Enhanced Aquila Optimizer Algorithm for Efficient Stance Classification in Online Social Networks

Na LI
Department of Information Technology, ZhengZhou Vocational College of Finance and Taxation,
Zhengzhou, Henan, 450048, China

***Abstract*—Stance classification in Online Social Networks (OSNs) is essential to comprehend users' standpoints on various issues relating to social, political, and commercial aspects. However, traditional methods applied to large datasets and complex text structures usually face several challenges. This study introduces the Enhanced Aquila Optimizer (EAO), a metaheuristic algorithm designed to improve convergence and precision in stance classification tasks. EAO incorporates three new strategies: Opposition-Based Learning (OBL) to improve the exploration, Chaotic Local Search (CLS) to escape from the local optima, and a Restart Strategy (RS) to rejuvenate the search process. Experimental assessments on benchmark OSN datasets prove the superiority of EAO in terms of accuracy, precision, and computational efficiency compared to state-of-the-art methods. These findings position EAO as a potential revolution for stance classification and other large-scale text analysis tasks by offering a robust solution that can be used in real-time for complex OSN scenarios.**

Keywords—Stance classification; online social networks; opposition-based learning; chaotic local search; Aquila Optimizer

I. Introduction

In the past couple of years, the rapid development and growth of Online Social Networks (OSNs) have been driven by the remarkable evolution of our economic and social lives due to the internet [1]. OSNs represent powerful communication platforms for sharing interests, exchanging ideas, or creating communities. Schneider, et al. [2] defined OSNs as networks of individuals with common interests in activities and mutual relationships. Most OSNs enable users to share messages, photos, and videos, rate objects, or discuss any topic, from health problems to political opinions [3]. Another social network is Facebook, estimated to have 2.38 billion monthly registered users as of March 2019, and Twitter has around 330 million monthly active users. These are dynamic and real-time sources of information wherein users actively post their views on certain topics. Thus, OSNs have become a significant means of understanding trends in public opinion and have valuable insights into how ideas originate and spread in complex social networks [4].

The challenges for position classification in OSNs are enormous, considering the immense volume and complexity of information exchanged daily. OSNs generate vast amounts of unstructured text data covering diverse subjects, from sixteen social issues to product reviews, often including various linguistic expressions, slang, abbreviations, and dialects of different regions [5]. This makes developing a model that can generalize for all types of content pretty tough. Moreover, the language used in OSNs is generally informal, with humor, sarcasm, and implicit cues included, making their correct classification challenging. Traditional machine learning techniques, including Neural Networks (NNs) and Support Vector Machines (SVM), often need to scale up efficiently for big dynamic data sets [6, 7]. These methods can only grab those subtle expressions reliant upon context and reduce the accuracy of finding users' positions. More advanced and scalable methods are needed to process the inherent complexity present in OSN data and improve the accuracy of position classification [8].

Metaheuristic optimization algorithms have gained considerable attention due to their efficiency in exploring large-sized search spaces associated with complex classification tasks [9]. Unlike the conventional methods of optimization, which suffer from high dimensionality and run the risk of getting trapped in local optima, metaheuristic algorithms have shown flexibility in navigating through such complex landscapes [10]. All these algorithms are population-based and explore a range of possible solutions at every iteration, with a better chance of converging to a global optimum. Their adaptability predisposes them to be good performers across various scenarios, especially in data-rich dynamic environments like OSNs. Metaheuristic algorithms explore the balance between exploration and exploitation in high-dimensional classification tasks and complex data to reach high accuracy while reducing computational costs [11]. This flexibility and strength of position makes metaheuristic optimization a valuable tool in solving the problems of position classification in OSNs.

Aquila Optimizer (AO) is a newly introduced metaheuristic algorithm. The algorithm is inspired by the hunting strategy of the Aquila bird, one of the best algorithms in balancing exploration and exploitation to find its prey. AO simulates four different phases of the hunting behavior of the Aquila bird for moving between diverse explorations to the search space and focused exploitation around promising solutions. So far, this approach has successfully solved several optimization tasks, positioning AO as one of the more promising choices for challenging high-dimensional problems. Like many other metaheuristics, however, AO has ample limitations. In complex problem spaces, convergence can be slow, and this method may get stuck in suboptimal regions. These all relate to the ability of the algorithm to balance the trade-off between exploration and exploitation, which is the central challenge in the efficient exploration of massive, complex Spatiotemporal data environments characterizing OSNs. This makes enhancements of AO of paramount importance and assures better performance of such robust applications on classification.

The complexity of position classification in OSNs, marked by vast data volumes, high dimensionality, and nuanced language, demands highly efficient optimization methods. Although AO performed so well in optimization performance in many scenarios, the efficiency of the algorithms is crippled when dealing with the scale and intricacies of OSN data. Convergence to optimality can take a long time and sometimes even get stuck in local optima, which will make this algorithm deteriorate more for the performance of large-scale classification tasks where timely and proper analysis is required. Based on these findings, we will represent an Enhanced Aquila Optimizer (EAO) by implementing extra mechanisms into AO to enhance its capability.

Metaheuristic optimization algorithms have drawn much interest since they can handle high dimensions and the inherent complexity of OSN's stance classification tasks. It is still challenging to efficiently map these complicated data structures into user and linguistic patterns. Recent breakthroughs in graph-based embeddings, such as Subgraph2vec, confirmed the power of random walk-based methods for meaningful representation extraction from structured data, including knowledge graphs [12]. Motivated by such techniques, our study focuses on optimizing the exploration and exploitation of high-dimensional data with the EAO.

EAO unifies three central strategies, namely Opposition-Based Learning (OBL), Chaotic Local Search (CLS), and a Restart Strategy (RS). OBL enhances the exploration capability by considering the opposite solutions; hence, it increases the diversity in exploiting the search space for high possibilities of converging to global optima. CLS contributes controlled randomness to the search process, which helps the algorithm maintain its distance from local optima and promotes diversity in potential solution regions. Finally, it uses RS in its last stage to reset the stagnated solution and reinitialize the diverted search agents into suboptimal areas, rejuvenating the exploration process and accelerating the convergence speed.

II. Related Work

Several works have addressed position classification in OSNs using machine learning and optimization techniques. Parimi and Rout [13] envisaged a paradigm based on similarity to spread rival and counter-rumors. They introduced a probabilistic score-based system for determining whether a user should support a rumor or a counter-rumor. This paper uses a neighborhood analysis-based propagation methodology to examine the effects of rumor and counter-rumor cascades in OSNs. Determining the minimum user count that will start the counter-rumor and reducing communication costs in the application is another complex problem this study attempts to tackle.

A comprehensive Hybrid Clustered Shuffled Frog-Leaping Algorithm-Particle Swarm Optimization (HCSFLA-PSO) algorithm was proposed by Hu, et al. [14] to quickly and continuously suppress rumors spread in OSNs. First, a novel scheme for refuting rumors and an inventive depiction of trust levels are presented by dissecting social interactions and examining intimacy, independence, and credibility. Second, a thorough HCSFLA-PSO algorithm is developed, utilizing the PSO algorithm's quick convergence and the SFLA's local clustering to refute rumors. This comprises the CP-HCSFLA-PSO component for real-time rumor refutation during truth evolution and the CNP-HCSFLA-PSO sub-algorithm for timely rumor refutation, adapted to various social relationships with differing levels of trust.

Saeidi [15] provided a straightforward, uncomplicated, and efficient approach to determining trust relationships between different OSN members. Consequently, four novel approaches for assessing the trustworthiness of users are developed and evaluated with the Anderson-Darling statistical hypothesis and Kolmogorov-Smirnov analyses to choose and verify the most suitable model using 20,613 empirical data from 4,552 volunteers in social networks. A metaheuristic algorithm based on the Artificial Bee Colony (ABC) optimization approach was designed to address the temporal complexity of the issue and identify the optimal model fit.

Fatehi, et al. [16] have developed a hybrid model that integrates graph-based and artificial intelligence methodologies to enhance the coverage and precision of online social networks. This method uses a distributed learning automaton rather than established graph-based search methods like breadth-first search, which can identify all reliable associations without limitations. Simulation findings conducted on an actual dataset from Epinions.com show enhanced accuracy and coverage relative to leading algorithms. The suggested approach demonstrates an accuracy of 0.939, indicating a 6% improvement over similar algorithms.

Mallick, et al. [17] devised a collaborative deep-learning algorithm for detecting fake news. The suggested method employs user input to assess the trustworthiness of communications, with message ranking established according to these evaluations. Messages of lower rank are preserved for linguistic analysis to verify their authenticity, whereas highly ranked information is acknowledged as legitimate communications. A Convolutional Neural Network (CNN) transforms user input into rankings inside the deep learning framework. Messages with negative ratings are returned to the system for CNN model training.

Vaghefi, et al. [18] investigated personal disclosure and brand perception on online networks like Foursquare and Twitter. Based on social and hyper-personal information processing theories, relationships between peers, distance, and advertising messages are examined. An integrated dataset reveals that self-disclosure is significantly influenced by checking in with friends and their proximity. Especially when interacting with well-informed peers, individuals prefer to ignore inquiries that reflect poorly on their health.

Bangyal, et al. [19] investigated sentiment analyses and the detection of fake news using machine learning and deep learning in different challenges, and significant data volumes arose and became dynamic over Twitter. They presented for the first time an innovative proposed method of detecting COVID-19-related false information in deep learning models, particularly the BiGRU model, for which the obtained accuracy was scored very impressively at around 91%. Bangyal, et al.

[20] compared classical machine learning classifiers such as Support Vector Machines and Random Forests. The results proved efficient in the competition and improved the robustness of even simpler models. Bangyal, et al. [21] extended the work by including semantic models and the TF-IDF and compared eight machine-learning classifiers with four deep-learning classifiers. The valuable outcome provided a comparison between accuracy and computational cost. While an advance in sentiment analysis, these works pointed to the need for scalable algorithms of high precision that handle stance classification in large and complex datasets. This motivated the development of the EAO.

Position classification in OSNs is an attractive topic since there has been a high demand from users to make effective use of people's opinions and to control information dissemination effectively. Various state-of-the-art machine learning and optimization techniques have been studied for the works shown in Table I regarding rumor suppression, trust evaluation, sentiment analysis, and fake news detection. These works span classical machine learning model applications, leveraging the latest metaheuristics algorithms and deep learning frameworks, each with different advantages and limitations. Despite such advances, most previous approaches also either suffer from issues like scalability and computation efficiency or are very poorly applicable to OSN data, which by default is complex and time-variant. This extends earlier work and bridges previous gaps in the literature that resulted in the development of the proposed EAO, providing for a state-of-the-art scalable and efficient stance classifier.

TABLE I. AN OVERVIEW OF PREVIOUS STUDIES

Reference	Methodology/algorithm	Dataset	Key findings	Limitations
[13]	Neighborhood analysis-based propagation and multi-objective genetic algorithm	Actual datasets for rumor and counter-rumor analysis	Efficiently reduced communication costs and minimized counter-rumor users	Limited scalability for larger datasets
[14]	Hybrid clustered shuffled frog-leaping algorithm-particle swarm optimization	Simulated datasets for real-time and evolving rumor refutation	Effectively suppressed rumors, accounting for trust and timeliness	High computational complexity for real-time adaptation
[15]	Artificial bee colony optimization	Facebook dataset (20,613 data points from 4,552 users)	Identified optimal trust models with reduced temporal complexity	Requires significant computational resources
[16]	A graph-based and distributed learning automaton	Epinions.com dataset	Enhanced accuracy and coverage by 6% and 10%, respectively	Does not address dynamic changes in network topology
[17]	Collaborative deep learning with CNN	Simulated datasets for fake news detection	Achieved 98% accuracy in detecting fake news	Dependent on user input for initial rankings
[18]	Integrated social and hyper-personal information processing theories	Integrated dataset (Foursquare and Twitter)	Revealed self-disclosure influenced by peer proximity and advertising	Limited to health-related and brand perception studies
[19]	BiGRU deep learning model	COVID-19-related false information dataset	Achieved 91% accuracy in detecting COVID-19-related fake news	Focused only on COVID-19-related data
[20]	Classical machine learning classifiers (SVM, Random Forest)	Twitter datasets	Proved efficient for simpler models in sentiment analysis	Simpler models are less effective for nuanced stance classification
[21]	Semantic models with TF-IDF, multiple classifiers	Twitter datasets with semantic enrichment	Provided a trade-off comparison between accuracy and computational cost	Needs scalable algorithms for complex stance classification tasks

III. PROPOSED METHOD

A. Problem Definition

Position classification is a task aimed at estimating the stance of users concerning particular topics in OSNs, including opinion classification as favourable, unfavourable, or neutral towards various entities or topics. Unlike general sentiment analysis, which broadly adjectives text into positive, negative, or neutral areas, position classification is a closer sentiment analysis. It pinpoints users' opinions concerning clearly defined subjects, such as political figures, social problems, products, or happenings, offering a better perspective on public opinion. This fact renders this task especially hard due to the dynamism of the OSN data, which are often unstructured, different in linguistic styles, and very massive.

One of the big questions is how to classify user opinions efficiently and accurately from a large dataset with high precision. OSNs generate magnificent volumes of textual data daily, from explicit endorsement to subtle criticism or neutral observations. Such data is usually unstructured and informal, including colloquialisms, abbreviations, and context-dependent meanings, which make their correct classification challenging. Moreover, the data in OSN is typically full of mixed emotions, sarcasm, and idiomatic expressions, complicating their classification.

This vast amount of data is bound together with complexity, and traditional machine learning with statistical methods faces scalability issues while handling this immense data. While the dataset grows, conventional models like support vector machines lose much of their accuracy and efficiency, or even simple neural networks take up much computational time and resources for training and inference. Moreover, these models must be revised to represent complicated relationships and subtleties specific to the context in the OSN language and limit generalization across a wide range of topics with users' expressions.

Metaheuristic optimization is one of the ways of taming this. Considering position classification as an optimization problem, metaheuristic algorithms will handle high-dimensional feature spaces to arrive at the uppermost solution, a highly performant classification with limited computational overhead. Within such a paradigm, the optimization algorithms must be solid and flexible enough to meet the peculiar demands of the data in OSNs. In this regard, it may likely involve the development of classical algorithms, such as Aquila Optimizer, for better exploration, adaptability, and higher accuracy regarding OSN position classification.

B. Enhanced Aquila Optimizer

AO is a newly formulated population-based optimization algorithm inspired by Aquila birds' predatory strategies proposed by Abualigah, et al. [22]. Aquila birds originate from the Northern Hemisphere and are considered one of the most famous predators, distinguished by their agility, strong talons, and strong feet. Thus, they may catch a wide range of prey, from squirrels and rabbits to marmots and hares. The proposed AO algorithm draws inspiration from four different foraging behaviors of Aquila birds that oscillate between exploration and exploitation during hunting. The AO algorithm starts with a randomly generated population of candidate solutions that can be mathematically expressed as follows:

$$X = \begin{bmatrix} X_{1,1} & X_{1,2} & \cdots & X_{1,j} & \cdots & X_{1,D} \\ X_{2,1} & X_{2,2} & \cdots & X_{2,j} & \cdots & X_{2,D} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ X_{n-1,1} & X_{n-1,2} & \cdots & X_{n-1,j} & \cdots & X_{n-1,D} \\ X_{n,1} & X_{n,2} & \cdots & X_{n,j} & \cdots & X_{n,D} \end{bmatrix} \quad (1)$$

Each element $X_{i,j}$ in this matrix represents the position of an agent, calculated as follows:

$$X_{i,j} = rand \times (UB_j - LB_j) + LB_j, \;\; i = 1, \dots, n, \;\; j = 1, \dots, D \quad (2)$$

Where *rand* represents a random number between 0 and 1, UB_j and LB_j are the upper and lower bounds for each dimension, n denotes the population size, and D is the number of decision variables. The AO algorithm consists of four unique stages, facilitating the balance between exploration and exploitation. These stages are triggered under the following conditions:

$$\begin{cases} Perform\ exploration, & if\ t \le \frac{2}{3} \times T \\ Perform\ exploitation, & otherwise \end{cases} \quad (3)$$

Where T refers to the total number of iterations and t is the current iteration.

1) Expanded exploration: At this stage, Aquila searches over a large area to find its prey by performing a high dive and then a soaring flight. In this phase, the position of each agent is updated using Eq. (4).

$$X_1(t+1) = X_{best}(t) \times \left(1 - \frac{t}{T}\right) + (X_M(t) - X_{best}(t) \times rand) \quad (4)$$

Xbest represents the best position, and XM(t) gives the average position of the current population generation.

2) Narrowed exploration: This process is Aquila's most commonly adopted hunting strategy. It involves a short gliding flight and contour-following maneuver. Eq. (5) updates the agent's position.

$$X_2(t+1) = X_{best}(t) \times Levy(D) + X_R(t) + (\gamma - x) \times rand \quad (5)$$

Where Levy(D) is a Levy flight distribution, D represents the number of dimensions, and XR indicates an agent's position at random. The Levy flight is defined by Eq. (6).

$$Levy(D) = s \times \frac{u \times \sigma}{|v|^{1/\beta}} \quad (6)$$

Where β=1.5, u, and v are random values, and s=0.01 is a scaling factor.

3) Expanded exploitation: In the third stage, the search scope is narrowed further; the agent is prepared for an attack through a low-flight preliminary assault. The position update is done as follows:

$$X_3(t+1) = \left(X_{best}(t) - X_M(t)\right) \times \alpha - rand + \left((UB - LB) \times rand + LB\right) \times \delta \quad (7)$$

Where α and δ are exploitation parameters set to 0.1.

4) Narrowed exploitation: In this final stage, Aquila chases and attacks the prey using an escape trajectory. Eq. (8) updates the agent's position.

$$X_t(t+1) = QF \times X_{best}(t) - (G_1 \times X(t) \times rand) - G_2 \times Levy(D) + rand \times G_1 \quad (8)$$

Here, QF represents the quality factor, where $QF(t) = \frac{t}{2 \times rand - 1} / (1-r)^2$, and $G_1 = 2 \times rand - 1$ and $G_2 = 2 \times \left(1 - \frac{t}{T}\right)$ manage different motion and attack angle aspects.

C. Opposition-based Learning

The OBL strategy was first proposed by Tizhoosh [23], and since then, it has been implemented for several swarm optimization algorithms to improve their performance significantly. Many researchers have practiced this technique on swarm optimization algorithms to enhance the exploration and convergence potential of the swarm optimization algorithm. For instance, OBL is combined with the SSA framework in [24] to avoid local optimization problems. In [25], the Harris Hawks Optimization (HHO) algorithm combines the OBL concept with a chaotic local search to substantially improve its exploration capability. Zhang, et al. [26] used OBL to enhance the algorithm's performance in arithmetic optimization. The OBL works based on comparing the original solution's fitness value with its opposition. The opposition solution of an integer x in the bounds [lb, ub] can be computed using Eq. (9).

$$\bar{x} = ub + lb - x \tag{9}$$

For a vector x, the opposite value of each component can be determined as follows.

$$\overline{x_j} = ub_j + lb_j - x_j \tag{10}$$

Where lb_j and ub_j represent the lower and upper bounds for the j^{th} dimension.

D. Chaotic Local Search

CLS is one of the well-known algorithms applied to various swarm optimization techniques, such as the Jaya Algorithm [27], brainstorm optimization [28], and WOA [29]. The CLS approach is usually implemented by using a logistic map in the following:

$$o^{s+1} = Co^s(1 - o^s) \tag{11}$$

Where s denotes the iteration and C is a control parameter, typically set to 4. Values of o^1 can be initialized at 0.25, 0.50, or 0.75. CLS focuses on local searches around the optimal solution found so far, aiming to enhance accuracy within that neighborhood. The CLS-generated values C_s in iteration i are computed as follows:

$$C_s = (1 - \mu) \times T + \mu\hat{C}_i, \;\; i = 1,2,\dots,n \tag{12}$$

Where $\hat{C}_i$ is calculated as:

$$\hat{C}_i = LB + C_i \times (UB - LB) \tag{13}$$

Here, μ represents a shrinking factor, determined by:

$$\mu = \frac{T-t+1}{T} \tag{14}$$

E. Restart Strategy

In the optimization process, some agents will be entrapped or stuck in a particular local optimal and fail to obtain the best performance. Such agents cannot contribute to improving the search but consume additional computing resources. Zhang, et al. [30] proposed RS that restarts or relocates such stagnant agents. The RS strategy traces the improvement frequency of each agent. If an agent does not find newer and better solutions, a trial will increase in value. When it reaches a certain threshold predefined, the position of the agent resets according to the following equations:

$$X(t+1) = lb + \text{rand} \cdot (ub - lb) \tag{15}$$

$$X(t+1) = \text{rand} \cdot (ub + lb) - X(t) \tag{16}$$

F. Proposed Algorithm

EAO, the improvisation proposed in this paper, tries to overcome its bottleneck with the help of standard AO by applying strategies like OBL, RS, and CLS. OBL applied during initialization and in the position updates ensures that the optimizer is initialized with a robust set of agents and explores a good amount of solution space. CLS fine-tunes the best solution in every iteration to ensure an enhanced search precision neighborhood. Finally, RS re-positions the stagnant agents, which stirs the exploration process. Fig. 1 shows the pseudocode of EAO. To obtain the computational complexity of EAO, one can focus on the three phases separately: initialization, assessment, and position update. This gives, in total, the complexity *O(EAO)*:

$$O(EAO) = O(\text{Initialization}) + O(\text{Assessment}) + O(\text{Position update}) + O(\text{CLS} + \text{OBL} + \text{RS}) \tag{17}$$

Assuming T as the total iterations, N as the population size, and D as the number of dimensions, each component has the following complexity:

- Initialization: *O(N)*
- Assessment: $O(N \times T)$
- Position update: $O(N \times D \times T)$
- RS and OBL: $O(N \times D \times T)$
- CLS: $O(N \times T)$

The overall complexity of the EAO can be expressed as follows:

$$O(EAO) = O(N) + O(N \times T) + O(N \times D \times T) + O(N \times D \times T) = O(N \times D \times T) \tag{18}$$

```
Initialize the population matrix X for the AO
Calculate the opposition-based set X̄ and select the top N solutions from X ∪ X̄
Set the initial parameters for the AO
while (iteration t < maximum iterations T) do
    Evaluate the objective function for each solution
    Identify the best agent X_best
    for each agent i from 1 to N do
        Update the mean position of the current solution set
        Compute the parameters y, x, G1, G2, and Levy(D)
        if current iteration t ≤ 2/3T then
            if (random value ≤ 0.5) then
                Update the current position using Eq. 4
                Compute the opposite position using Eq. 10
            else
                Update the current position using Eq. 5
                Compute the opposite position using Eq. 10
        else
            if (Fitness value rand ≤ 0.5) then
                Update the current position using Eq. 7
                Compute the opposite position using Eq. 10
            else
                Update the current position using Eq. 8
                Compute the opposite position using Eq. 10
        end if
    end for
Apply the restart strategy using Eqs. 15 and 16
Apply CLS strategy using Eq. 12
```

Fig. 1. Enhanced Aquila Optimizer.

IV. Performance Evaluation

Stance detection involves an automated method for determining how a writer expresses support, opposition, or neutrality toward a particular argument or topic. The scope of analysis in this field is extensive and may include subjects such as individuals, organizations, governmental policies, social movements, or commercial products. For instance, a detailed examination of Barack Obama's speeches could be conducted to ascertain his position on regulating guns in the U.S. Individuals convey their opinions on various issues through various platforms, including Facebook, YouTube, Twitter, and online forums. This approach applies to several areas, such as stand-alone stance classification, information retrieval, automatic text summarization, and text inference. Over the past decade, significant research has focused on modeling stances in digital media. This study utilized four datasets from an existing database, comprising training and testing data derived from tweets. These datasets cover over two million stance expressions across four topics.

In this work, stance classification is analyzed as an optimization problem. The dataset is initially converted into structured data for analysis. Such datasets used to perform experimentation include Hillary Clinton, the Legalization of Abortion, Atheism, and the Feminist Movement, which constitute an optimization space. Firstly, documents are generated for each dataset after the preprocessing steps. Since the number of tweets is taken across rows in the document array, the total volume of words obtained by preprocessing defines the dimensionality of a single tweet. If a word from a set is featured in a given tweet, its column will be assigned a value equal to 1. In other cases, when some word from a set does not appear in the text of a certain tweet, the respective column will be recorded with 0. Consequently, the document matrix is only composed of zeros and ones.

A vector of word weights is also constructed based on word weights within documents. The "maximum word passes" denotes the highest quantity of records where a single word occurs. The weight for each word is then calculated as the ratio of the word's frequency to the maximum occurrence across all documents. These calculated weights collectively form the word weight vector used in the optimization analysis.

In this research, the similarities among the population participants (potential solutions) and the document structure are critical for constructing an effective classification system. Specifically, correlations between document matrix components and possible solutions are carefully evaluated. Various methods exist for measuring similarity, including Overlap, Dice, Jaccard, and Cosine. This study found that the Jaccard similarity measure yielded the best results, and a modified Jaccard similarity measure quantified correlations across texts. This approach determines standard features by calculating the proportion of shared features over the entire text.

Mathematically, for an individual $X_i = (X_{i1}, X_{i2}, \ldots, X_{iM})$ in the population and a corresponding line $D_j = (D_{j1}, D_{j2}, \ldots, D_{jM})$ Jaccard similarity is calculated as follows:

$$J(D_j, X_i) = \frac{X_i \cap D_j}{X_i \cup D_j} \tag{19}$$

The extended Jaccard similarity is computed using Eq. (20).

$$J(D_j, X_i) = \frac{\sum_{k=1}^{M} D_{jk} X_{ik}}{\sum_{k=1}^{M} (D_{jk}^2) + \sum_{k=1}^{M} (X_{ik}^2) - \sum_{k=1}^{M} D_{jk} X_{ik}} \tag{20}$$

Furthermore, to account for the significance of word frequency, tweet word counts relative to the total document word count were included. This ratio is calculated using Eq. (21).

$$ratio_j = \frac{No.of\ words\ in\ the\ j^{th}\ comment}{Total\ No.of\ words\ in\ the\ document} \tag{21}$$

This ratio helps assign appropriate word weights based on their occurrence within the preprocessed dataset. The new similarity measure, incorporating both the Jaccard similarity and the word frequency ratio, is defined as:

$$Similarity_{ij} = \alpha \times jaccard + \beta \times ratio_j \tag{22}$$

Where α and β are coefficients whose sum equals one, allowing for a balanced weighting scheme to optimize the model's performance.

The similarity of each X_i within the population was evaluated based on all tweets included in the dataset. Classification of tweets was conducted by comparing the similarity value to a predefined threshold: tweets were classified as either above or below this threshold. To determine an individual's fitness in similarity analysis, Eq. (23) was used:

$$F = \alpha \times \frac{TP}{length} + \beta \times \frac{TP}{FP+TP} + \gamma \times \frac{TN}{FP+TN} + \omega \times \frac{TP}{FN+TP} \tag{23}$$

Where False Negative (FN) refers to the number of instances in which the rule incorrectly identified as negative when they belong to the positive class, True Negative (TN) stands for the number of cases accurately recognized as negative by the rule, False Positive (FP) signifies the count of instances incorrectly labeled as positive by the rule, even though they belong to the negative class, and True Positive (TP) represents the number of cases correctly identified by the rule as belonging to the positive class.

The coefficients $\alpha, \beta, \gamma, and\ \omega$ are weight values that must collectively sum to one. These weights are customizable and can be adjusted to optimize the performance of the fitness function, allowing the algorithm to be tailored for the best possible results in a given context.

The proposed EAO algorithm's effectiveness in addressing the stance detection problem, a complex task in online social network analysis, was evaluated and compared against several classifiers. The experiments were performed using MATLAB R2018b on a system equipped with an Intel Core i5-12400F CPU running at 2.5 GHz and 8 GB of RAM. The comparative analysis of the algorithms' results was based on the following classification metrics.

$$F - measure = \frac{2TP}{FN+FP+2TP} \tag{24}$$

$$Recall = \frac{TP}{FN+TP} \tag{25}$$

$$Precision = \frac{TP}{FP+TP} \tag{26}$$

$$Correctly\ labaled\ data = \frac{TN+TP}{FN+TN+FP+TP} \tag{27}$$

Tables II to V present a comparative analysis of EAO, ACO, and AO algorithms applied to stance detection datasets on various social issues. Each table illustrates the performance of multiple classification algorithms, focusing on F-measure, recall, precision, accuracy, and correctly labeled data metrics. For the Feminist Movement dataset (Table II), EAO performs better than others, achieving an accuracy rate of 61.387%, while NaiveBayes and staking also show competitive results. Similarly, in the Atheism dataset (Table III), EAO achieves the highest accuracy of 64.593%, followed closely by random forest.

In the analysis of the Legalization of Abortion dataset (Table IV), EAO emerges as the top-performing classification algorithm with an impressive accuracy rate of 70.201%, significantly higher than the other contenders. Finally, for the Hillary Clinton dataset (Table V), EAO shows better accuracy at 83.921%, indicating enhanced performance than the other algorithms.

TABLE II. RESULTS FOR THE FEMINIST MOVEMENT DATASET

Algorithm	F-measure	Recall	Precision	Accuracy (%)	Correctly labeled data
EAO	0.574	0.598	0.613	61.387	173
NaiveBayes	0.561	0.552	0.601	55.112	158
Stacking	0.522	0.563	0.521	56.845	164
QDA	0.522	0.563	0.521	56.845	163
REPTree	0.518	0.569	0.548	56.849	163
Random forest	0.512	0.517	0.505	51.586	148
Random tree	0.509	0.514	0.502	51.581	147
Extra tree	0.459	0.462	0.475	45.619	129

TABLE III. RESULTS FOR THE ATHEISM DATASET

Algorithm	F-measure	Recall	Precision	Accuracy (%)	Correctly labeled data
EAO	0.621	0.578	0.654	64.593	143
Random forest	0.619	0.628	0.623	62.784	139
Extra tree	0.559	0.571	0.569	57.705	127
Random tree	0.588	0.592	0.586	59.084	131
NaiveBayes	0.556	0.617	0.648	61.375	136
QDA	0.506	0.578	0.584	57.714	128
REPTree	0.501	0.528	0.577	52.716	117
Stacking	0.501	0.528	0.577	52.716	117

TABLE IV. RESULTS FOR THE LEGALIZATION OF ABORTION DATASET

Algorithm	F-measure	Recall	Precision	Accuracy (%)	Correctly labeled data
EAO	0.691	0.686	0.695	70.201	197
NaiveBayes	0.676	0.672	0.688	67.135	187
Random forest	0.666	0.678	0.682	67.845	191
Extra tree	0.643	0.651	0.647	65.342	182
Stacking	0.599	0.625	0.611	62.742	175
Random tree	0.597	0.601	0.605	60.254	168
QDA	0.578	0.635	0.578	63.572	177
REPTree	0.576	0.572	0.641	57.098	159

TABLE V. RESULTS FOR THE HILLARY CLINTON DATASET

Algorithm	F-measure	Recall	Precision	Accuracy (%)	Correctly labeled data
EAO	0.811	0.853	0.856	83.921	248
NaiveBayes	0.813	0.821	0.808	82.372	242
Random forest	0.801	0.818	0.799	82.371	242
REPTree	0.783	0.835	0.822	83.715	243
Extra tree	0.779	0.782	0.776	78.306	230
Random tree	0.727	0.733	0.721	73.546	216
QDA	0.632	0.677	0.719	67.786	199
Stacking	0.628	0.587	0.702	58.306	173

V. CONCLUSION

This paper proposed and evaluated EAO on the OSN stance detection problem, a complex high-dimensional classification problem. The performance of EAO was rigorously compared to ACO and classical AO on multiple datasets about social issues. In this regard, a comparative analysis was conducted, where the EAO outperforms its competitors on the classification accuracy of F-measure, recall, and precision for most of them, thus proving the more remarkable ability of EAO in exploring the search space in most cases. Our results suggest that embedding superior methodologies such as OBL and CLS contributes to robustness and efficiency while handling complex problems with EAO. EAO effectively resolves the challenges posed by stance detection in OSNs through improvements in convergence speed while sustaining diversity among solutions. This work may be extended by further optimization or hybridization strategies, including adaptation of EAO for other tasks related to social media analysis and deep learning frameworks for better performance. The proposed algorithm EAO has considerable potential for wild applications in dynamic and data-heavy settings, serving as a valid tool for large-scale social network data analysis.

REFERENCES

[1] H. Zhihong and L. Tao, "Presenting a Novel Method for Identifying Communities in Social Networks Based on the Clustering Coefficient," International Journal of Advanced Computer Science and Applications, vol. 14, no. 8, 2023.

[2] F. Schneider, A. Feldmann, B. Krishnamurthy, and W. Willinger, "Understanding online social network usage from a network perspective," in Proceedings of the 9th ACM SIGCOMM Conference on Internet Measurement, 2009, pp. 35-48.

[3] M. Bhattacharya, S. Roy, S. Chattopadhyay, A. K. Das, and S. Shetty, "A comprehensive survey on online social networks security and privacy issues: Threats, machine learning-based solutions, and open challenges," Security and Privacy, vol. 6, no. 1, p. e275, 2023.

[4] M. R. Kondamudi, S. R. Sahoo, L. Chouhan, and N. Yadav, "A comprehensive survey of fake news in social networks: Attributes, features, and detection approaches," Journal of King Saud University-Computer and Information Sciences, vol. 35, no. 6, p. 101571, 2023.

[5] G. Jethava and U. P. Rao, "Exploring security and trust mechanisms in online social networks: An extensive review," Computers & Security, p. 103790, 2024.

[6] S. Shin, Z. Jiang, R. E. Lim, and J. Lyu, "Forecasting the Spread of Sustainability Movement: Computational Analysis on Social Media Messages Promoting Climate Actions," Journal of Current Issues & Research in Advertising, vol. 45, no. 3, pp. 282-300, 2024.

[7] G. V. Kumar, M. I. Bellary, and T. B. Reddy, "Prostate cancer classification with MRI using Taylor-Bird squirrel optimization based deep recurrent neural network," The Imaging Science Journal, vol. 70, no. 4, pp. 214-227, 2022.

[8] P. Rajesh, M. Ismail. Ismail. B, M. Alam, and M. Tahernezhadi, "Network forensics investigation in virtual data centers using elk," in 2021 International Symposium on Electrical, Electronics and Information Engineering, 2021, pp. 175-179.

[9] M. Shoeibi, M. M. S. Nevisi, S. S. Khatami, D. Martín, S. Soltani, and S. Aghakhani, "Improved IChOA-Based Reinforcement Learning for Secrecy Rate Optimization in Smart Grid Communications," Computers, Materials and Continua, vol. 81, no. 2, pp. 2819-2843, 2024, doi: https://doi.org/10.32604/cmc.2024.056823.

[10] V. Nyemeesha, M. Kavitha, and B. Mohammed Ismail, "Detection and classification of skin cancer using unmanned transfer learning based probabilistic multi-layer dense networks," International Journal of Computational Intelligence and Applications, vol. 21, no. 04, p. 2250027, 2022.

[11] M. Shoeibi, M. M. S. Nevisi, R. Salehi, D. Martín, Z. Halimi, and S. Baniasadi, "Enhancing Hyper-Spectral Image Classification with Reinforcement Learning and Advanced Multi-Objective Binary Grey Wolf Optimization," Computers, Materials and Continua, vol. 79, no. 3, pp. 3469-3493, 2024, doi: https://doi.org/10.32604/cmc.2024.049847.

[12] E. Bozorgi, S. Soleimani, S. K. Alqaiidi, H. R. Arabnia, and K. Kochut, "Subgraph2vec: A random walk-based algorithm for embedding knowledge graphs," arXiv preprint arXiv:2405.02240, 2024, doi: https://doi.org/10.48550/arXiv.2405.02240.

[13] P. Parimi and R. R. Rout, "Genetic algorithm based rumor mitigation in online social networks through counter-rumors: a multi-objective optimization," Information Processing & Management, vol. 58, no. 5, p. 102669, 2021.

[14] X. Hu, X. Xiong, Y. Wu, M. Shi, P. Wei, and C. Ma, "A hybrid clustered SFLA-PSO algorithm for optimizing the timely and real-time rumor refutations in online social networks," Expert Systems with Applications, vol. 212, p. 118638, 2023.

[15] S. Saeidi, "A new model for calculating the maximum trust in Online Social Networks and solving by Artificial Bee Colony algorithm," Computational Social Networks, vol. 7, no. 1, p. 3, 2020.

[16] N. Fatehi, H. S. Shahhoseini, J. Wei, and C.-T. Chang, "An automata algorithm for generating trusted graphs in online social networks," Applied Soft Computing, vol. 118, p. 108475, 2022.

[17] C. Mallick, S. Mishra, and M. R. Senapati, "A cooperative deep learning model for fake news detection in online social networks," Journal of Ambient Intelligence and Humanized Computing, vol. 14, no. 4, pp. 4451-4460, 2023.

[18] M. S. Vaghefi, D. L. Nazareth, S. P. Nerur, and K.-Y. Chen, "Self-disclosure in online social networks: An empirical study of location-based check-ins and impression management," Information & Management, vol. 61, no. 7, p. 104017, 2024.

[19] W. H. Bangyal, S. Amina, R. Shakir, G. Ubakanma, and M. Iqbal, "Using Deep Learning Models for COVID-19 Related Sentiment Analysis on Twitter Data," in 2023 International Conference on Human-Centered Cognitive Systems (HCCS), 2023: IEEE, pp. 1-6.

[20] W. H. Bangyal, M. Iqbal, A. Bashir, and G. Ubakanma, "Polarity Classification of Twitter Data Using Machine Learning Approach," in 2023 International Conference on Human-Centered Cognitive Systems (HCCS), 2023: IEEE, pp. 1-6.

[21] W. H. Bangyal et al., "Detection of Fake News Text Classification on COVID-19 Using Deep Learning Approaches," Computational and mathematical methods in medicine, vol. 2021, no. 1, p. 5514220, 2021.

[22] L. Abualigah, D. Yousri, M. Abd Elaziz, A. A. Ewees, M. A. Al-Qaness, and A. H. Gandomi, "Aquila optimizer: a novel meta-heuristic optimization algorithm," Computers & Industrial Engineering, vol. 157, p. 107250, 2021.

[23] H. R. Tizhoosh, "Opposition-based learning: a new scheme for machine intelligence," in International conference on computational intelligence for modelling, control and automation and international conference on intelligent agents, web technologies and internet commerce (CIMCA-IAWTIC'06), 2005, vol. 1: IEEE, pp. 695-701.

[24] A. G. Hussien, "An enhanced opposition-based salp swarm algorithm for global optimization and engineering problems," Journal of Ambient Intelligence and Humanized Computing, vol. 13, no. 1, pp. 129-150, 2022.

[25] A. G. Hussien and M. Amin, "A self-adaptive Harris Hawks optimization algorithm with opposition-based learning and chaotic local search strategy for global optimization and feature selection," International Journal of Machine Learning and Cybernetics, vol. 13, no. 2, pp. 309-336, 2022.

[26] Y.-J. Zhang, Y.-F. Wang, Y.-X. Yan, J. Zhao, and Z.-M. Gao, "LMRAOA: An improved arithmetic optimization algorithm with multi-leader and high-speed jumping based on opposition-based learning solving engineering and numerical problems," Alexandria Engineering Journal, vol. 61, no. 12, pp. 12367-12403, 2022.

[27] J. Zhao, Y. Zhang, S. Li, Y. Wang, Y. Yan, and Z. Gao, "A chaotic self-adaptive JAYA algorithm for parameter extraction of photovoltaic models," Math. Biosci. Eng, vol. 19, no. 6, pp. 5638-5670, 2022.

[28] Y. Yu, S. Gao, S. Cheng, Y. Wang, S. Song, and F. Yuan, "CBSO: a memetic brain storm optimization with chaotic local search," Memetic Computing, vol. 10, pp. 353-367, 2018.

[29] H. Chen, Y. Xu, M. Wang, and X. Zhao, "A balanced whale optimization algorithm for constrained engineering design problems," Applied Mathematical Modelling, vol. 71, pp. 45-59, 2019.

[30] H. Zhang et al., "Ensemble mutation-driven salp swarm algorithm with restart mechanism: Framework and fundamental analysis," Expert Systems with Applications, vol. 165, p. 113897, 2021.