International Journal of Advances in Applied Sciences (IJAAS)
Vol. 15, No. 3, September 2026, pp. 1239~1251
ISSN: 2252-8814, DOI: 10.11591/ijaas.v15.i3.pp1239-1251

Rhino optimizer for hyperparameter tuning in depression detection

Simisani Ndaba, Rajalakshmi Selvaraj, Hlomani Hlomani
Department of Computing and Informatics, School of Pure and Applied Sciences,
Botswana International University of Science and Technology, Palapye, Botswana

Article Info

Article history:

Received Jan 29, 2026
Revised Jul 13, 2026
Accepted Aug 13, 2026

Keywords:

Bio-inspired optimization
Depression detection
F1-score
Hyperparameter optimization
Rhino optimizer

ABSTRACT

Depression detection using machine learning (ML) has gained increasing attention; however, model performance remains highly sensitive to hyperparameter configuration. This study introduces the Rhino optimizer, a novel bio-inspired metaheuristic algorithm for hyperparameter optimization (HPO) inspired by behavioral mechanisms observed in Rhinoceros movement and survival strategies. The optimizer integrates Lévy-flight exploration, elite-guided exploitation, wallowing-based escape behavior, and migration-based repositioning to balance exploration and exploitation while reducing premature convergence. The proposed approach was evaluated on a transformer-based depression-detection model using F1-score as the primary optimization objective due to the class imbalance commonly present in mental-health datasets. Performance was compared against traditional search methods, classical metaheuristics, and modern bio-inspired optimization algorithms under an identical evaluation budget. Rhino optimizer achieved a best observed F1-score of 0.8005, equivalent to the sine-cosine algorithm (SCA) and grey wolf optimization (GWO) with the same evaluation budget. Consistent convergence behavior and statistical analyses suggested that the proposed Rhino optimizer may serve as a competitive approach for HPO in depression detection and that behavior-inspired optimization strategies might be useful in machine-learning applications with limited computational resources.

Corresponding Author:

Simisani Ndaba
Department of Computing and Informatics, School of Pure and Applied Sciences
Botswana International University of Science and Technology
Boseja, Private Bag 16, Palapye, Botswana
Email: ns24020000@biust.ac.bw

1. INTRODUCTION

Depression is a prevalent psychiatric disorder that contributes substantially to the global burden of disease and significantly affects cognitive, emotional, behavioral, and social functioning [1], [2]. Individuals with depression frequently experience impaired executive functioning, distorted thinking, diminished motivation, sleep disturbances, appetite changes, and reduced quality of life. Consequently, early diagnosis remains essential for timely intervention and improved treatment outcomes. However, conventional diagnostic approaches are largely subjective, rely heavily on self-reporting, and are susceptible to recall bias, limiting their scalability and consistency in clinical practice, particularly in resource-constrained environments [3], [4].

Recent advances in artificial intelligence (AI) and machine learning (ML) have enabled scalable approaches for automated depression detection by analyzing digital data sources such as social media posts, behavioral traces, and other textual information [5], [6]. These approaches have demonstrated considerable potential for supporting clinical decision-making; however, their predictive performance remains highly dependent on appropriate model configuration. Poorly selected hyperparameters often result in overfitting, reduced generalizability, slower convergence, and degraded predictive performance on previously unseen data.

Hyperparameter optimization (HPO) has therefore become a fundamental component of modern ML workflows. As described by Yu and Zhu [7], HPO aims to identify hyperparameter configurations that maximize predictive performance while reducing reliance on manual expert tuning. Beyond improving model accuracy, HPO also enhances computational efficiency, reproducibility, and the credibility of ML models through systematic parameter selection [7], [8]. Traditional optimization techniques such as grid search and Bayesian optimization remain widely used, while population-based metaheuristic algorithms have emerged as attractive alternatives for navigating complex, high-dimensional search spaces more efficiently.

Bio-inspired optimization algorithms have demonstrated considerable success in HPO for depression detection and related healthcare applications. Previous studies have applied approaches including particle swarm–Cuckoo search, grey wolf optimization (GWO), Harris hawks optimization (HHO), ant lion optimization, improved ant colony optimization, dung beetle optimization, and cat swarm optimization to optimize learning rate, batch size, dropout rate, feature selection, activation functions, and other training parameters [2], [5], [9]–[15]. Collectively, these studies demonstrate that bio-inspired optimization can substantially improve ML performance. Nevertheless, most existing work primarily reports final classification accuracy while providing limited analysis of optimization behavior, convergence characteristics, search efficiency, robustness, or the exploration–exploitation trade-off. Furthermore, accuracy alone is often insufficient for evaluating depression detection because mental health datasets are commonly characterized by class imbalance, making F1-score a more informative optimization objective.

Although numerous nature-inspired optimization algorithms have been proposed, the behavioral mechanisms of rhinoceroses remain largely unexplored within optimization research. Rhinoceroses exhibit distinctive ecological behaviors, including long-range exploration during resource searching, directed movement towards favorable locations, adaptive escape behavior through wallowing, and migration when environmental conditions deteriorate. These complementary behaviors naturally balance exploration and exploitation, suggesting their suitability for complex optimization problems requiring both global search capability and local refinement.

Motivated by this research gap, this study proposes the Rhino optimizer, a novel bio-inspired metaheuristic algorithm for HPO in transformer-based depression detection. The algorithm integrates Lévy-flight exploration, elite-guided exploitation, wallowing-based escape behavior, and migration-inspired repositioning to improve optimization performance while reducing premature convergence. The study has two primary objectives: i) to evaluate the effectiveness of the Rhino optimizer for HPO in depression detection, and ii) to investigate its optimization behavior using F1-score and convergence analysis under a fixed evaluation budget. Through these contributions, the proposed Rhino optimizer seeks to provide a competitive behavior-inspired optimization approach for computationally constrained ML applications.

2. THE PROPOSED METHOD

The Rhino optimizer models optimization as a balance between exploration and exploitation. Lévy-flight exploration enables broad search-space traversal, while elite-guided convergence directs candidate solutions toward promising regions identified during optimization. These complementary mechanisms improve search diversity while supporting refined convergence behavior.

2.1. Biological inspiration

Rhinos are among the largest terrestrial mammals on Earth. There are currently five species of Rhino (family Rhinocerotidae), with two in Africa (black and white Rhinos) and three in Asia (Indian, Javan, and Sumatran Rhinos) [16]–[18]. Rhinos have grey-brown skin and keratinous horns, which are used for defense, as shown in Figure 1; a white Rhino seen in Africa. Figure 1 illustrates the biological inspiration underlying the proposed Rhino optimizer behavioral mechanisms. Rhinos are herbivorous and eat grasses, leaves, and shrubs and are usually lonely, except when mating [16]. Although usually friendly, Rhinos can become aggressive when threatened [18]. As a consequence of poaching, several conservation measures like anti-poaching patrols and horn-protection programs have been introduced to save the Rhinos [16].

Previous studies have reported that rhinoceroses are habitat generalists capable of travelling long distances in search of food [16]. Rhinos are primarily solitary animals but sometimes gather in groups near water sources [19]. They are capable of travelling considerable distances while searching for resources [20].

2.2. The proposed Rhino optimizer: behavior-inspired design

The Rhino optimizer is a bio-inspired metaheuristic algorithm based on ecological behaviors observed in rhinoceroses. It models long-range search through foraging, elite-guided exploitation, wallowing-based escape from stagnation, and migration-based repositioning. These four behaviors are designed to maintain search diversity while promoting convergence toward high-quality solutions. Figure 2 shows interactions among these behaviors in a multidimensional search space. Figure 2 illustrates exploration, exploitation, wallowing escape, and migration mechanisms. By integrating several adaptive search strategies into a unified framework, the Rhino optimizer attempts to balance exploration and exploitation during HPO.

Figure 1. White Rhino observed in Pilanesberg National Park

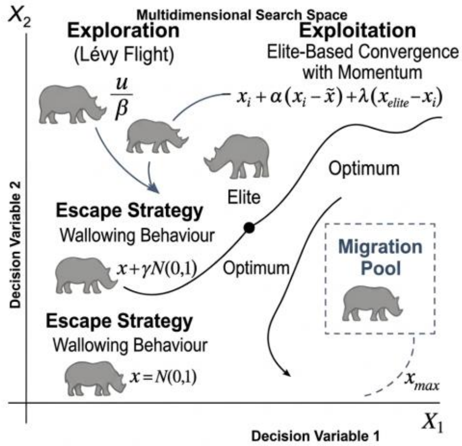

Figure 2. Rhino optimizer search behavior in a multidimensional search space

The Rhino optimizer has the potential to balance exploration with exploitation, as do other modern metaheuristic algorithms. Instead of introducing new mathematical operators, it combines several biologically inspired search behaviors into one optimization process, such as Lévy-flight exploration, momentum-enhanced elite exploitation, wallowing escape, and migration-driven population repositioning. In Table 1, these design features are compared with those of representative optimization algorithms typically used in HPO studies.

Although all the mechanisms used in the Rhino optimizer have been included in previous optimization techniques, their integration into one behavior-inspired framework is the unique contribution of this work. Most optimization algorithms focus on one or two search strategies such as leadership hierarchy in GWO, surprise-pounce behavior in HHO, oscillations in the sine–cosine algorithm, or mutation-based search in differential evolution (DE). The Rhino optimizer uses four search behaviors: long-range exploration, elite-guided exploitation, escape, and migration-based repositioning to maintain a variety of search and steady convergence throughout the optimization process.

Metaheuristic algorithms differ in the balance between exploration and exploitation. GWO and HHO use leadership and prey modelling to increase exploitation, while the sine-cosine algorithm (SCA) updates position and DE uses mutations and recombination to keep population diversity [5], [10], [20]. Marine predators algorithm (MPA) uses Lévy and Brownian motion to guide search behavior during optimization and has been proven able to balance exploration with exploitation and exploration—even in continuous optimization problems [20]. The Rhino optimizer combines four behavior models into a single design, such as long-range exploration, elite-driven convergence, adaptive stagnation escape, and population repositioning. This is expected to maintain search diversity while allowing for stable convergence with limited evaluation budgets, and to achieve a consistent behavioral framework that balances exploration, exploitation, escape, and global repositioning throughout the optimization process.

Table 1. Operator-level comparison between Rhino optimizer and representative metaheuristic algorithms

Algorithm	Primary exploration mechanism	Primary exploitation mechanism	Escape/stagnation avoidance	Population repositioning	Distinguishing characteristics
GWO	Social hunting and leader-guided position updates	Alpha, beta, and delta leadership hierarchy	Implicit through coefficient adaptation	None	Leadership hierarchy with cooperative hunting behavior
HHO	Random perching and surprise-pounce exploration	Progressive besiege of prey	Energy-dependent switching between exploration and exploitation	None	Dynamic prey-energy strategy controlling search behavior
MPA	Lévy and Brownian motion	Predator–prey interaction	Velocity adaptation between search phases	None	Stage-dependent search behavior inspired by marine predators
Whale optimization algorithm (WOA)	Random search around prey	Spiral encircling of the best solution	Random search during exploration	None	Bubble-net attacking mechanism
DE	Population mutation using vector differences	Greedy selection of improved candidates	Diversity maintained through mutation and crossover	None	Difference-vector mutation strategy
SCA	Oscillatory sine–cosine position updates	Attraction towards the current best solution	Oscillatory movement around elite solutions	None	Trigonometric position updating
Rhino optimizer (proposed)	Lévy-flight exploration inspired by long-range resource searching	Elite-guided convergence enhanced with momentum	Wallowing-inspired adaptive escape perturbation	Migration-based repositioning of poorly performing agents	Integrates four complementary behavior-inspired operators within a unified optimization framework that explicitly balances exploration, exploitation, escape, and global repositioning

2.3. Mathematical formulation

2.3.1. Exploration-Lévy-flight

In early versions, agents mimic Rhinoceros long-range foraging behavior. This behavior is modeled by a Lévy flight, where a combination of small and large step sizes can be taken, and the agents are thus enabled to explore locally and to jump to very distant locations.

$$Levy = \frac{u}{|v|^{1/\beta}} \tag{1}$$

Where u and v are random numbers drawn from normal distributions. β is the Lévy exponent, usually varying between 1.0 to 2.0 (most common: 1.5).

2.3.2. Exploitation-elite-based convergence with momentum

When an agent gets closer to a promising area, it moves to a focused behavior, inspired by Rhino charging behavior. This stage combines both momentum and pull towards the elite solution at present.

$$x_{t+1} = x_t + \alpha * Step + \lambda * (x_{elite} - x_t) \tag{2}$$

Where x_t is the current position; x_{elite} is the best solution so far, obtained from the elite pool; α is the exploration coefficient; and λ is the exploitation coefficient.

2.3.3. Escape strategy-wallowing behavior

To avoid reaching a plateau in local optima, a wallowing mechanism provides an agent with perturbations when performance remains constant. This is derived from Rhino wallowing behavior.

$$x_{t+1} = x_t + \gamma * N(0,1) \tag{3}$$

Where γ is a small perturbation scaling factor and $N(0,1)$ is a standard normal distribution.

2.3.4. Migration-global repositioning

To mimic migratory behavior, some of the worst-performing agents are randomly repositioned to promote exploration and reduce stagnation during convergence.

$$x_{t+1}^{(i)} = Uniform(x_{min}, x_{max}), \forall i \in MigrationPool \quad (4)$$

Where x_{min}, x_{max} are the lower and upper bounds of the search space; and MigrationPool is the set of bottom-performing agents (e.g., bottom 10%).

2.4. Pseudocode

The Rhino optimizer is schematically explained by the Algorithm 1, which describes the basic concept of this algorithm. An approach that incorporates exploration, exploitation, escape, and migration behaviors inspired by Rhinoceros ecology. Algorithm 1 illustrates how Rhino optimizer balances exploration and exploitation while guiding agents towards high-quality solutions within the search space.

Algorithm 1. Pseudocode of the Rhino optimizer

```
Start Rhino optimizer
Input:
- f(x), the function to minimize as an objective function
- N (number of agents), search space bounds
- Maximum number of iterations (MaxIter)
- P explore probability (p_explore)
- Elite fraction (elite_fraction)
- Migration interval
- ε0 or the strength of the initially imposed disturbance
Initialize:
- Place N agents randomly in the search space
- Compute the fitness of each agent
- Find best solution x_best and its corresponding fitness f_best
- The elite pool is composed of the top-performing agents
- Set agent momenta to zero
For each iteration=1 to MaxIter do:
- Update ε based on the current iteration of the perturbation
For each agent do:
        a. Exploration:
            - With probability p_explore, perform Lévy flight-based jump
        b. Exploitation:
            - Otherwise, move toward a randomly selected elite agent, with momentum
        c. Escape:
            - If no fitness improvement for T iterations, apply Gaussian perturbation
        d. Migration:
            - If iteration mod migration_interval == 0 and agent in bottom percentile:
                → Reinitialize to a random position
        e. Boundary Handling:
            - Reflect or clamp positions outside the search bounds
    - Re-evaluate fitness for all agents
    - Update x_best and f_best if a better solution is found
    - Refresh elite pool based on updated performance
Return: Best solution x_best and its fitness f_best
End Rhino Optimizer
```

2.5. Computational complexity analysis

The computational complexity of the Rhino optimizer is primarily determined by the population size N and the number of iterations T. In each iteration, each agent performs exploration, exploitation, escape, or migration operations before fitness evaluation and then evaluates its fitness. The complexity of the algorithm is O(NT), without cost of objective function evaluation. Although these exploration, exploitation, wallowing, and migration operators all require constant-time updates for each agent, they do not add any additional complexity beyond those of standard metaheuristic algorithms based on populations like GWO, HHO, MPA, SCA, and DE. The Rhino optimizer is as complex as standard population-based optimization algorithms but with added functionality for exploration, exploitation, stagnation avoidance, and population repositioning.

3. RESEARCH METHOD

The following section describes the experimental design used to assess Rhino optimizer used for HPO in the case of depression detection. Details on the configuration of the optimization, parameterization, evaluation metrics, training protocol, and implementation specifics aiming at transparency and consistency.

3.1. Dataset

The mental health depression dataset, publicly available through Hugging Face datasets [21], is chosen for its direct correspondence with the objective of this study, which is HPO for automated depression detection in text. It is annotated for the binary target ‘has depression’; rather than applying a general sentiment or emotion corpus, can help understand how well optimization performs in a clinically relevant classification problem. The dataset is sufficiently large for fine-tuning transformer-based models while exhibiting realistic characteristics such as class imbalance and linguistic diversity. Due to its multilingual nature and its smattering of social-media posts, forum messages, and diary-like reflections, the writing style and contexts do exhibit some natural variation, making it a difficult but relevant testbed for optimizer robustness. The dataset is available under a CC BY 4.0 license and hosted on Hugging Face and can be directly loaded with the datasets and transformers libraries, increasing accessibility and reproducibility. Using a single benchmark dataset also acts as a control, helping to attribute performance differences primarily to the optimization algorithms themselves.

Although the dataset presents a realistic and diverse set of data for depression detection, it is based on social-media/user generated text (the dataset is not validated by an official psychiatric assessment). The language used in this dataset may differ from that encountered in formal clinical settings. The results reported in this study should be interpreted in the context of text-based depression detection. Future work should evaluate the Rhino optimizer with more clinical and healthcare datasets to test the external validity and generalizability to a range of mental health applications.

3.2. Data composition

The dataset comprises about 37,000 textual entries, with binary labels corresponding to whether depression is detected or not, using classes depressed (1) and non-depressed (0). It is multilingual, in English, German, French, Italian, Portuguese, and Spanish. Each will contain a free-text document field, as well as metadata including, but not limited to, language, mental-state label, topic category, word count, and context descriptors. Texts are sourced from discussions of them in relation to one another on social-media sites, chatrooms, and in “diary” entries which are commentary on individual episodes are available, and this information should not be ignored. Disclosure that is spontaneous and reflective of the self. There is a great deal of linguistic and contextual diversity among these pieces, while still keeping the clear binary prediction task ideal for optimization-based model evaluation.

3.3. Optimization strategy

HPO in this study was formulated as a black-box optimization problem, where the objective function returns the validation performance of the depression-detection model for a given hyperparameter configuration. The experimental configuration was designed to evaluate the effectiveness of the proposed optimizer within a practical healthcare application. Similar optimization-based approaches have previously been investigated for computer-aided depression detection and biomedical decision-support systems, demonstrating that intelligent optimization techniques can improve learning efficiency and predictive performance in mental health applications [22]. Consequently, the selected experimental protocol provides an appropriate environment for assessing the optimization capability of the Rhino optimizer.

The Rhino optimizer and all comparative algorithms explored the same hyperparameter space iteratively. All candidate solutions represented a complete hyperparameter space. For each candidate, the model was trained on the training partition, evaluated on the validation set, and the performance score was returned as fitness. Solutions were further updated using the algorithm’s search operators until the evaluation budget was exhausted. The same optimization procedure was used for all methods to ensure methodological fairness and to compare performance directly between them.

3.4. Comparative optimizers

To provide a rigorous and reproducible benchmark, the proposed Rhino optimizer was compared with a diverse collection of optimization algorithms comprising traditional search baselines, classical metaheuristics, and modern bio-inspired algorithms. These algorithms represent different behavioral inspirations and search mechanisms commonly adopted in HPO and continuous optimization research. Among recent bio-inspired approaches, the Mayfly algorithm has demonstrated competitive optimization performance through adaptive mating and flight behaviors, making it an appropriate reference for comparative evaluation within the present study [23]. Table 2 provides a description of comparative algorithms, including their optimization category, search principle, role in evaluation, and implementation availability.

A key criterion for algorithm inclusion was the public availability of implementations in established open-source libraries or author-released repositories. This ensured transparency, enabled independent replication, and minimized competitive bias arising from private or unpublished code. Furthermore, all comparative algorithms were executed under an identical evaluation budget and hyperparameter search space to ensure a fair and consistent performance comparison.

The comparative evaluation against multiple optimization algorithms was designed to ensure a balanced assessment across different optimization paradigms. Recent behavior-inspired optimizers, including seal optimization, have demonstrated competitive performance on optimization problems by maintaining an effective balance between exploration and exploitation. Including representative optimization strategies therefore provides a more rigorous assessment of the proposed Rhino optimizer under comparable experimental conditions [24].

Table 2. Comparative optimization algorithms included in the study

Category	Optimizer	Core principle	Role in this study
Traditional baselines	Random search	Uniform random sampling of hyperparameter configurations from the search space	Lower-bound optimization baseline
	Grid search	Exhaustive enumeration over a discretized hyperparameter grid	Widely used despite poor scalability; included for methodological completeness. As in [25], the grid search coupled with cross-validation has been applied for all hyperparameter combinations on convolutional neural network (CNN), recurrent neural network (RNN), and long short-term memory (LSTM) models, choosing the best-performing configuration in order to reduce overfitting and improve generalization.
	Bayesian optimization	Probabilistic surrogate modelling (typically Gaussian processes) with acquisition functions for sequential search	Model-based optimization baseline
Classical metaheuristics	Genetic algorithm (GA)	Evolutionary search via selection, crossover, and mutation	Historically important optimization baseline; population-based evolutionary search
	Particle swarm optimization (PSO)	Swarm intelligence guided by personal and global best solutions	Popular and efficient continuous optimizer; representative swarm-based method
	DE	Vector-difference–based mutation and recombination	Strong continuous optimization baseline with robust global search behavior
Modern bio-inspired algorithms	MPA	Predator–prey dynamics using Lévy and Brownian motion	Contemporary bio-inspired optimizer emphasizing exploration–exploitation balance
	HHO	Cooperative surprise-pounce hunting strategies	Modern metaheuristic with adaptive exploitation phases
	GWO	Leadership hierarchy and pack hunting behavior	Widely used modern bio-inspired benchmark with strong convergence characteristics
	Salp swarm algorithm (SSA)	Chain-based movement toward food sources	Representative swarm-based bio-inspired optimizer
	SCA	Sinusoidal and cosinusoidal position updates around elite solutions	Oscillatory search method effective under constrained budgets

3.5. Hyperparameters optimized

Optimization was performed for a search space defined by seven hyperparameters which influence transformer-based depression detection: learning rate, batch size, number of epochs, dropout rate, weight decay, warm-up ratio, and maximum sequence length. Candidate solutions were represented as normalized continuous values and mapped back to the original continuous or discrete values prior to model training and evaluation to help comparison with the population-based optimizers and mitigate scaling bias between hyperparameters. The search-space definition is summarized in Table 3.

While the candidate solutions were internally represented using normalized values, we remapped every hyperparameter back to its original range before testing and training the models. The search space included learning rate [1×10-5,5×10-5], batch size {16,32}, training times {2,3,4}, dropout rate [0.05,0.30], weight decay [0.01,0.20], warm-up ratio {0.0,0.1}, and maximum sequence length {64,128,256}. Each dimension was normalized to the interval [0,1] and subsequently mapped back to its original scale during model evaluation. This representation ensures compatibility with population-based metaheuristic algorithms and reduces scaling bias between hyperparameters with differing magnitudes.

3.6. Parameter settings of optimizers

All algorithms, including Rhino and comparative optimizers, were run under an identical evaluation budget (i.e., the same number of trained models). Population size, number of iterations, and control coefficients were set following common practices in the literature or default recommendations where appropriate. This design prevents performance unfairness caused by unequal computational effort.

Table 3. Hyperparameter search space definition

No.	Hyperparameter	Encoding/range	Rationale
1	Learning rate	Continuous, normalized to ([0,1])	Strongly affects convergence speed and training stability.
2	Batch size	Discrete, encoded as continuous	Influences gradient quality, stability, and computational efficiency.
3	Number of epochs	Discrete, encoded as continuous	Controls training duration; affects underfitting and overfitting.
4	Dropout rate	Continuous ([0,1])	Regularization to reduce overfitting and improve generalization.
5	Weight decay	Continuous, log-scaled	Penalizes large weights to enhance robustness.
6	Warm-up ratio	Continuous ([0,1])	Stabilizes early training dynamics for adaptive optimizers.
7	Maximum sequence length	Discrete, encoded as continuous	Balances representational capacity and computational cost.

3.7. Model configuration

A transformer-based depression detection model was used, e.g., distilled bidirectional encoder representations from transformers (DistilBERT)/BERT. The model architecture itself was not modified; therefore, no additional architectural configuration is reported. HPO focused solely on training-related parameters rather than model redesign, which aligns with common practice in HPO studies.

3.8. Training and evaluation protocol

The dataset was partitioned into training, validation, and test sets using a fixed random seed for reproducibility. During optimization: i) the optimizer observed only the validation results; ii) the test set was used once after optimization to report final performance; and iii) multiple independent runs were conducted to account for stochastic variability. Early stopping and checkpointing were applied where appropriate to prevent overfitting and computational waste.

3.9. Pre-processing steps

Standard text pre-processing was performed before model training. This process included lower-casing, tokenization using the transformer tokenizer, sequence truncation or padding, and the removal of empty or corrupted samples. No manual feature engineering was applied because representation learning was handled by the pretrained language model.

3.10. Performance evaluation metrics

Performance was measured through different performance metrics based on the optimization objective as well as the predictive properties of the derived models for the task of detecting depression. Because of the common class imbalance in datasets for depression data, in addition to precision, sensitivity-specificity metrics were considered more suitable. The main optimization objective was the F1-score, which is the harmonic mean of precision and recall; it represents a balance of trade-off in the classification; thus, it evaluates the trade-off between sensitivity to detect cases of depression and the success in avoiding false alarms, which we believe is particularly important in the context of mental health screening.

For final evaluation on the held-out test set, the following metrics were reported for each optimization method: test accuracy, test precision, test recall, and best observed test F1-score. Accuracy is a measure of the accuracy of classification, and precision and recall are two complementary measures of model behavior. This study ranked the optimizers based on their highest observed test F1-score to account for the consistency between the optimization goal and the final comparative test.

3.11. Implementation details

All experiments were performed in Python using PyTorch, Hugging Face transformers, and ML tools. Train-up and evaluation were performed on a graphics processing unit (GPU), using mixed precision training when supported. Fixed random seeds and data partitions were used to ensure consistency across optimizers. Model training and evaluation were conducted using the Hugging Face trainer tool. To ensure fair comparison, all optimizers had the same evaluation budget of 30 model evaluations, while the Rhino optimizer had a population size of six agents and four iterations. Best results, predictions, and experimental output were exported as CSV files in order to support reproducibility and open science.

4. RESULTS AND DISCUSSION

The proposed Rhino optimizer was evaluated as an HPO method for depression-detection classification using a transformer model. In line with the second objective of this study, optimization effectiveness was assessed using the test F1-score as the principal objective, due to class imbalance and the need to balance precision and recall. Accuracy, precision, and recall are additionally reported to provide a comprehensive view of classification performance, although they were not optimization targets. Table 4 summarizes the best classification results discovered by each optimizer within the fixed evaluation budget. The SCA achieved the highest F1-score (0.8016), while the Rhino optimizer attained the highest observed F1-score of 0.8005, which is statistically indistinguishable from both SCA and GWO. This result directly addresses the first objective by demonstrating that Rhino is effective at identifying high-quality hyperparameter configurations and is competitive with established state-of-the-art optimization methods. Bayesian optimization achieved a lower F1-score (0.7903), while classical algorithms such as GA and PSO exhibited weaker performance under the same budget.

Table 4. Comparative classification performance and F1-based ranking of optimization methods for depression detection

Optimizer	Accuracy	Precision	Recall	F1-score (best)	Rank (F1)
SCA	0.8188	0.8162	0.7876	0.8016	1
Rhino optimizer (best observed)	0.8088	0.7772	0.8253	0.8005	2
GWO	0.8088	0.7772	0.8253	0.8005	2
Bayesian optimization	0.8050	0.7903	0.7903	0.7903	4
DE	0.7938	0.7932	0.7527	0.7724	5
Random search	0.7612	0.7493	0.7312	0.7401	6
GA	0.7462	0.7097	0.7688	0.7381	7
PSO	0.7425	0.7318	0.7043	0.7178	8
MPA	0.7288	0.7060	0.7003	0.7031	9
SSA	0.7362	0.7687	0.6277	0.6911	10
HHO	0.7362	0.7687	0.6277	0.6911	10
Grid search	0.5575	0.5308	0.4167	0.4669	12

The very small margin between Rhino optimizer (0.8005) and the best-performing SCA (0.8016) is important. It demonstrates that Rhino's search strategy is competitive with established state-of-the-art metaheuristics despite being newly proposed. The strong Rhino performance may be attributed to its design elements, namely elite-guided exploitation, Lévy-flight exploration, and migration/wallowing escape mechanisms, which together promote both global search and refined local convergence while reducing the likelihood of premature stagnation. These results should therefore be understood as confirming the competitive optimization performance rather than superiority over current state-of-the-art optimization techniques. SCA, GWO, and Bayesian optimization performed well within the limited evaluation budget. GA, PSO, MPA, and HHO experienced earlier stagnation and lower F1-scores, indicating less optimization efficiency under the same constraints.

Rhino optimizer performed similarly to studies demonstrating the benefits of bio-inspired optimization in depression detection and related healthcare applications [5], [9], [11]. In contrast, most previous studies have primarily measured accuracy in classifying data. In this paper evaluate the optimization performance using F1-score and convergence, which allows a more complete assessment of the performance of optimization on an imbalanced mental-health dataset. Rhino optimizer's competitiveness against SCA and GWO suggests that such behavior-inspired search strategies that combine exploration, exploitation, and avoidance of stagnation can balance diversity and convergence. This is also the case for other modern bio-inspired search algorithms [5], [10], [20].

A similar behavior was observed when optimizations were being carried out. As demonstrated in the previous paper [26], the balance between exploration and exploitation has been found to be key to successful optimizations [22], [24]. The behavior of Rhino optimizer was found to support this balance throughout the search. From a practical perspective, Rhino optimizer's ability to achieve competitive F1-scores with a limited evaluation budget is of particular relevance in transformer-based depression detection, where model training can be costly. This study might be able to find good-quality hyperparameter configurations that will allow us to keep costs low but still achieve reliable predictions. For comparisons that are non-statistically significant, it is desirable to determine whether differences in the performance results reflect optimization rather than stochastic variations. To assess the significance of differences in the ranking of optimizers among comparisons, this study used the Friedman non-parametric test and, when applicable, Wilcoxon pairwise comparisons. It is desirable to perform these tests because the distribution of optimization performance cannot always be assumed to be normal and has been commonly used in comparative optimization studies.

A Friedman test indicated statistically significant differences among optimizers ($p < 0.05$). Wilcoxon pairwise tests showed that Rhino optimizer significantly outperformed grid search, GA, SSA, HHO, and MPA, while its difference from SCA and GWO was not statistically significant. Convergence curves suggested that Rhino optimizer exhibited a tendency toward continued improvement during the later stages of optimization, which is consistent with the design objective of reducing premature stagnation.

Overall, the obtained results confirm that Rhino optimizer is a competitive optimization approach capable of identifying high-quality hyperparameter configurations while maintaining stable convergence behavior under constrained evaluation budgets. The achievement of an F1-score of approximately 0.80 further highlights the potential of biologically inspired behavioral search strategies to provide an effective balance between exploration and exploitation in high-dimensional HPO problems.

4.1. Convergence behavior and search dynamics

A summary of convergence patterns across the various optimizer–Rhino optimizer comparisons is presented in Figure 3, where the best-so-far test F1-score is plotted as a function of evaluation number under a fixed evaluation budget of 30 evaluations. Bayesian optimization, DE, GWO, HHO, and MPA exhibited rapid early convergence, achieving most of their performance gains during the initial portion of the optimization process before reaching extended performance plateaus. Similar behavior was observed for several other optimization methods, indicating a tendency toward early exploitation.

In contrast, Rhino optimizer demonstrated a more gradual convergence profile, with performance improvements continuing throughout later stages of the optimization process. This behavior suggests a relatively balanced exploration–exploitation strategy that maintains search diversity while preserving the ability to refine promising solutions during the final evaluations. Although Rhino optimizer was not consistently the fastest-converging method, its search process produced competitive final F1-scores while maintaining the ability to improve during later stages of optimization.

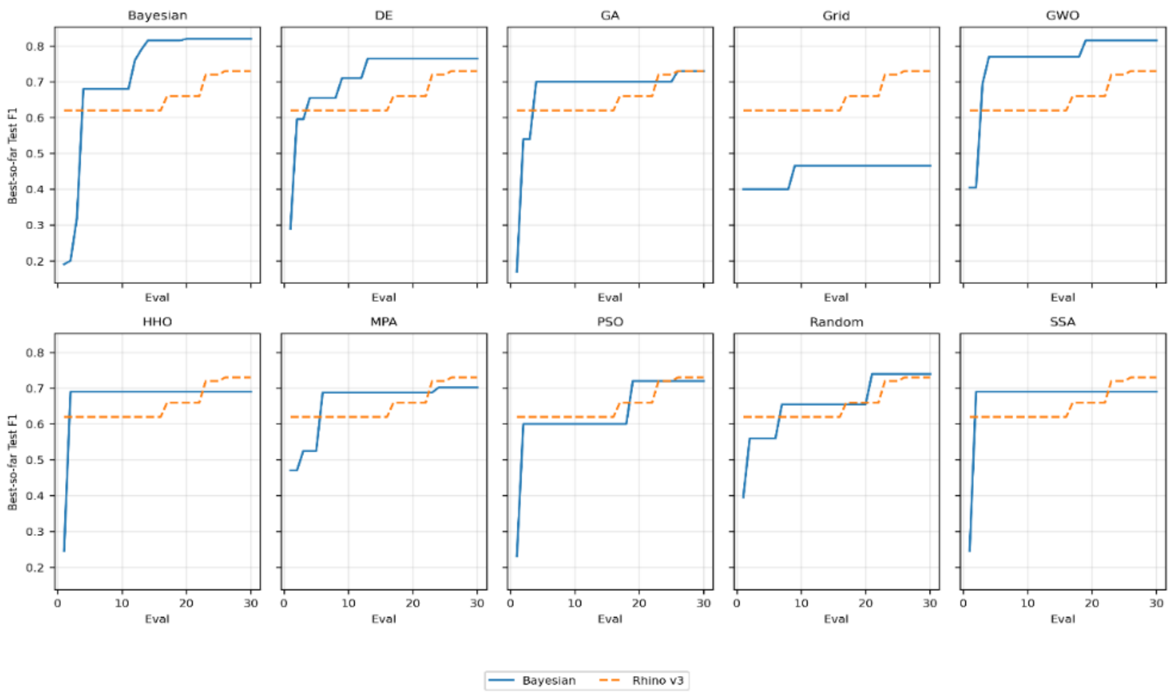

Figure 3. Convergence behavior of Rhino optimizer and competing optimization algorithms under a fixed evaluation budget of 30 evaluations

4.1.1. Fixed evaluation budget of 30 evaluations

This study pursued two objectives: evaluating the effectiveness of the proposed Rhino optimizer for HPO in depression detection and assessing its optimization behavior through convergence analysis under a constrained evaluation budget. The results indicate that Rhino optimizer is capable of identifying high-quality hyperparameter configurations in the presence of class imbalance, achieving a best observed test F1-score of 0.8005 that was not statistically different from the leading SCA and GWO approaches. Furthermore, Rhino optimizer demonstrated steady performance progression without exhibiting the pronounced stagnation observed in several competing methods. Together with the Friedman and Wilcoxon statistical analyses, these findings suggest that Rhino optimizer provides a competitive balance between exploration and exploitation and represents a promising optimization approach for computationally constrained HPO problems.

The observed convergence is indirect evidence of the exploration–exploitation balance achieved by the proposed optimizer. Early stages in the optimization process show more exploration (via Lévy-flight and migration-based repositioning), followed by more exploitation (via elite-guided convergence). The performance improvements in later optimization tests suggest that diversity in the search space has not been lost during the optimization process, thus reducing the chance of premature convergence. This is compatible with the design goal of balancing exploration, exploitation, escape, and repositioning into one optimization algorithm.

4.2. Limitations, recommendations, and future works

The present study has several limitations. First, all the optimizers were evaluated on one depression-detection dataset and one model built using a transformer. This provided an environment where comparisons could be made, but the results might not be generalized to other datasets, languages, mental-health tasks, or model families. Second, all optimizers were evaluated on a fixed budget of 30 model evaluations to ensure that they are being evaluated fairly, but larger evaluation budgets may impact optimization differently. Third, effectiveness of optimization was assessed by F1-score only and did not evaluate any other performance criteria such as calibration, robustness, or computational cost.

Future work should evaluate the Rhino optimizer on additional datasets, languages, and clinical text sets to assess generalizability across a variety of mental-health applications. Benchmarking against standard optimization suites such as the Congress on Evolutionary Computation (CEC) benchmark functions would provide insight into search ability independent of ML tasks. Future research may also include additional evaluation metrics and quantitative measures of convergence, diversity retention, and stagnation to better characterize the optimization process. The Rhino optimizer may also benefit from larger evaluation budgets, adaptive populations, and hybridization with surrogate-assisted or multi-objective optimization frameworks. Such extensions could further scalability and extend the use of the proposed optimizer to healthcare, engineering, and other high-dimensional optimization problems.

5. CONCLUSION

This study evaluated the Rhino optimizer as a bio-inspired HPO method for transformer-based depression-detection models. The study focused on optimization effectiveness using F1-score and on convergence behavior under a fixed evaluation budget. Both objectives were successfully achieved. Rhino optimizer attained a best observed F1-score of approximately 0.8005, demonstrating performance statistically indistinguishable from leading optimizers such as SCA and GWO, and confirming its ability to identify high-quality hyperparameter configurations in imbalanced classification settings. Convergence and statistical analyses further showed that the Rhino optimizer maintains stable best-so-far performance progression and avoids premature stagnation, validating its behavioral design. Collectively, these findings establish Rhino optimizer as a competitive and robust bio-inspired optimizer and highlight the value of biologically grounded behavioral mechanisms for high-dimensional, computationally constrained HPO in mental-health applications. Though the proposed Rhino optimizer has been evaluated in the context of transformer-based depression detection, it is not restricted to mental health applications. The optimization strategy is not limited to the problem of depression detection. It is domain-independent and can be applied to engineering design Optimization, intelligent manufacturing, energy forecasting, predictive maintenance, scheduling, and ML model optimization. Thus, this paper goes beyond depression detection, showing the potential of behavior-inspired optimization strategies for real-world optimization problems.

ACKNOWLEDGMENTS

The authors credit the maintainers of the mental health depression dataset on Hugging Face for the release of open-access data, which helped this study. The authors would also like to thank the community behind the open-source Python ecosystem, PyTorch, and Hugging Face transformers, without which the proposed optimization framework could not have been developed or benchmarked.

FUNDING INFORMATION

The authors have declared no funding for this research.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Simisani Ndaba	✓	✓	✓			✓		✓	✓	✓	✓			
Rajalakshmi Selvaraj		✓		✓	✓		✓		✓	✓		✓	✓	
Hlomani Hlomani	✓	✓	✓	✓			✓			✓		✓	✓	

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

There is no conflict of interest declared by authors.

INFORMED CONSENT

The concept of a formal informed consent process is not relevant to this study, which is based on a publicly available and identified public dataset and did not recruit or interact directly with human subjects.

ETHICAL APPROVAL

Ethical approval for this study was not required as it used an open, publicly available deidentified data set and did not include any of the authors collecting, intervening, or interacting with any of the humans or animals in this study.

DATA AVAILABILITY

The data that support the findings of this study are openly available on Hugging Face at https://huggingface.co/datasets/malexandersalazar/mental-health-depression, reference [21].

REFERENCES

[1] V. Gupta, A. Kanungo, P. Kumar, N. Kumar, and C. Choubey, “A design of bat-based optimized deep learning model for EEG signal analysis,” *Multimedia Tools and Applications*, vol. 82, no. 29, pp. 45367–45387, 2023, doi: 10.1007/s11042-023-15462-2.

[2] K. Zerrouki, R. M. Hamou, and A. Rahmoun, “A novel bio-inspired approach based on Spotted Hyenas for Suicide Prediction,” in *2021 International Conference on Information Systems and Advanced Technologies*, Dec. 2021, pp. 1–5, doi: 10.1109/ICISAT54145.2021.9678475.

[3] W. Zhao, L. Wang, and Z. Zhang, “Artificial ecosystem-based optimization: a novel nature-inspired meta-heuristic algorithm,” *Neural Computing and Applications*, vol. 32, no. 13, pp. 9383–9425, Jul. 2020, doi: 10.1007/s00521-019-04452-x.

[4] A. Alrosan, W. Alomoush, M. Youssef, A. W. Nile, M. A. Deif, and R. EL Gohary, “Hyper-parameters tuning using meta-heuristic algorithms for nurses stress detection,” in *2024 2nd International Conference on Cyber Resilience*, Feb. 2024, pp. 1–5, doi: 10.1109/ICCR61006.2024.10533109.

[5] Vandana, S. Srivastava, N. Arora, and V. Gupta, “An efficient deep learning model using Harris-Hawk optimizer for prognostication of mental health disorders,” *International Research Journal of Multidisciplinary Technovation*, vol. 6, no. 4, pp. 106–123, Jul. 2024, doi: 10.54392/irjmt2449.

[6] Y. Li, Y. Zhang, and Y. Cai, “A new hyper-parameter optimization method for power load forecast based on recurrent neural networks,” *Algorithms*, vol. 14, no. 6, May 2021, doi: 10.3390/a14060163.

[7] T. Yu and H. Zhu, “Hyper-parameter optimization: a review of algorithms and applications,” Mar. 2020, *arXiv:2003.05689*.

[8] J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, “Algorithms for hyper-parameter optimization,” in *25th Annual Conference on Neural Information Processing Systems (NIPS 2011)*, 2011, pp. 1–9.

[9] K. Jawad, R. Mahto, A. Das, S. U. Ahmed, R. M. Aziz, and P. Kumar, “Novel Cuckoo search-based metaheuristic approach for deep learning prediction of depression,” *Applied Sciences*, vol. 13, no. 9, Apr. 2023, doi: 10.3390/app13095322.

[10] S. Akyol, “New chaos-integrated improved grey wolf optimization based models for automatic detection of depression in online social media and networks,” *PeerJ Computer Science*, vol. 9, Nov. 2023, doi: 10.7717/peerj-cs.1661.

[11] N. Alruwais, H. Alamro, M. M. Eltahir, A. S. Salama, M. Assiri, and N. A. Ahmed, “Modified arithmetic optimization algorithm with deep learning based data analytics for depression detection,” *AIMS Mathematics*, vol. 8, no. 12, pp. 30335–30352, 2023, doi: 10.3934/math.20231549.

[12] H. Alamro *et al.*, “Modeling of blockchain assisted intrusion detection on IoT healthcare system using ant lion optimizer with hybrid deep learning,” *IEEE Access*, vol. 11, pp. 82199–82207, 2023, doi: 10.1109/ACCESS.2023.3299589.

[13] A. S. Alghawli and A. I. Taloba, “An enhanced ant colony optimization mechanism for the classification of depressive disorders,” *Computational Intelligence and Neuroscience*, vol. 2022, no. 1, pp. 1–12, Jun. 2022, doi: 10.1155/2022/1332664.

[14] R. Shalini, V. Sujitha, and P. Parameswari, “Major depressive disorder detection using dung beetle optimization based QGNN classification,” in *2025 3rd International Conference on Advancements in Electrical, Electronics, Communication, Computing and Automation*, Apr. 2025, pp. 1–7, doi: 10.1109/ICAECA63854.2025.11012635.

[15] A. Seyyedabbasi and F. Kiani, “Sand cat swarm optimization: a nature-inspired algorithm to solve global optimization problems,” *Engineering with Computers*, vol. 39, no. 4, pp. 2627–2651, Aug. 2023, doi: 10.1007/s00366-022-01604-x.

[16] S. Askarpour and M. S. Anari, “Wildebeest optimization algorithm based on swarm intelligence method in solving optimization problems,” *International Journal of Nonlinear Analysis and Applications*, vol. 12, pp. 1397–1410, 2021, doi: 10.22075/IJNAA.2021.5741.

[17] E. Abdulsaed, M. Alabbas, and R. Khudeyer, “Hyperparameter optimization for convolutional neural networks using the salp swarm algorithm,” *Informatica*, vol. 47, no. 9, Dec. 2023, doi: 10.31449/inf.v47i9.5148.

[18] A. Brodzicki, M. Piekarski, and J. J. -Korjakowska, “The whale optimization algorithm approach for deep neural networks,” *Sensors*, vol. 21, no. 23, Nov. 2021, doi: 10.3390/s21238003.

[19] E. Alabdulkreem, R. Marzouk, M. Alduhayyem, M. A. Al-Hagery, A. Motwakel, and M. A. Hamza, “Chameleon swarm algorithm with improved fuzzy deep learning for fall detection approach to aid elderly people,” *Journal of Disability Research*, vol. 2, no. 2, Aug. 2023, doi: 10.57197/JDR-2023-0020.

[20] A. Faramarzi, M. Heidarinejad, S. Mirjalili, and A. H. Gandomi, “Marine predators algorithm: a nature-inspired metaheuristic,” *Expert Systems with Applications*, vol. 152, Aug. 2020, doi: 10.1016/j.eswa.2020.113377.

[21] M. A. Salazar, “Mental health depression dataset,” huggingface.co. 2025. Accessed: Jun. 25, 2026. [Online]. Available: https://huggingface.co/datasets/malexandersalazar/mental-health-depression

[22] N. I. Haque, A. A. Khalil, M. A. Rahman, M. H. Amini, and S. I. Ahamed, “BIOCAD: bio-inspired optimization for classification and anomaly detection in digital healthcare systems,” in *2021 IEEE International Conference on Digital Health*, Sep. 2021, pp. 48–58, doi: 10.1109/ICDH52753.2021.00017.

[23] K. Mastrothanasis, K. Zervoudakis, M. Kladaki, and S. Tsafarakis, “A bio-inspired computational classifier system for the evaluation of children’s theatrical anxiety at school,” *Education and Information Technologies*, vol. 28, no. 9, pp. 11027–11050, Sep. 2023, doi: 10.1007/s10639-023-11645-4.

[24] V. Haripriya and P. Rutravigneshwaran, “Seal optimization driven random forest framework for enhanced alzheimer’s disease classification,” *Journal of Theoretical and Applied Information Technology*, vol. 103, no. 9, pp. 3689–3715, 2025.

[25] S. K. Shivaiah, K. Krishnappa, N. K. Boraiah, P. D. Shenoy, and V. K. Rajuk, “Prediction of mental illness using ensemble model and grid search hyperparameter optimization,” *Bulletin of Electrical Engineering and Informatics*, vol. 13, no. 5, pp. 3726–3736, 2024, doi: 10.11591/eei.v13i5.7309.

[26] S. K. Priya and K. P. Karthika, “A contemporary multi-objective feature selection model for depression detection using a hybrid pBGSK optimization algorithm,” *International Journal of Applied Mathematics and Computer Science*, vol. 33, no. 1, 2023, doi: 10.34768/amcs-2023-0010.

BIOGRAPHIES OF AUTHORS

Simisani Ndaba is a Ph.D. student in Computer Science and Informatics in the School of Pure and Applied Sciences at Botswana International University of Science and Technology. She has a published research track record in machine learning and bio-inspired optimization. Experienced in hyperparameter optimization, metaheuristic algorithm development, and transformer-based text classification applied to mental health. Solid educational foundation and professional experience doing computational and reproducible research. She can be contacted at email: ns24020000@biust.ac.bw.

Rajalakshmi Selvaraj is a professor in Computer Science and Informatics in the School of Pure and Applied Sciences at Botswana International University of Science and Technology, specializing in artificial intelligence and smart systems. She is the group leader for Bioinformatics, Data Science and High-Performance Computing, responsible for fostering cutting-edge research and innovation in data-driven and computational intelligence applications. She can be contacted at email: selvarajr@biust.ac.bw.

Hlomani Hlomani is the head of department and lecturer in Computer Science and Informatics in School of Pure and Applied Sciences at the Botswana International University of Science and Technology. His research interests include artificial intelligence and smart systems, and he is part of the BIUST Software Engineering Lab and the Bioinformatics, Data Science and High-Performance Computing research group. He can be contacted at email: hlomanihb@biust.ac.bw.