ← f(Xnew)
stagnation_counter ← 0; improved ← true
End if
End for
k ← k + 1
End while
End if
2.4. Update parameter a; t ← t + 1
End while
Step3: Return X_best, f_best
End.

With normalization, the characters of over-variation and tabulations were eliminated from Arabic characters. This is done by compressing the Arabic Zones (e.g., 'ـ' to 'ي', 'ة' to 'ه'), writing all hamzas as if they were in (ا), and removing the taweel prolongation mark.

For English, processing included converting abbreviations. Unwrapping phrases (e.g., "don't" to "do not") reduced lexicon fragmentation and enabled better classification.

The problem is that if the length of each line (column-wise) exceeds a certain threshold, horizontal coordination will be necessary to make a meaningful document. Previous researchers [27] and [28] have suggested that this would help to improve the quality of the textual Infinite Sequence, in its marginal status upgrades at least.

Table 2. Summary of arabic and english datasets used.

Reference	Language	Platform	Total Samples	Classes	Class Distribution	Textual and Linguistic Properties
[22]	Arabic	Instagram	47,225	Bullying, Toxic, Positive, Neutral	Bullying: 12,552 Toxic: 5,935 Positive: 17,374 Neutral: 11,364	Highly diverse dialects, informal structure, and semantic ambiguity
[23]	English	Twitter	24,782	Hate Speech, Offensive Language, Neither	Hate: 1,429 Offensive: 19,190 Neither: 4,163	Structurally consistent, standardized spelling, clearer context

The text is then segmented using the word_tokenize algorithm from the NLTK library, resulting in a set of consecutive word units as shown below. This result is in agreement with studies in other countries [29].

Then, stopwords and noisy high-frequency words (e.g., في [in], من [from], على) were eliminated, which add little to semantic understanding. These involved stopwords from standard Arabic, dialectal form, and English, respectively, using custom-generated and curated lists from NLTK. This process reduced computational effort, minimized semantic noise, and highlighted the most valuable terms during the classification process.

Finally, for both web search logs and community-based data, stemming was performed using the Information Science Research Institute (ISRI) stemmer for Arabic language terms and the Porter stemmer for English terms, to reduce words to their root forms while preserving the meaning context. Mainly due to this removal of redundancy, it indirectly enhanced text representation.

All preprocessing techniques were implemented in Python and had an empirical impact on enhancing the quality of feature representation, which could help improve the performance of models for multilingual hate speech detection.

Therefore, preprocessing and purifying were not just considered as technical operations but as orderly procedures closely related to the proposed system, directly facilitating the subsequent feature representation process.

5.3. Enhanced feature representation

As mentioned by [30], feature representation is a fundamental step in converting unstructured texts into digital vectors that machine learning models can process. Studies have shown that this step contributes to achieving more accurate text classification, especially when dealing with diverse and noisy datasets from social media content. In this paper, TF-IDF has been used to calculate the importance of words within texts, based on the definition in [31], and its equation is defined as follows:

$$\mathrm{TF}-\mathrm{IDF}\left(\mathrm{t},\ \mathrm{d}\right)=\mathrm{TF}\left(\mathrm{t},\ \mathrm{d}\right)\times\log\frac{N}{df\ \ (t)} \tag{3}$$

where: N is the total number of documents, df(t) is the number of documents containing term t.

TF-IDF was selected over more complex word embedding techniques such as Word2Vec, FastText, and the Bidirectional Encoder Representations from Transformers (BERT) due to its interpretability, computational efficiency, and effectiveness in handling the variability in Arabic and English social media texts; its inability to capture semantic relationships necessitated the design of an improved thresholding strategy. To address this limitation, S-GWO has been employed, which introduces stagnation detection and dynamic adjustment of the search dimensions, enabling more adaptive exploration and avoiding premature convergence. The enhanced feature representation pipeline consists of four stages:

Stage 1: Document Term Matrix Creation: A Count Vectorizer produces a Document Term Matrix (DTM) by listing unique words and their frequencies from the cleaned text. Global frequencies are then calculated for refinement.

Stage 2: Frequency Optimization: Using global frequency values, S-GWO identifies the optimal frequency threshold by balancing exploration and exploitation within set limits.

Stage 3. Filtering: The specified threshold is applied to refine the vocabulary by removing tokens with low contextual contribution, ensuring that only distinctive and semantically relevant features are retained.

Stage 4. TF-IDF Transformation: Finally, the filtered DTM model is transformed into a TF-IDF matrix using only the selected features. This process ensures that the resulting feature space is both semantically rich and statistically significant, providing efficient inputs for classification models.

This pipeline enhances links between preprocessing and classification, ensuring that features reflect both statistical and contextual significance, a critical aspect when modeling hate speech in Arabic dialects and English texts.

5.4. Machine learning algorithms

As stated by [32], machine learning techniques are crucial to identify hate speech in both organized and unstructured language, especially given the variability of social media language. This paper employs six classifiers based on statistical, probabilistic, and distance-based models to achieve high classification accuracy on multilingual data. Table 3 describes the models used in this paper. All models were implemented using the Scikit-learn library, with some hyperparameters modified to suit the nature of the text data.

6. Proposed system architecture

As illustrated in Fig. 5, the proposed system is a multi-stage system for detecting hate speech across Arabic and English datasets. It begins with a dedicated preprocessing process to normalize raw social media texts, followed by lexical representation using count vectors. Feature refinement is performed via S-GWO, which adaptively selects informational tokens and adjusts the frequency threshold. The resulting features are transformed using TF-IDF and used to train six classifiers (RF, KNN, SVM, LR, NB, SGD), which are evaluated through standard performance metrics. This architecture enhances cross-linguistic generalization, while S-GWO enables feature selection with flexibility and accuracy that exceeds static methods.

7. Experiments

Two experiments were executed to verify the accuracy of the proposed system in detecting hate speech and its generalizability to other languages: one experiment for each

Table 3. Machine learning model.

No.	Machine Learning model	Model type	Description
1	RF	Ensemble, Non-Linear	An ensemble learning technique called RF builds several decision trees during training and uses majority vote to aggregate their predictions. It is applied to problems involving regression and classification. It is effective for handling high-dimensional data and reduces overfitting through random feature selection, bagging, and bagging [33], making it suitable for noisy hate speech data. In this paper, balanced class weights were applied to address the problem of class inequality.
2	LR	Statistical, Linear	LR calculates the likelihood that an input point falls into a specific category. A linear combination of the input features is subjected to the logistic function: $\boldsymbol{logy} = \mathbf{1} / (\mathbf{1} + \mathbf{e}^{\wedge(-(\mathbf{b0} + \mathbf{b1} * \mathbf{x1} + \mathbf{b2} * \mathbf{x2}))})$ (4) where **X1** and **X2** are the features (input value), **b0, b1**, and **b2** are the coefficients, and **e** is the natural logarithm's base. LR is effective for binary and multiclass classification, particularly when features exhibit a linear relationship [34]. In this paper, the model was implemented using the Saga Solver tool to fit large datasets, and 500 iterations were used to ensure stability when dealing with the TF-IDF representation.
3	KNN	Distance-Based, Non-Linear	The KNN algorithm has been used to classify data based on the closest samples in the feature space, due to its simplicity and flexibility in dealing with non-linear data [35]. To improve accuracy, this paper employs a distance-based weighting method, where samples closer to the decision boundary are given greater weight in the decision. The Euclidean distance was used to measure proximity, according to the relationship: $d\ \mathrm{EUC}(x, y) = \sqrt[2]{\sum_{i=1}^{m} (xi - yi)^2}$ (5) Where x_i is the weight of the phrase "i" in document x, y_i is the weight of the term "i" in document y, and m is the number of unique words in the collection of documents. This approach significantly improved classification performance.
4	SVM	Margin-Based, Non-Linear (Kernel)	SVM is a potent classifier that divides data points of various classes using a hyperplane. Maximizing the distance between support vectors is the goal, as the data points nearest to the hyperplane are the focus. It is beneficial for text classification with high-dimensional data [36]. In this paper, the default configuration (Radial Base Function (RBF) kernel) is used, which is very suitable and accurate for handling non-linear boundaries in high-dimensional data.
5	SGD	Linear Gradient-Based	SGD is an optimization method that uses incremental weight updates with each data point to minimize the loss function [37]. It is efficient for large datasets and linear models. **Weight Update Formula:** $\mathbf{wt} + 1 = wt - \alpha \frac{\eta\, \mathrm{L}}{\eta\, Wt}$ (6) Where: w_t is the weight vector, η is the learning rate, and L is the loss function In this paper, an efficient linear classifier is implemented using the SGD classifier module with the hinge loss function to simulate the performance of linear SVM and apply regularization to enhance the model's generalization ability and reduce overfitting.

(Continued)

Table 3. Continued

No.	Machine Learning model	Model type	Description
6	NB	Probabilistic, Semi-Linear	A straightforward probabilistic classifier, the NB method determines a set of probabilities by figuring out the frequencies and value combinations in a given dataset [38]. Based on Bayes' theorem, the Bayes algorithm predicts the likelihood that a given set of features belongs to a particular class. [39], assuming the independence of features. **Bayes' Theorem:** $P(c\|d) = \frac{p(c)*p(d\ \|c)}{P(d)}$ (7) where: P(c|d) is Posterior probability, P(d|c) is the likelihood, P(c) is the prior probability, and P(d) is the evidence. In this paper, the Naïve Multinomial Bayes algorithm was employed because it represents the most suitable formulation for text classification, based on a multinomial distribution that considers the frequency of words in documents. It has been applied directly to text representations using TF-IDF, which is used to determine whether a text belongs to hate speech.

dataset used, with scenarios before and after applying the S-GWO algorithm. The data was split into training and testing sets, using six different classifiers (RF, LR, KNN, SVM, SGD, and NB), and the results were compared using the same metrics to ensure consistency. The following sections discuss the details of the experimental process:

7.1. Experimental setup

Experiments were run on a Lenovo ThinkPad E16 laptop powered by Intel®Core™ i7-1355U, 16GB of RAM, 512GB SSD, 2GB NVIDIA®GeForce MX550 GPU, and running Windows (64-bit). Frames were written using Python 3.6 (64-bit) in the IDLE environment.

7.2. Parameter settings

To enhance feature extraction, the **(S-GWO)** algorithm was used. The key parameters used are shown in Table 4.

The sphere fitness function is used as the evaluation metric due to its simplicity and efficiency in minimizing errors. The goal is to achieve an optimal set of features while maintaining model robustness.

Note: The parameters Dmin and Dmax are defined as ratios in Table 4, but are expressed as absolute values in Fig. 4 for illustrative purposes (e.g., $D_{min} = 5$, $D_{max} = 30$ when $Dp = 50$).

7.3. Evaluation metrics

The next step is to assess the accuracy, precision, recall, and F1 score of the suggested enhanced multiclass hate speech detection system after it has been built and implemented. Fig. 6, a confusion matrix that includes four different prediction categories—True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN)—is used to facilitate this assessment. The equations for the confusion matrix are as used in [40]:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{8}$$

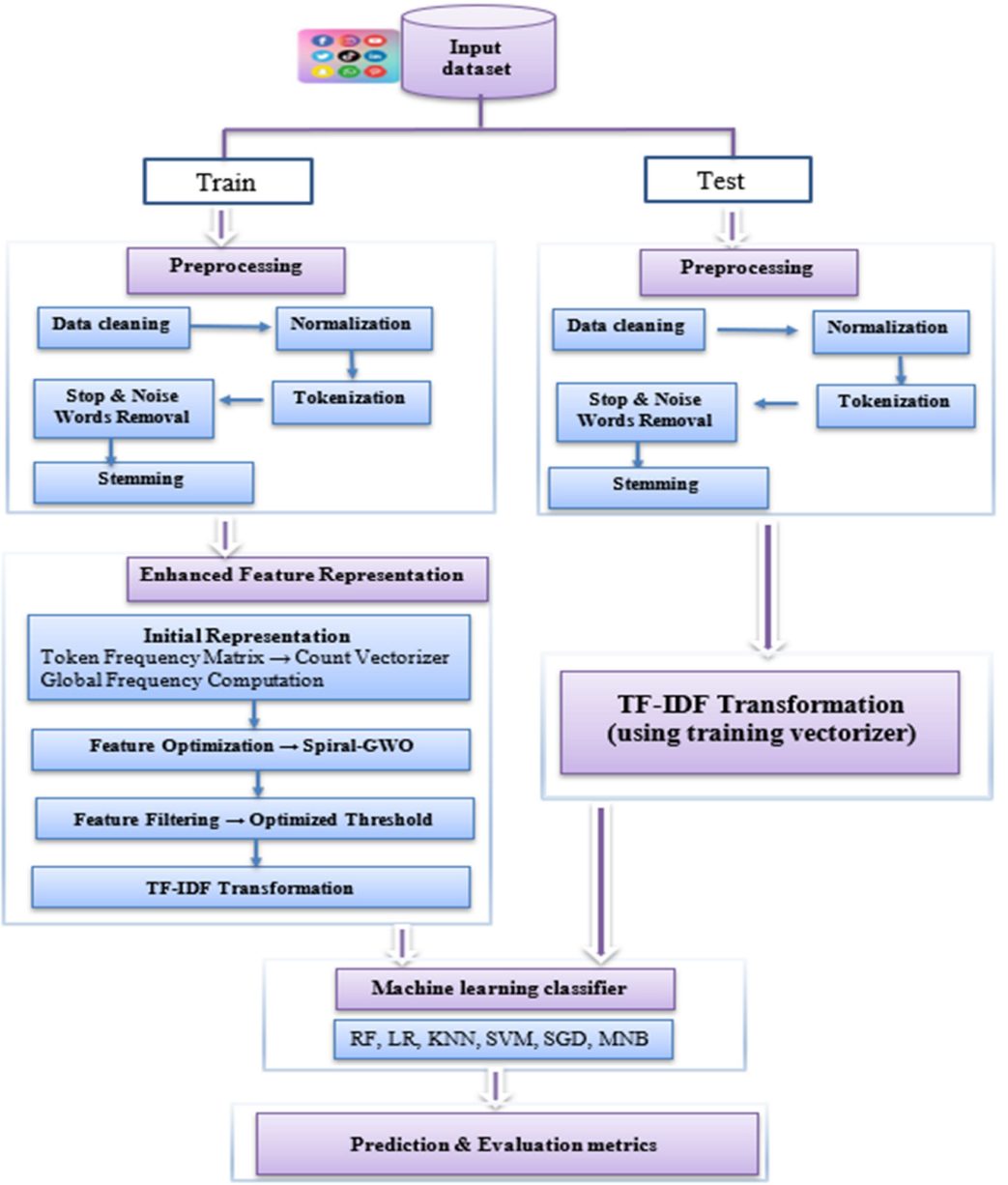

Fig. 5. Proposed system architecture.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{9}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{10}$$

Table 4. Parameter settings of the S-GWO algorithm.

Parameter name	Notation	Type	Constraint within an interval	Description
Lower bound	LB	Constant	[0.0001]	Smallest global frequency values of the feature
Upper bound	UB	Constant	[0.0009]	The largest global frequency values feature
Maximum Iteration	T	Integer	$T = 100$	Total number of optimization iterations.
Stagnation limit	S	Integer	$S = 10$	Maximum allowed consecutive non-improvements before spiral activation.
Initial Population	N	Integer	$N = 30$	Number of candidate solutions at the start.
New Population		Variable	Dynamic	Improved feature thresholds after each iteration.
Evaluation function	f(x)	Function	$F(x)= \sum(x_i{}^2)$	Sphere function to evaluate fitness during optimization
Spiral cap	Kmax	Integer	e.g., 5	Maximum spiral-enabled iterations per activation.
Minimum dimension ratio	Dmin	Float	[0,1]	Minimum ratio of dimensions updated in spiral mode.
Max dimension ratio	Dmax	Float	[0,1]	Maximum ratio of dimensions updated in spiral mode.
Problem dimensionality	Dp	Integer	Dataset-specific	Number of features/variables to optimize.
Phase boundary	ϕ	Float	[0,1]	Threshold to separate early/late search phases.
Selection boundary	σ	Float	[0,1]	Switch from random to guided dimension selection.
Angular component	θ	Variable	$U(0,2\pi)$ or $2\pi(t/T)$	Hybrid angle: random in early iterations, scheduled in later iterations.
Distance to alpha	r	Variable	≥ 0	Euclidean distance between wolf X_i and best wolf X_α.

	Predicated No	Predicated Yes
Actual No	TN	FP
Actual Yes	FN	TP

Fig. 6. Confusion matrix.

$$F1 - \text{Score} = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{11}$$

8. Results

The experimental findings of the suggested hate speech identification system under two experimental conditions are presented in this section.

8.1. Results of the arabic text dataset

Evaluation results of the proposed system for Arabic text captions classified into four categories: bullying, toxic, positive, and neutral. Each classifier was tested in two conditions: before and after applying the proposed SGWO, which was used to improve the quality of feature representation. The classifiers were able to access a broader and more informative

Table 5. Arabic dataset evaluation without the S-GWO algorithm.

Without Optimization												
Model	RF				LR				KNN			
Class	Bullying	Neutral	Positive	Toxic	Bullying	Neutral	Positive	Toxic	Bullying	Neutral	Positive	Toxic
Precision	82%	76%	97%	68%	57%	53%	90%	45%	85%	76%	97%	62%
recall	95%	91%	76%	84%	72%	69%	72%	39%	90%	91%	76%	94%
F1-Score	88%	83%	85%	75%	64%	60%	80%	42%	88%	83%	85%	74%
Without Optimization												
Model	SVM				SGD				NB			
Class	Bullying	Neutral	Positive	Toxic	Bullying	Neutral	Positive	Toxic	Bullying	Neutral	Positive	Toxic
Precision	84%	76%	97%	61%	68%	51%	94%	6%	65%	49%	95%	13%
Recall	92%	90%	76%	94%	65%	68%	66%	56%	65%	68%	67%	52%
F1-Score	88%	82%	85%	74%	66%	58%	78%	11%	65%	57%	78%	21%

Table 6. Arabic dataset evaluation with S-GWO algorithm.

With Optimization												
Model	RF				LR				KNN			
Class	Bullying	Neutral	Positive	Toxic	Bullying	Neutral	Positive	Toxic	Bullying	Neutral	Positive	Toxic
Precision	94%	86%	98%	82%	79%	72%	93%	73%	94%	87%	99%	78%
Recall	99%	97%	86%	90%	88%	86%	82%	64%	98%	97%	85%	95%
F1-Score	96%	91%	92%	86%	83%	78%	88%	68%	96%	91%	92%	86%
With Optimization												
Model	SVM				SGD				NB			
Class	Bullying	Neutral	Positive	Toxic	Bullying	Neutral	Positive	Toxic	Bullying	Neutral	Positive	Toxic
Precision	94%	86%	99%	77%	83%	66%	96%	30%	85%	64%	97%	26%
Recall	97%	96%	85%	95%	77%	82%	74%	82%	73%	83%	77%	82%
F1-Score	96%	91%	92%	85%	80%	73%	84%	44%	79%	72%	86%	40%

collection of input features as a result of the substantial rise in the number of **extracted features from 833 to 8,162.** When working with intricate and dialect-rich Arabic texts, the objective is to assess how feature optimization affects model performance.

As shown in Table 5, the baseline results revealed that RF, KNN, and SVM performed best on the dominant categories (positive and bullying), achieving F1-scores ranging between 85% and 88%. However, the model's performance declined significantly on minority and semantically ambiguous categories, such as toxic and neutral. In particular, SGD and NB recorded the lowest F1-scores in the toxic class, which are 11% and 21%, respectively.

After applying feature optimization, Table 6 shows improved F1-scores for all classifiers and classes. The performance of the toxic class increased to 44% for SGD and 40% for NB. The performance of the neutral class also improved, with LR performance increasing from 60% to 78% and SGD from 58% to 73%. Additionally, improvements in precision and recall were observed across all classes, particularly for the underperforming models.

Tables 7 and 8 present the system performance results with and without S-GWO, demonstrating that RF, KNN, and SVM achieved improvements from 84% to 92% in both precision and weighted F1-score. LR, SGD, and NB also recorded gains of 15%, 9%, and 10%, respectively.

To assess the robustness of the proposed system, the top-performing classifiers (RF, KNN, and SVM) were executed five times on the optimized Arabic dataset. Table Table 9 reports the mean and standard deviation for Accuracy, Precision, Recall, and F1-Score.

Table 7. Proposed system accuracy evaluation without the S-GWO algorithm.

Model	Precision	Recall	F1-Score	Accuracy	Time in milliseconds (ms)
RF	86%	84%	84%	84%	25877
KNN	87%	84%	84%	84%	18
SVM	87%	84%	84%	84%	34541
LR	70%	67%	67%	67%	186
SGD	78%	66%	70%	66%	70
NB	76%	66%	69%	66%	64

Table 8. Proposed system accuracy evaluation with the S-GWO algorithm.

Model	Precision	Recall	F1-Score	Accuracy	Time in milliseconds (ms)
RF	93%	92%	92%	92%	88159
KNN	93%	92%	92%	92%	22
SVM	93%	92%	92%	92%	71688
LR	83%	82%	82%	82%	287
SGD	83%	77%	79%	77%	108
NB	84%	77%	79%	77%	77

Table 9. Statistical Evaluation on Arabic dataset after GWO (Top 3 Classifiers).

Model	Precision (%) ± Std	Recall (%) ± Std	F1 Score (%) ± Std	Accuracy (%) ± Std
RF	92.61 ± 0.01	92.10 ± 0.01	92.08 ± 0.01	92.10 ± 0.01
SVM	92.88 ± 0.00	91.81 ± 0.00	91.78 ± 0.00	91.81 ± 0.00
KNN	92.68 ± 0.00	92.10 ± 0.00	92.05 ± 0.00	92.10 ± 0.00

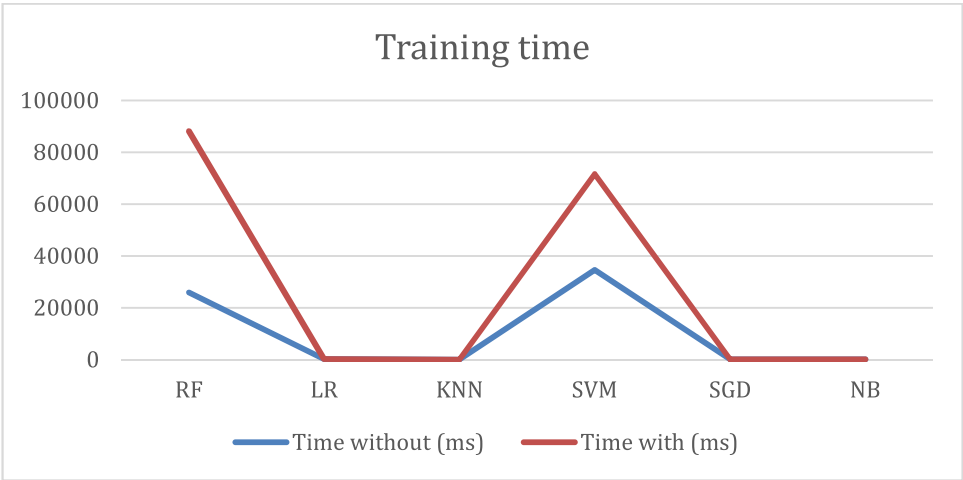

Fig. 7. Training Time comparison of machine learning models of Arabic datasets with and without using the S-GWO algorithm.

The negligible variation (±0.0001) across runs confirms the stability of the results and the effectiveness of the enhanced feature selection method.

Fig. 7 shows changes in execution time, indicating an increase in processing time for complex models such as RF and SVM. Fig. 8 shows the corresponding improvement in accuracy for all classifiers. KNN recorded the shortest running time (22ms) while maintaining the highest accuracy performance.

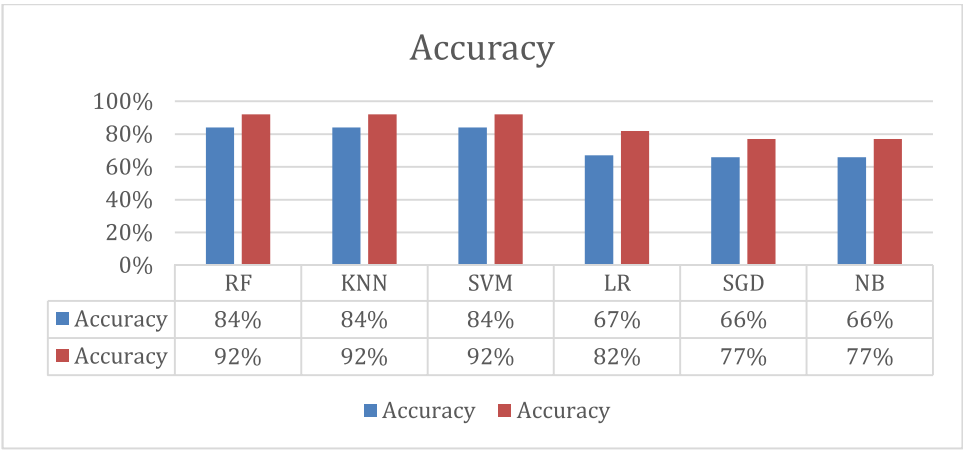

Fig. 8. Accuracy comparison of machine learning models of Arabic datasets with and without using the S-GWO algorithm.

Table 10. English dataset evaluation without the S-GWO algorithm.

Without Optimization									
Model	RF			LR			KNN		
Class	HS	Offensive	Neither	HS	Offensive	neither	HS	Offensive	neither
Precision	100%	99%	100%	83%	85%	93%	97%	100%	99%
Recall	95%	100%	98%	34%	98%	83%	98%	100%	99%
F1-Score	97%	100%	99%	49%	91%	88%	97%	100%	99%
Without Optimization									
Model	SVM			SGD			NB		
Class	HS	Offensive	Neither	HS	Offensive	neither	HS	Offensive	neither
Precision	91%	100%	99%	29%	97%	92%	7%	99%	62%
Recall	98%	99%	98%	78%	94%	88%	87%	87%	92%
F1-Score	94%	99%	99%	43%	96%	90%	13%	93%	74%

8.2. Results of the english text dataset

The enhanced Arabic-based model was tested on the English hate speech dataset by Davidson. To assess the generalizability of the suggested system, it was tested using tweets divided into three groups: hate speech, offensive, and neither. The system was evaluated before and after applying the proposed S-GWO algorithm, which was used to improve the quality of feature representation, using the same six classifiers: RF, LR, KNN, SVM, SGD, and NB. The total number of extracted features also increased significantly, from 1,300 to 9,759, reflecting a more comprehensive representation of the data.

Before optimization, the best-performing classifiers (RF, KNN, and SVM) achieved high F1-scores in all categories, particularly in the abusive and none categories, where scores reached 100%, as shown in Table 10. However, detecting the hate speech category was more challenging for models such as NB (13%) and SGD (43%), due to limited precision and recall. LR also demonstrated average performance in the hate speech category, with an F1-score of 49%.

After optimization (as shown in Table 11), performance improved across all classifiers. Significant improvements were observed in the category of hate speech. LR performance

Table 11. English dataset evaluation with S-GWO algorithm.

With Optimization									
Model	RF			LR			KNN		
Class	HS	Offensive	neither	HS	Offensive	neither	HS	Offensive	neither
Precision	100%	100%	100%	92%	91%	93%	99%	100%	100%
Recall	98%	100%	99%	45%	99%	94%	99%	100%	100%
F1-Score	99%	100%	100%	61%	95%	93%	99%	100%	100%
Model	SVM			SGD			NB		
Class	HS	Offensive	neither	HS	Offensive	neither	HS	Offensive	neither
Precision	98%	100%	100%	63%	99%	96%	8%	100%	66%
Recall	99%	100%	99%	95%	97%	94%	100%	88%	96%
F1-Score	99%	100%	100%	76%	98%	95%	15%	93%	78%

Table 12. Proposed system accuracy evaluation without the S-GWO algorithm.

Model	Precision	Recall	F1-Score	Accuracy	Time in milliseconds (ms)
RF	99%	99%	99%	99%	7188
LR	86%	86%	84%	86%	1637
KNN	99%	99%	99%	99%	0
SVM	99%	99%	99%	98%	9482
SGD	95%	92%	93%	92%	46
NB	94%	87%	90%	87%	0

Table 13. Proposed system accuracy evaluation with the S-GWO algorithm.

Model	Precision	Recall	F1-Score	Accuracy	Time in milliseconds (ms)
RF	100%	100%	100%	100%	29266
LR	92%	92%	91%	92%	2546
KNN	100%	100%	100%	100%	0
SVM	100%	100%	100%	100%	25845
SGD	97%	96%	97%	96%	53
NB	95%	89%	91%	89%	2

increased from 49% to 61%, SGD from 43% to 76%, and NB from 13% to 15%. The top models—RF, KNN, and SVM—achieved full F1-scores across all three categories.

Improvements were also observed in overall metrics, as shown in Table 12 and Table 13. F1-scores and accuracy increased across all models, with RF, KNN, and SVM models achieving 100% accuracy. Meanwhile, LR performance improved from 86% to 92%, and NB from 87% to 89%.

Fig. 9 illustrates the differences in execution times, where post-optimization operations increase the computational cost of models such as RF (from 7,188 ms to 29,266 ms) and SVM (from 9,482 ms to 25,845 ms). However, as shown in Fig. 10, these increases were accompanied by a significant improvement in accuracy. KNN remained the most efficient, achieving 100% accuracy with no measurable training time in both settings.

9. Discussion

This section presents the experimental results, analyzes the performance of the proposed system, and compares its effectiveness with that of previous studies.

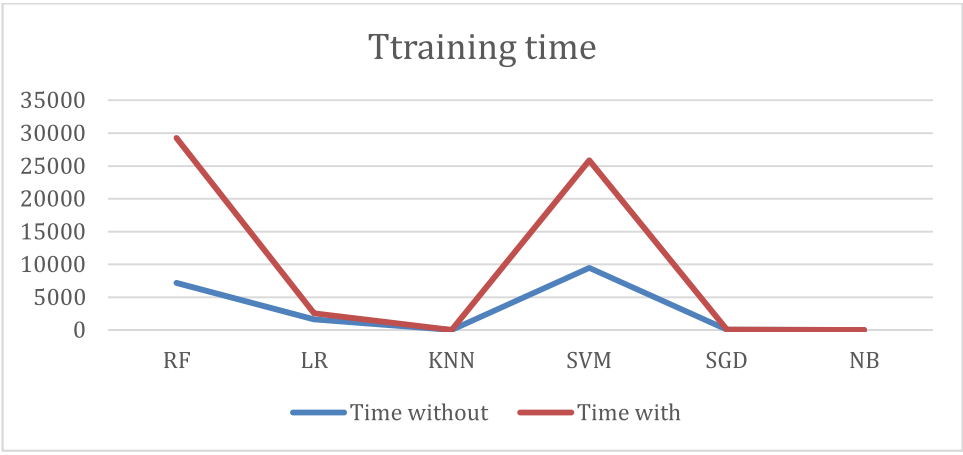

Fig. 9. Training Time comparison of machine learning models of english datasets with and without using the S-GWO algorithm.

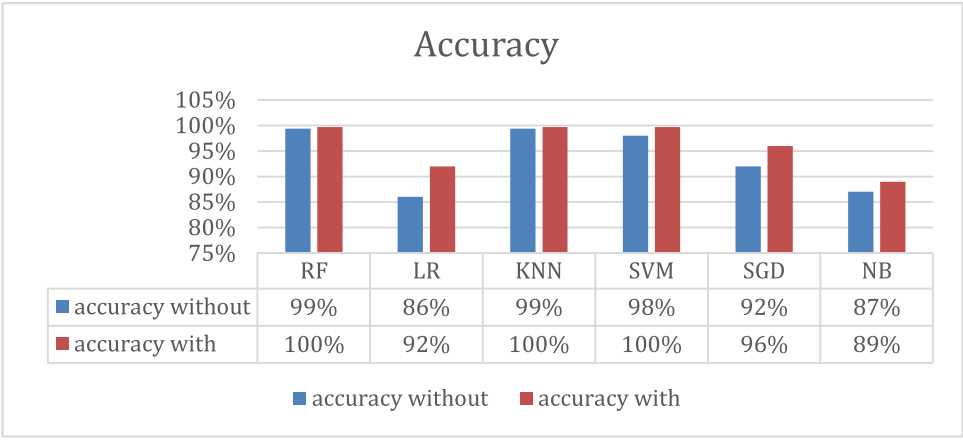

Fig. 10. Accuracy comparison of machine learning models of english datasets with and without using the S-GWO algorithm.

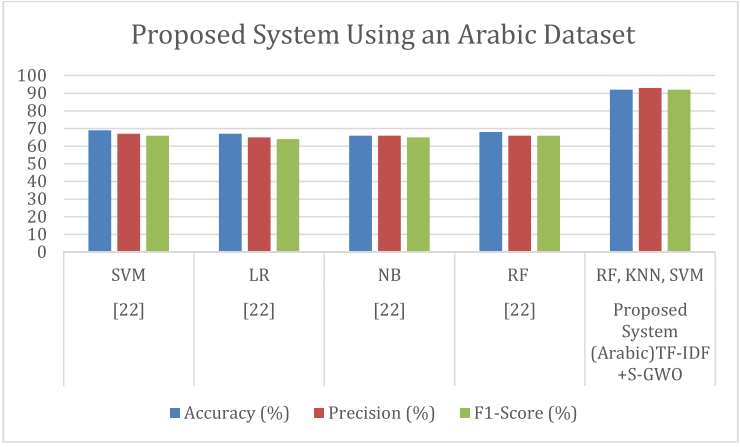

Fig. 11. Proposed system using an arabic dataset.

Table 14. Performance comparison of the proposed system and previous studies using the same datasets.

Reference	Method	Classifier	Accuracy (%)	Precision (%)	F1-Score (%)
[22]	TF-IDF + Preprocessing	SVM	69	67	66
		LR	67	65	64
		NB	66	66	65
		RF	68	66	66
Proposed System (Arabic)	TF-IDF + S-GWO	RF, KNN, SVM	92	93	92
Reference	Method	Classifier	Accuracy (%)	Precision (%)	F1-Score (%)
[23]	TF-IDF + POS + Sentiment	LR	—	91	90
Proposed System (English)	TF-IDF + S-GWO	RF, KNN, SVM	100	100	100

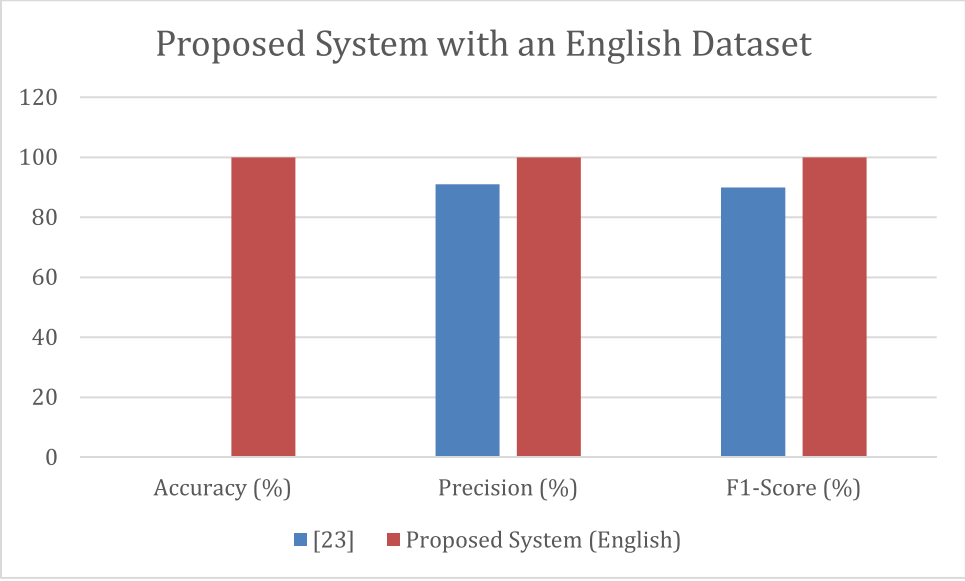

Fig. 12. Proposed system using an english dataset.

9.1. Performance analysis and interpretation

The results clearly demonstrate that it was possible to detect hate speech in both Arabic and English with remarkable accuracy by integrating a meta-heuristic optimization approach (i.e., feature optimization by GWO) with conventional machine learning models. Introducing a spiral motion to the GWO algorithm significantly expanded the feature representation, increasing the number of features from 833 to 8,162 in the Arabic data and from 1,300 to 9,759 in the English data, which directly improved the models' discriminative ability, particularly in underperforming categories such as "toxic" and "neutral" in Arabic and "hate speech" in English. In the Arabic data (Table 5 and Table 6), the F1-score for the toxic category increased from 11% to 44% using SGD and from 21% to 40% using NB, while the performance of LR in the neutral category improved from 60% to 78%. At the overall performance level (Table 7 and Table 8), accuracy and weighted mean F1-score increased from 84% to 92% for classifiers such as RF, KNN, and SVM, with improvements recorded for LR (+15%), SGD (+9%), and NB (+10%). For English data, significant improvements were also achieved in the difficult categories, with the F1-score for the "hate speech" category increasing from 43% to 76% using SGD, from 49% to 61% using LR, and from 13% to 15% using NB. After optimization, the best-performing classifiers

Table 15. Performance comparison of the proposed system and previously discussed studies.

Source	Language	Method	Best Classifier	Accuracy (%)	Precision (%)	F1-Score (%)
[6]	Arabic	Preprocessing + Text normalization + n-gram + RFE + L1 regularization	feature selection using RFE and L1 regularization	84.0	89.0	81.0
[7]	Arabic	Preprocessing + WEKA with two stemmers and Python + SVM	SVM + WEKA with light stemmer	85.49	86.6	86.2
[8]	Arabic	MARBERT + classic machine learning (LR, RF)	LR (For the offensive tweet)	80	78	78
			LR (For the hate speech tweet)	89	72	76
[9]	Arabic, Arabizi, French, English	Preprocessing + FastText Skip-Gram (SG) + Deep Learning (CNN, LSTM, BiLSTM) + Machine Learning	CNN + FasText Skip-Gram embedding	—	87	86
[10]	Arabic	TF-IDF + Oversampling + Machine Learning	NB	AUC = 89	79.90	76.86
[11]	Arabic	Preprocessing + Word Embedding	SVM + Word Embedding	86.3	—	85
[12]	Arabic	wo-stage optimization: (Eq. (1)) fine-tuning Aravec SkipGram embeddings (Eq. (2)) hybrid GA with SVM and XGBoost	SVM with the Aravec Skip Gram word embedding	88.2	87.8	87.8
[13]	Arabic	W2V/TF-IDF + machine learning + Fine-tuned Arabic Transformers	MARBERT (Fine-tuned)	60	62	61
[14]	Arabic	Machine learning-based models, deep learning-based models, learning, and fine-tuning LMs	Task1: AraBERT-base	—	—	Task 1: micro-F1 =84.5
			Task2: AraBERT-large			Task 2: micro-F1 =86.05
[15]	Arabic	Preprocessing + Count Vectorizer and + TF-IDF Vectorizer + Machine Learning	CNB + TF-IDF Vectorize	—	—	77.9
[16]	English	A modified TF-IDF + Machine Learning (RF)	TF-IDF + RF	—	—	—

(Continued)

Table 15. Continued

Source	Language	Method	Best Classifier	Accuracy (%)	Precision (%)	F1-Score (%)
[17]	Arabic	Hybrid feature selection using BGWO + PSO + SCA, with SVM (C = 10), applied on an imbalanced Arabic Hadith text dataset.	BGWO-PSO-SCA + SVM	88.08	—	—
[18]	IoTID20 Non-linguistic (network feature data)	DE + XGBoost	XGBoost	83.72	—	—
Proposed System	Arabic	TF-IDF + S-GWO	RF, KNN, SVM	92	93	92
Proposed System	English	TF-IDF + S-GWO	RF, KNN, SVM	100	100	100

(RF, KNN, and SVM) achieved ideal values of 100% in accuracy and F1-score across all categories. Although the expansion in the number of features led to a rise in training time (Fig. 7 and Fig. 9), particularly for RF and SVM, where the training time more than tripled, this increase was justified by the significant performance improvement (Fig. 8 and Fig. 10). It is worth noting that KNN maintained 100% accuracy in both scenarios with virtually no training time, enhancing its suitability for real-time applications. These results validate the robustness and generalizability of the proposed system, as it effectively handles Arabic texts with morphological complexity and dialect diversity, as well as more structured English texts.

Furthermore, although no explicit data balancing techniques were applied, the system demonstrated strong performance across underrepresented classes. This suggests that the enhanced feature selection method contributed to improving class separability, allowing the classifiers to handle imbalanced distributions effectively without increasing computational cost or compromising data authenticity. Additionally, statistical evaluation based on repeated executions (as shown in Table 10) confirmed the consistency of results, with negligible standard deviations, highlighting the reliability and reproducibility of the proposed system. The results also demonstrate that combining improved feature selection with classical machine learning models represents a practical and effective solution for detecting hate speech in multilingual and content-diverse environments, without the need to resort to complex deep learning models.

9.2. Comparative evaluation with previous studies

To verify the effectiveness of the proposed system, two comparative evaluations were conducted: the first using studies with the same datasets, and the second involving larger-scale works addressing similar tasks.

Table 14 presents a direct comparison between the proposed system and previous studies that used the same Arabic and English datasets. The proposed system achieved 92% accuracy and F1-scores on the dataset using RF, KNN, and SVM classifiers. On the English dataset, it achieved a perfect score of 100% on both metrics using the same classifiers. Fig. 11 and Fig. 12 show the results of the proposed system for the dataset.

To further assess the robustness of the proposed system, a broader comparison was conducted with recent studies that addressed hate speech detection in different languages, both Arabic and English. Table 15 summarizes the performance of these studies. Although some of these studies relied on deep learning techniques or hybrid models, the proposed system achieved higher performance using traditional models enhanced by improved feature representation.

10. Conclusions

This paper makes a significant contribution to the field of hate speech detection by developing a lightweight and efficient system that refines text representation using an S-GWO, eliminating the need for complex deep learning models or frequent recalibration of classifier parameters. The refined representation adapts to each dataset independently, producing consistent, high-quality features that substantially enhanced the performance of multiple traditional classifiers without compromising computational efficiency.

The proposed system achieved outstanding performance on independent Arabic and English datasets, with F1-scores reaching approximately 92% and 100%, respectively, using traditional classifiers such as RF, KNN, and SVM. In particular, the system achieved robust classification even in challenging categories such as “Toxic” and “Neutral” in Arabic and “Hate Speech” in English. These results clearly outperformed previous benchmarks, including the study in [22], which achieved an accuracy of only 69% using the same classifiers and dataset. Furthermore, the KNN model proved to be a practical option, achieving high accuracy with minimal training time, thereby supporting the system’s scalability for real-time or resource-constrained applications. These achievements are backed by a comprehensive analysis of the results and quantitative improvements outlined in the Discussion section, which enhances the reliability of the proposed system.

This paper demonstrates that combining traditional representation techniques with heuristic optimization algorithms represents a practical and effective alternative to complex models, with proven generalization capabilities in multilingual, resource-scarce, or dialect-diverse environments. Looking ahead, the system can be extended toward multimodal analysis, incorporating audio and video data, within an integrated multimedia detection system.

Acknowledgment

None.

Declaration of conflicts of interest

The authors declare that they have no conflict of interest.

Authors’ contributions

The authors confirm their contributions to the paper as follows: study conception, design, data collection, analysis, investigation, writing, and interpretation of results were made by Noor S. Farhan; result review, editing, and final approval of the version to be published were made by Matheel E. Abdulmunim and Hasanen S. Abdullah.

Data availability

The datasets used in this paper are publicly available and are available at the following URL:

https://docs.google.com/spreadsheets/d/1B7_ODBw4zyS0ecZJ1edEb-aiFVrDqWH3/edit?usp$=$sharing&ouid$=$104674471513681692287&rtpof$=$true&sd$=$true.

https://github.com/t-davidson/hate-speech-and-offensive-language/tree/master/data.

References

1. A. I. Gazi, A. Rahaman, F. Rabbi, M. N. Nabi, and M. R. B. S. Senathirajah, "The role of social media in enhancing communication among individuals: Prospects and problems," *Environment and Social Psychology*, vol. 9, no. 11, Nov. 2024, Art. no. 2979, doi: 10.59429/esp.v9i11.2979.
2. N. M. K. Tareen, H. K. Tareen, S. Noreen, and M. Tariq "Hate speech and social media: A systematic review," *Turkish Online Journal of Qualitative Inquiry*, vol. 12, no. 8, pp. 5285–5294, Sept. 2021.
3. M. M. Abdelsamie, S. S. Azab, and H. 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*, vol. 14, no. 1, Art. no. 111, May 2024, doi: 10.1007/s13278-024-01258-1.
4. E. M. Al-Shawakfa, A. M. R. Alsobeh, S. Omari, and A. Shatnawi, "RADAR#: An ensemble approach for radicalization detection in Arabic social media using hybrid deep learning and transformer models," *Information*, vol. 16, no. 7, Art. no. 522, Jun. 2025, doi: 10.3390/info16070522.
5. S. Gite *et al.*, "Textual feature extraction using ant colony optimization for hate speech classification," *Big Data Cogn. Comput.*, vol. 7, no. 1, Art. no. 45, Mar. 2023, doi: 10.3390/bdcc7010045.
6. A. Alakrot, M. Fraifer, and N. S. Nikolov, "Machine learning approach to detection of offensive language in online communication in Arabic," in *Proc. 2021 IEEE 1st Int. Maghreb Meeting of the Conf. on Sciences and Techniques of Automatic Control and Computer Engineering MI-STA*, Tripoli, Libya, pp. 244–249, doi: 10.1109/MI-STA52233.2021.9464402.
7. S. Almutiry and M. Abdel Fattah, "Arabic cyberbullying detection using Arabic sentiment analysis," *Egyptian Journal of Language Engineering*, vol. 8, no. 1, pp. 39–50, 2021, doi: 10.21608/ejle.2021.50240.1017.
8. K. H. Makram *et al.*, "CHILLAX - at Arabic hate speech 2022: A hybrid machine learning and transformers-based model to detect Arabic offensive and hate speech," in *Proc. 5th Workshop on Open-Source Arabic Corpora and Processing Tools with Shared Tasks on Qur'an QA and Fine-Grained Hate Speech Detection*, Marseille, France, 20 Jun. 2022, pp. 194–199.
9. I. Guellil, A. Adeel, F. Azouaou, M. Boubred, Y. Houichi, and A. A. Moumna, "Ara-women-hate: An annotated corpus dedicated to hate speech detection against women in the Arabic community," in *Proc. of the Workshop on Dataset Creation for Lower-Resourced Languages within the 13th Language Resources and Evaluation Conf.*, Marseille, France, Jun. 2022, pp. 68–75.
10. M. E. AlFarah, I. Kamel, Z. Al Aghbari, and D. Mouheb, "Arabic cyberbullying detection from an imbalanced dataset using machine learning," *Soft Computing and its Engineering Applications*, in *Proc. 3rd Int. Conf. on Soft Computing and its Engineering Applications (icSoftComp 2021)*, Anand, India, pp. 397-409, doi: 10.1007/978-3-031-05767-0_31.
11. F. Shannag, B. H. Hammo, and H. Faris, "The design, construction, and evaluation of an annotated Arabic cyberbullying corpus," *Education and Information Technologies*, vol. 27, no. 8, pp. 10977–11023, Apr. 2022, doi: 10.1007/s10639-022-11056-x.
12. F. Shannaq, B. Hammo, H. Faris, and P. A. Castillo-Valdivieso, "Offensive language detection in Arabic social networks using evolutionary-based classifiers learned," *IEEE Access*, vol. 10, pp. 75018–75039, 2022, doi: 10.1109/ACCESS.2022.3190960.
13. A. Ahmad *et al.*, "Hate speech detection in the Arabic language: Corpus design, construction, and evaluation," *Frontiers in Artificial Intelligence*, vol. 7, Feb. 2024, Art. no. 1345445, doi: 10.3389/frai.2024.1345445.
14. S. Alghamdi, Y. Benkhedda, B. Alharbi, and R. Batista-Navarro, "AraTar: A Corpus to support the fine-grained detection of hate speech targets in the Arabic language," in *Proc. 6th Workshop on Open-Source Arabic Corpora and Processing Tools (OSACT) with Shared Tasks on Arabic LLMs Hallucination and Dialect to MSA Machine Translation*, Torino, Italia, May 2024, pp. 1-12.
15. T. Alsubait and D. Alfageh, "Comparison of machine learning techniques for cyberbullying detection on YouTube Arabic comments," *International Journal of Computer Science and Network Security*, vol. 21, no. 1, pp. 1–5, Jan. 2021, doi: 10.22937/IJCSNS.2021.21.1.1.

16. N. T. Sri, G. K., N. Kotha, and H. Kandoori, "Modified TF-IDF with machine learning classifier for hate speech detection on Twitter," *Turkish Journal of Computer and Mathematics* Education, vol. 14, no. 3, Oct. 2023, doi: 10.17762/turcomat.v14i03.14177.
17. M. B. Subkhi, C. Fatichah, and A. Z. Arifin, "Feature selection using hybrid binary grey wolf optimizer for Arabic text classification," *The Journal of Technology and* Science, vol. 33, no. 2, pp. 105–116, Sep. 2022, doi: 10.12962/j20882033.v33i2.13769.
18. S. Bajpai, K. Sharma, and B. K. Chaurasia, "Anomaly detection in IoT networks using differential evolution and XGBoost," in *Proc. of Int. Conf. on Recent Innovations in Computing (ICRIC 2023)*, Jammu, India, pp. 907–921, doi: 10.1007/978-981-97-3442-9_64.
19. J. K. S. Paw, S. Kadhim, and S. Ameen, "An optimized machine learning model by metaheuristic coronavirus optimization algorithm for precise iris recognition," *Advances in Artificial Intelligence and Machine Learning Research*, vol. 5, no. 1, pp. 3389–3408, March 2025.
20. O. Alomari *et al.*, "Hybrid feature selection based on principal component analysis and grey wolf optimizer algorithm for Arabic news article classification," *IEEE Access*, vol. 10, pp. 121816–121830, 2022, doi: 10.1109/access.2022.3222516.
21. H. Li, H. Kang, J. Li, Y. Pang, G. Sun, and S. Liang, "Single-objective and multi-objective mixed-variable grey wolf optimizer for joint feature selection and classifier parameter tuning," *Applied Soft Computing*, vol. 165, Nov. 2024, Art. no. 112121, doi: 10.1016/j.asoc.2024.112121.
22. R. ALBayari and S. Abdallah, 2022, "Instagram-based benchmark dataset for cyberbullying detection in Arabic text," [Online]. Available: https://docs.google.com/spreadsheets/d/1B7_ODBw4zyS0ecZJ1edEbaiFVrDqWH3/edit?usp=sharing&ouid=104674471513681692287&rtpof=true&sd=true.
23. T. Davidson, D. Warmsley, M. Macy, and I. Weber, 2017, "Hate Speech and Offensive Language Dataset," [Online]. Available: https://github.com/t-davidson/hate-speech-and-offensive-language.
24. M. Q. Saadi and B. N. Dhannoon, "Arabic cyberbullying detection using support vector machine with cuckoo search," *Iraqi Journal of Science*, vol. 64, no. 10, pp. 5322–5330, Oct. 2023, doi: 10.24996/ijs.2023.64.10.37.
25. N. Garg and K. Sharma, "Text pre-processing of multilingual for sentiment analysis based on social network data," *International Journal of Electrical and Computer Engineering*, vol. 12, no. 1, pp. 776–784, 2022, doi: 10.11591/ijece.v12i1.pp776-784.
26. M. O. Hegazi, Y. Al-Dossari, A. Al-Yahy, A. Al-Sumari, and A. Hilal, "Preprocessing Arabic text on social media," *Heliyon*, vol. 7, no. 2, Art. no. e06191, Feb. 2021, doi: 0.1016/j.heliyon.2021.e06191.
27. G. Jaradat, M. Shehab, D. Ibrahim, S. Najdawi, and R. Sihwail, "Deep learning approaches for detecting cyberbullying on social media," *Journal of Computational and Cognitive Engineering*, vol. 0, no. 0, pp. 1–15, Mar. 2025, doi: 10.47852/bonviewJCCE52024162.
28. K. M. G. S. Karunarathna and R. A. H. M. Rupasingha, "Learning to use normalization techniques for preprocessing and classification of text documents," *International Journal of Multidisciplinary Studies*, vol. 9, no. 2, pp. 69–81, Jul. 2022.
29. A. Erkan and T. Güngör, "Analysis of deep learning model combinations and tokenization approaches in sentiment classification," *IEEE Access*, vol. 11, pp. 134951–134968, 2023, doi: 10.1109/ACCESS.2023.3337354.
30. B. P. R. Rella, "The role of feature engineering in machine learning: techniques, challenges, and automation with data engineering," *Iconic Research and Engineering Journals*, vol. 8, no. 10, pp. 805–823, April 2025.
31. H. EL-Zayady, M. S. Mohamed, K. M. Badran, and G. I. Salama, "A hybrid approach based on personality traits for hate speech detection in Arabic social media," *Electrical and Computer Engineering*, vol. 13, no. 2, pp. 1979–1988, Apr. 2023, doi: 10.11591/ijece.v13i2.
32. A. M. Alduailaj and A. Belghith, "Detecting Arabic cyberbullying tweets using machine learning," *Mach. Learn. Knowl. Extr.*, vol. 5, no. 1, pp. 29–42, Jan. 2023, doi: 10.3390/make5010003.
33. O. A. M. López, A. M. López, and J. Crossa, *Multivariate Statistical Machine Learning Methods for Genomic Prediction*. Cham, Switzerland: Springer, 2022. doi: 10.1007/978-3-030-89010-0.
34. N. S. Mullah and W. M. N. W. Zainon, "Advances in machine learning algorithms for hate speech detection in social media: A review," *IEEE Access*, vol. 9, pp. 88364–88376, 2021, doi: 10.1109/ACCESS.2021.3089515.
35. A. Jung, *Machine Learning: The Basics*. Singapore: Springer, 2022. doi: 10.1007/978-981-16-8193-6.
36. S. Pathak, A. Pawar, S. Taware, S. Kulkarni, and A. Akkalkot, "A survey on machine learning algorithms for risk-controlled algorithmic trading," *International Journal of Scientific Research in Science and Technology*, vol. 10, no. 3, pp. 1069–1089, May-Jun. 2023, doi: 10.32628/IJSRST523103163.
37. B. Gaye, D. Zhang, and A. Wulamu, "Sentiment classification for employees reviews using regression vector-stochastic gradient descent classifier (RV-SGDC)," *PeerJ Computer Science*, vol. 7, Art. no. e712, Sep. 2021, doi: 10.7717/peerj-cs.712.
38. H. U. Rahman, M. Divya, B. R. Reddy, K. S. Kumar, and P. R. Vani, "Cyberbullying detection using natural language processing," *International Journal for Research in Applied Science & Engineering Technology (IJRASET)*, vol. 10, no. v, pp. 5241–5248, May 2022, doi: 10.22214/ijraset.2022.43683.

39. S. T. A. Al-Latief, A. Ahmad, S. M. Khadim, S. Yussof, and A. Alkhayyat, “WAR strategy algorithm-based hybrid optimization for accurate and rapid speech recognition,” *Iraqi Journal for Computer Science and Mathematics*, vol. 6, no. 1, Mar. 2025, Art. no. 13, doi: 10.52866/2788-7421.1243.
40. D. K. Eddine, Y. Boualleg, K. E. Haouaouchi, “Ensemble of pre-trained language models and data augmentation for hate speech detection from Arabic tweets,” 2024, *arXiv:2407.02448*.